DESIGN WEB UX

Préface d'Olivier **Andrieu**

Alexandra **Martin**
Mathieu **Chartier**

Techniques de référencement web

2e éd.

AUDIT ET SUIVI SEO

EYROLLES

Techniques de référencement web 2e éd.

« Ce duo de choc et de charme vous livre les clés techniques du référencement dont vous devrez tenir compte pour un meilleur positionnement. Un livre incontournable ! »

Isabelle Canivet
Experte en stratégie de contenu et SEO
http://www.yellowdolphins.com

« Le monde SEO francophone manquait d'un ouvrage technique destiné aux développeurs n'étant pas obligatoirement spécialistes des arcanes du référencement naturel. »

Olivier Andrieu
Expert en référencement
http://www.abondance.com

Un livre technique pour les référenceurs

L'objectif de cet ouvrage est de répondre aux nouvelles contraintes imposées par le monde du référencement web. Non pas que le **SEO** change radicalement de manière quotidienne, il faut bien avouer que des nouveautés ne cessent d'être annoncées et nous devons être de plus en plus aguerris pour réagir face à ces évolutions.

Après quelques sains rappels sur l'historique des moteurs de recherche et la méthodologie du référencement, ce livre se positionne en effet de façon beaucoup plus technique que les autres sur le sujet. On y découvre par exemple les techniques à maîtriser pour créer un Sitemap ou un fichier robots.txt : c'est un véritable **manuel de programmation** pour référenceur. S'ajoutent à cela deux parties indispensables pour gérer l'audit et le suivi SEO d'un site web.

Un **ouvrage essentiel** pour les chefs de projet web, référenceurs et développeurs ! Enrichie de nombreux exemples et de nouvelles sections sur le référencement d'applications et de sites mobiles, la sécurité et le SEO local, la seconde édition de cet ouvrage est également 100 % à jour sur les derniers algorithmes de Google !

Autodidacte chevronnée, Alexandra Martin, aka Miss SEO Girl, est l'auteur du blog www.miss-seo-girl.com. D'un profil webmarketing et spécialiste du référencement naturel, ses compétences sont complétées par celles de Mathieu Chartier, véritable passionné par la technique et le référencement dès ses premières heures, et auteur de plusieurs livres et du site blog.internet-formation.fr.

AU SOMMAIRE

DESIGN WEB UX

www.editions-eyrolles.com

Techniques de **référencement web**

AUDIT ET SUIVI SEO

DANS LA MÊME COLLECTION

C. LALLEMAND, G. GRONIER. – **Méthodes de design UX.**
N°14143, 2015, 488 pages.

S. DAUMAL. – **Design d'expérience utilisateur.**
N°14176, 2e édition, 2015, 220 pages.

S. POLLET-VILLARD. – **Créer un seul site pour toutes les plates-formes.**
N°13986, 2014, 144 pages.

K. DELOUMEAU-PRIGENT. – **CSS maintenables avec Sass et Compass.**
N°13640, 2e édition, 2014, 252 pages.

J. PATONNIER, R. RIGOT. – **Projet responsive web design.**
N°13713, 2013, 162 pages.

I. CANIVET, J.-M. HARDY. – **La stratégie de contenu en pratique.**
N°13510, 2012, 176 pages.

C. SCHILLINGER. – **Intégration web – Les bonnes pratiques.**
N°13370, 2012, 390 pages.

SUR LE MÊME THÈME

O. ANDRIEU. – **Réussir son référencement web.**
N°14118, édition 2016 (à paraître).

O. ANDRIEU. – **SEO zéro euro.**
N°14033, 2014, 224 pages.

I. CANIVET. – **Référencement mobile.**
N°13667, 2013, 456 pages.

I. CANIVET. – **Bien rédiger pour le Web.**
N°13750, 3e édition, 2014, 736 pages.

D. ROCH. – **Optimiser son référencement Wordpress.**
N°14182, 2e édition, 2015, 296 pages.

E. MARCOTTE. – **Responsive web design.**
N°13331, 2011, 160 pages.

F. DRAILLARD. – **Premiers pas en CSS 3 et HTML 5.**
N°13944, 6e édition, 2015, 472 pages.

Alexandra **Martin**
Mathieu **Chartier**

Techniques de référencement web

2e éd.

AUDIT ET SUIVI SEO

Préface d'Olivier **Andrieu**

EYROLLES

Éditions Eyrolles
61, bd Saint-Germain
75240 Paris Cedex 05

© Groupe Eyrolles, 2016, ISBN : 978-978-2-212-14333-1

Préface

Je me souviens d'un jour de septembre 1993. Après un long voyage en avion et un décalage horaire de nombreuses heures, je sortais des bureaux d'Eyrolles après avoir signé le contrat de mon premier ouvrage, *Internet guide de connexion*, qui allait sortir l'année suivante. Vous vous rendez compte, un livre entier sur les différentes façons de se connecter à Internet... Un sujet qui rendrait le moindre internaute perplexe aujourd'hui... C'était pourtant il y a vingt ans à peine. Pour l'anecdote, ce livre proposait en annexe, sur quelques pages, la liste exhaustive des sites web français avec leur description... Et ce n'était pas écrit petit...

Et puis j'ai créé ma société, Abondance, en 1996, pour proposer une offre de référencement sur Excite, Infoseek et autres Lycos. C'était en période pré-googlienne :-). Autre anecdote : j'ai eu mon quart d'heure de gloire « warholien » en faisant une formation au référencement (qui ne s'appelait pas encore SEO) et au fonctionnement des moteurs de recherche chez... Altavista ! Je vous parle d'un temps que les moins de vingt ans... Vous connaissez la suite...

Pourquoi je vous parle ici tel un ancien combattant ? Parce que, durant toutes ces années, j'ai eu l'occasion de participer à des dizaines, peut-être même des centaines de conférences et de formations sur ce domaine si passionnant qu'est le SEO. Souvent, à la fin de ces « prestations », les gens viennent discuter avec vous, poser des questions, demander des éclaircissements sur certains points ou par rapport à leurs besoins spécifiques. Parfois, également, des participants vous expliquent qu'ils ont beaucoup appris en lisant vos livres, en parcourant votre site, en suivant une de vos formations. Ça fait bien sûr très plaisir, non seulement parce que ça flatte l'ego (ça, on pourrait s'en passer assez rapidement, d'autant plus que le mien n'est pas fondamentalement développé), mais surtout parce que cela vous montre que le travail que vous faites n'est pas vain et qu'il aide d'autres personnes. N'est-ce pas là l'essence même de notre existence ? De nombreux e-mails très sympathiques vous indiquent également que, parfois, un de vos ouvrages a suscité des vocations. Le plaisir que cela procure est évident, même si ça vous renvoie parfois à l'état d'ancêtre...

Alexandra et Mathieu, les auteurs du livre que vous tenez entre les mains, font partie des personnes qui m'ont contacté, un jour, pour me dire qu'ils avaient commencé à « mettre les mains dans le cambouis des moteurs » à l'aide, entre autres, d'un de mes ouvrages. Ils ont fait preuve d'une gentillesse et d'une humilité qui, au-delà des qualités humaines que l'on peut ainsi discerner, sont, selon moi, un ingrédient essentiel pour devenir un bon référenceur. Bref, ce sont « des gens bien ». J'ai été d'autant plus intéressé par leur projet qu'il me semblait combler un manque dans notre domaine : un ouvrage qui s'adresse avant tout aux développeurs, avec une vision technique du référencement. Non seulement une personne qui connaît déjà plus ou moins le monde du SEO et qui veut se perfectionner au niveau du code, mais

également l'inverse, c'est-à-dire le développeur pointu dans son métier mais qui cherche à mieux comprendre le référencement naturel. Je suis certain que ce livre répondra à cette double attente.

Car le challenge est important : depuis vingt ans que je navigue dans les méandres du Web et que je fais des audits SEO, j'ai pu me rendre compte à quel point les sites web n'étaient pas toujours développés en prenant en compte les critères de pertinence de Google – ce qui nécessite parfois de reprendre à zéro un projet. Tout cela parce que les développeurs n'étaient, le plus souvent, pas sensibilisés aux arcanes des moteurs de recherche et à leur fonctionnement. De plus, ils n'étaient pas toujours conscients de tous les outils qu'il était possible d'utiliser et de créer pour optimiser et suivre un site ainsi que sa visibilité.

Je formule donc le vœu que ce livre aide à la fois les développeurs et les spécialistes SEO aux joies du codage. Et, comme il paraît que la programmation informatique est prévue dans les années qui viennent dès le plus jeune âge au programme scolaire de nos chères têtes blondes, je souhaite le meilleur à cet ouvrage : devenir dans le futur un manuel scolaire reconnu pour que les sites web que nos enfants créeront soient nativement *search engine friendly*.

Olivier Andrieu

Éditeur du site Abondance.com

Avant-propos

Pourquoi ce livre ?

Ce livre a été rédigé afin de répondre aux dernières contraintes imposées par le monde du référencement web. Non pas que la SEO change radicalement de manière quotidienne, mais il faut bien avouer que des nouveautés ne cessent d'arriver et que nous devons être de plus en plus aguerris pour répondre à ces évolutions.

Certaines thématiques sont souvent balayées au sein des articles sur la Toile voire dans des ouvrages dédiés au référencement, les auteurs ont souhaité fournir des réponses précises et détaillées sur ces sujets en passant à la fois par la technique, la programmation et l'usage avancé d'outils disponibles sur le marché. Leur volonté a aussi été portée sur l'analyse et le suivi des données ainsi que sur l'audit de sites web pour trouver les forces et faiblesses des pages web en termes de SEO, c'est pourquoi une majeure partie du livre est consacrée à ces sujets.

Ce livre a été rédigé pour donner des explications avancées sur les évolutions des moteurs de recherche mais aussi sur des critères précis et techniques de référencement. En définitive, le contenu est partagé assez équitablement entre théorie, technique, suivi et audit de sites web. Toutefois, tous les sujets ne peuvent pas être traités avec la même minutie, ce qui explique que des choix ont été effectués pour tenter d'apporter un maximum de savoir et de consistance aux lecteurs, tout en mixant pédagogie et technique.

Nombre d'ouvrages de qualité existent sur le sujet, non sans rappeler celui d'Olivier Andrieu, de Daniel Roch ou même de Mathieu Chartier, l'un des auteurs de ce livre, mais ils s'adressent généralement aux débutants en matière de référencement. Ici, la volonté des auteurs a été d'adapter leurs connaissances pour des spécialistes, des développeurs ou webmasters qui souhaitent passer un cap dans ce métier. Non pas que les débutants soient exclus et ne puissent pas comprendre les propos tenus, certaines parties imposent toutefois une bonne connaissance technique des langages et du Web, et bien que les auteurs fassent leur maximum pour clarifier les contenus, des difficultés peuvent être ressenties.

Le livre peut être lu de manière linéaire puisque les auteurs ont suivi la logique méthodologique du référencement en débutant avant tout par l'indexation des pages web puis les optimisations du positionnement pour conclure avec des méthodes de suivi et d'audit approfondies. Pour les plus spécialistes, la lecture

peut également se faire de manière décousue afin de répondre aux besoins directs puisque les sections littéraires ont aussi une certaine indépendance.

Vous pouvez retrouver l'ensemble des codes et programmes du livre à l'adresse http://goo.gl/onrvlx ou sur la fiche de l'ouvrage sur www.edition-eyrolles.com.

À propos des auteurs

Consultante en référencement, Alexandra Martin accompagne les professionnels dans la mise en place de leur stratégie de visibilité globale sur le Web depuis 2008. Formée dans le monde du marketing, la SEO est devenue une passion insatiable qu'elle partage depuis plusieurs années sur son blog www.miss-seo-girl.com au travers d'articles variés et à la portée de tous. Retrouvez-la sur son compte Twitter @Miss_Seo_Girl et sur son blog pour découvrir ou redécouvrir son univers.

Mathieu Chartier est un ancien archéologue reconverti au Web à la fin des années 2000 après avoir suivi un master professionnel dans ce domaine. Passionné par la technique et le référencement dès ses premières heures, il a logiquement développé ces aspects au travers de ses ouvrages, de son blog professionnel (http://blog.internet-formation.fr) mais aussi dans son activité de consultant, formateur et webmaster multitâches. Vous pouvez le retrouver sur Twitter (@Formation_web) ou sur son blog, n'hésitez pas à lui poser des questions ou à lui demander conseil en matière de SEO...

Mathieu Chartier, Olivier Andrieu et Alexandra Martin

Remerciements

Nous tenons à remercier toutes les personnes qui nous ont aidés et suivis lors de la rédaction de ce livre, à commencer par les éditions Eyrolles qui nous ont donné la chance de réaliser ce rêve.

Nos premières pensées vont vers nos proches, nos amis et bien sûr notre famille dont la patience, la tolérance, le soutien et l'amour ont été nos forces pour mener à bien ce projet éditorial. Nos esprits et nos cœurs se tournent vers nos moitiés respectives, Yann et Anne-Sophie, ainsi que vers d'autres personnes bien trop nombreuses pour être marquées à l'encre noire, bien que nous puissions glisser au moins quelques noms : Gaby, Denis, Ilou, Julien, Guillaume, Rodrigo, Yannis, Elena, Chrystelle, Christophe, Sandra, Cynthia, Carine, Céline et tant d'autres sans qui nous ne saurions rien…

Par ailleurs, il nous est impossible de citer tous les référenceurs, marketeurs et développeurs qui ont pu nous aider activement ou inconsciemment mais voici tout de même une liste de noms qui comptent à nos yeux et sans qui la discipline serait fade : Guillaume Degré, Laurent Bourrelly, XavFun, Thomas Cubel, Hasni Khabeb, Ronan Chardonneau, Edouard Ouvrard, Daniel Roch, Dejan Markovic, Isabelle Canivet, Ferréole Lespinasse, Marie Pourreyron, Sandrine Khou, Sylvain Richard, Sébastien Monnier, Renaud Joly, Nicolas Robineau et tous les autres que nous n'avons plus la place de nommer…

Cette liste non exhaustive de spécialistes à qui nous tenons ne pourrait être complète sans présenter notre préfacier Olivier Andrieu, dont les livres nous ont sensibilisés au métier, et sans qui le référencement n'aurait pas la même saveur en France. Nous tenons à le féliciter pour tout ce qu'il a apporté à la sphère SEO avec une humilité sans faille et une passion communicative, mais aussi à le remercier pour son soutien et son aide.

Enfin, nous tenons à remercier l'ensemble des lecteurs de nos blogs respectifs, www.miss-seo-girl.com et blog.internet-formation.fr, et espérons que vous continuerez à nous être fidèles.

Nous vous aimons tous, merci pour votre soutien inégalable.

Alexandra Martin et Mathieu Chartier

Table des matières

Introduction

Tout au long de ce livre, nous allons étudier des techniques avancées en matière de référencement, parfois peu exploitées ou méconnues, afin d'être plus visible et de gagner des positions dans les moteurs de recherche.

Nous traiterons d'une multitude de sujets mais nous insisterons essentiellement sur les parties les plus techniques de la discipline ainsi que sur le suivi et l'audit d'un site web en matière de SEO. Si vous n'êtes pas encore à l'aise avec le vocable relatif à ce domaine, vous pouvez vous référer à l'excellent glossaire de Jonathan Rousseau (source : http://goo.gl/kGl0kh) qui vous permettra d'appréhender sans aucun souci la suite de cet ouvrage.

Pour que l'entrée en matière ne soit pas trop brutale pour les moins connaisseurs d'entre vous, nous allons tout d'abord présenter dans cette introduction un rapide historique des deux moteurs principaux du marché en France, Google et Bing. Nous vous présenterons ensuite les bases techniques essentielles à maîtriser pour comprendre les codes proposés tout au long des chapitres qui composent ce livre. Nous vous souhaitons une agréable lecture…

Historique de Google et Bing

Quelques dates et chiffres clés côté Google/Bing

Historique de Google

Tout commence grâce à la passion démesurée de l'informatique de deux étudiants de l'université de Stanford, Larry Page (22 ans) et Sergey Brin (21 ans), qui se rencontrent en 1995 et deviendront rapidement des amis mais aussi de futurs grands noms du Web.

Dès 1996, ils débutent leur aventure commune avec la création d'un premier moteur de recherche nommé « BackRub ». Ce moteur fonctionnait uniquement via les serveurs de l'université de Standford et était relativement gourmand en bande passante au point d'être abandonné à la demande de l'université.

Une entreprise planétaire

En 2013, la société comptait plus de 50 000 salariés à travers le monde. Tous proviennent d'horizons différents et sont polyglottes, afin de mieux représenter les utilisateurs internationaux de Google.

Les bureaux sont appelés « Googleplex » et la société en dénombre pas moins de soixante-dix dans plus de quarante pays à travers le monde, dont un à Paris.

Il ne faudra pas longtemps pour que le projet revoie le jour sous l'effigie de « Google » puisque dès le 4 septembre 1998, les deux amis créent la société éponyme à Mountain View en Californie, dans la

Silicon Valley. Les deux fondateurs engagèrent rapidement leur premier salarié, Craig Silverstein, et leur société devint la puissante entreprise que nous connaissons.

Figure I-1

Larry Page et Sergey Brin dans leur jeunesse

Depuis 2011, le PDG de Google est Larry Page. C'est la seconde fois qu'il occupe ce poste depuis la naissance de Google. Il a juste cédé sa place entre 2001 et 2011 à Eric Schmidt, ancienne tête pensante de Google et actuellement président exécutif.

Le nom Google a pour origine un terme mathématique, « googol » en anglais, désignant le chiffre 1 suivi de 100 zéros. Ce terme traduit l'ambition des deux fondateurs de gérer un volume infini d'informations sur la Toile. Une rumeur circule et précise qu'au moment de l'enregistrement, une erreur de frappe surgit et au lieu de taper « googol », le mot Google fut tapé et conservé. Le nom de domaine google.com a été déposé le 15 septembre 1997.

Pourquoi tant de « O » ?

Quand nous procédons à une recherche, Google propose un nombre incalculable de pages en bas des résultats. Dans ce cas, la lettre « O » de Google se multiplie comme le terme mathématique « googol » le désigne et devient « goooooogle ».

Le slogan de Google est « Don't be evil », traduit littéralement par « Ne soyez pas malveillants ». Il souligne la volonté de Google de toujours faire les choses correctement d'un point de vue éthique (bien que cela puisse être discuté parfois). Google tente d'appliquer cette règle au sein de sa société et demande la même chose aux référenceurs. Ne soyez pas malhonnêtes, ne trichez pas, soyez fair-play et respectez les consignes. Cette devise s'applique à la lettre et explique les nombreuses pénalités appliquées par le moteur de recherche.

En SEO, on différencie ainsi les « White Hat » (internautes « propres »), les « Grey Hat » (référenceurs qui essaient de rester dans les consignes) ainsi que les « Black Hat » (référenceurs mal intentionnés ou qui outrepassent les consignes) mais tout cela n'est en réalité qu'une histoire de jargon… Il faut surtout

retenir que Google édicte des *guidelines* qu'il est préférable de respecter, mais des Black Hat sont aussi des référenceurs qui font progresser la discipline avec leurs découvertes, il faut donc aussi admettre qu'ils jouent un rôle parfois favorable au SEO.

Google se caractérise par un historique riche, rempli d'événements majeurs. À dire vrai, la société est en perpétuelle évolution depuis sa naissance en 1998. Elle rachète sans cesse des entreprises, lance de nouveaux services ou met à jour l'existant. Nous pouvons citer notamment Analytics, AdWords, Gmail, YouTube, Chrome, Google+, etc., autant de services que Google met gratuitement à disposition de ses utilisateurs, bien que certains n'aient pas duré tels que Google Wave ou encore l'illustre agrégateur iGoogle.

Terminons notre tour d'horizon par un historique daté de Google :

- octobre 2000 : lancement de Google AdWords ;
- septembre 2002 : Google Actualités (avec déjà 4 000 sources d'actualités) est lancé ;
- avril 2004 : naissance du service Gmail ;
- août 2004 : Google annonce son entrée en bourse à Wall Street ;
- février 2005 : lancement de Google Maps ;
- juin 2005 : naissance de Google Earth ;
- novembre 2005 : mise en place de Google Analytics ;
- 2006 : lancement de Picasa (janvier), de Google Agenda (avril), de Google Trends (mai) et de Google Apps (août) ;
- octobre 2006 : rachat de YouTube ;
- novembre 2007 : lancement de l'OS mobile Android ;
- septembre 2008 : le navigateur Chrome est déployé ;
- octobre 2009 : accord avec Twitter pour insérer les tweets dans les résultats de recherche (ce partenariat n'est plus d'actualité) ;
- mars 2011 : lancement du bouton +1 pour « liker » les pages ;
- juin 2011 : naissance du réseau social Google+.

Plus d'informations sur l'historique

Pour en savoir plus sur l'historique détaillé de Google, vous pouvez consulter la page suivante : https://www.google.fr/about/company/history/.

Google est le leader du marché de la recherche web, nous le savons mais les statistiques nous le confirment constamment. Les chiffres sont éloquents tant Google a connu une progression fulgurante dès ses premiers pas dans le monde :

- en mai 2014 en France, Google détient environ 92 % de parts de marché (source : http://goo.gl/kZtljH) ;
- l'index de Google contient environ 30 trillions de documents ;
- Google met à disposition de ses usagers près de 200 produits et services (API, apps mobiles…) dont Blogger, Google Agenda, Google Earth, Google Docs, Google Alerts, etc. ;

- Google a fait l'acquisition de plus de 100 entreprises dont certaines très connues comme Picasa (juillet 2004), Keyhole (octobre 2004, devenu ensuite Google Earth), Urchin Software (mars 2005, utilisé pour créer Google Analytics), YouTube (octobre 2006), FeedBurner (mai 2007, spécialiste du flux RSS et Atom) ou encore eBook Technologies (janvier 2011).

La stratégie de Google

L'objectif de Google est d'investir dans des entreprises spécialisées dans divers domaines d'activité afin d'étoffer son offre de produits et de services sans passer par des prestataires extérieurs. Google s'ouvre de plus en plus aux réseaux sociaux, au monde mobile, à la robotique, à la domotique et à l'e-commerce.

Historique de Bing

Bing est un moteur de recherche récent développé par Microsoft après avoir mis en place des technologies de recherche telles que Live Search, Windows Live Search (abandonné dès 2011) ou encore MSN Search. Il a été lancé officiellement le 3 juin 2009.

« Bing » provient d'une onomatopée inventée par Microsoft car ce nom était simple à retenir, fonctionnel et pouvait être apparenté au son émis nativement en cas de prise de décision sur Windows. Pour l'anecdote, Bing aurait pu ne jamais exister sous ce nom puisqu'il était à l'origine surnommé Kumo dans les captures de présentation en mars 2009 (source : http://goo.gl/zA70rd), ce qui montre que la firme cherchait un nom court, percutant et facilement mémorisable.

Figure I-2

Bing aurait pu s'appeler Kumo.

Dès le mois de juillet 2009, peu de temps après le lancement du moteur, un partenariat entre Microsoft et Yahoo! a été signé afin de fonder une alternative plus puissante et efficace contre Google. Désormais, la technologie de recherche de Bing est déployée sur le moteur de Microsoft mais aussi sur Yahoo!, bien que la gestion des liens sponsorisés soit quant à elle administrée essentiellement sur les compétences de Yahoo! et de ses ingénieurs.

En mai 2010, Microsoft a réussi à obtenir un partenariat avec les constructeurs de la marque BlackBerry afin que son moteur soit déployé de manière systématique sur ces supports mobiles.

Retenons également que le logo de Bing a été totalement revisité en 2013 à l'image de Windows 8 et de son *flat design*. Avant cette refonte, le moteur arborait un logo textuel bleu qui a été remplacé par un « b » schématique accompagné du mot « Bing » tout en jaune.

Évolutions des moteurs de recherche

Les moteurs de recherche ont tous connu des progressions nettes ses dernières années, que ce soit Google, Bing, Yahoo!, Baidu, Ask ou Yandex. Il est impossible de détailler toutes les évolutions, mais nous allons présenter rapidement ici quelques mises à jour et services qui ont modifié le visage de la recherche web dans le monde.

Quelques mises à jour de Google

Il est important de se rappeler que l'année 2009 a été marquante chez Google et que le moteur que nous utilisons quotidiennement n'a plus du tout la même allure... Cette année a marqué l'arrivée de la recherche en « temps réel » et de la recherche universelle avec Google Maps dans les résultats de recherche par exemple.

Parmi les évolutions les plus marquantes dans l'histoire de Google, il ne fait aucun doute que Google Caffeine, déployée dès juin 2010, a changé l'approche du moteur de recherche en termes d'indexation et de perception des pages web. Caffeine a été une mise à jour complète du système d'indexation des pages dans le moteur de recherche afin de booster énormément la méthode de crawl des robots et la qualité d'enregistrement. L'autre mise à jour majeure du moteur a certainement été Google Mayday, déployée un mois avant (mai 2010) et qui a permis de mieux interpréter les mots-clés et les requêtes larges issues de la « longue traîne ».

D'autres mises à jour ont aussi été importantes mais elles relèvent plutôt de l'acquisition ou de la création de nouveaux services, ainsi que de la mise en place de filtres et de pénalités de plus en plus exigeantes. Nous traiterons ce sujet plus en détail par la suite... Nous aurions pu citer également le développement de la recherche en temps réel, notamment autour du *Minty Fresh Indexing* destiné à enregistrer des pages quasiment en instantané afin de toujours proposer les meilleurs résultats aux internautes en temps et en heures.

Continuons notre tour des mises à jour avec le *Knowledge Graph* (ou « graphe de connaissances ») de Google destiné à apporter des informations complémentaires aux usagers lors des recherches. Lancé le 16 mai 2012 aux États-Unis, ce nouvel outil intégré dans les résultats de recherche a pour objectif de comprendre les attentes des internautes et de leur fournir des informations complémentaires sur leur recherche.Le processus se déroule en trois temps :

- l'analyse par Google de la recherche de l'internaute (analyse de chacun des mots-clés de la requête) ;
- la création de relations entre les mots-clés (Google fait appel à une gigantesque base de données pour nouer des liens sémantiques). Fin 2012, cette base de données contenait plus de 500 millions d'entités, ainsi que plus de 3,5 milliards de faits et de relations entre ces différents objets ;

- la proposition de résultats encore plus performants et des informations complémentaires sur la recherche effectuée.

Figure I-3

La partie à droite est dédiée au Knowledge Graph.

Le Knowledge Graph de Google analyse le sens des requêtes et tente d'apporter des données supplémentaires pour agrémenter la recherche des internautes. Il est basé sur l'approche ontologique, un modèle de structuration des données pour lequel : un objet de base est une entité, un attribut est une caractéristique de l'objet, une relation est un lien entre plusieurs objets et une classe est un ensemble d'entités.

Il s'agit d'un modèle sémantique et intelligent qui comprend les entités du monde réel et les éventuelles relations qui les lient les unes aux autres. Le Knowledge Graph a donc participé à l'évolution de la recherche sémantique et a apporté une nouvelle vision de la recherche.

Dans les faits, cette évolution marquante de la recherche s'appuie sur des sources publiques telles que Wikipedia, Freebase (abandonnée peu à peu depuis 2015) et CIA World Factbook. Le Knowledge Graph s'applique surtout quand il s'agit de monuments célèbres, de personnalités, de lieux géographiques marquants, de films et musiques, d'œuvres d'art, mais également de sites de marques ou reconnus sur la Toile (mais dans une moindre mesure).

Pas d'impact sur le positionnement

Être dans le Knowledge Graph n'impacte pas directement le positionnement, mais l'effet inverse peut être imaginé. En effet, il est extrêmement rare de voir apparaître le bloc d'informations dans un autre contexte que sur une requête basée sur le nom de domaine d'un site, autrement dit pour une page positionnée en tête des résultats. En définitive, l'intérêt est de renforcer sa visibilité sur des requêtes fortes car les usagers auront du mal à ne pas faire confiance à un site bien positionné mais également présent dans la zone du Knowledge Graph.

Les dernières mises à jour marquantes du moteur de recherche sont certainement Google Hummingbird (2014) et RankBrain (2015), des algorithmes destinés à mieux analyser et comprendre les contenus des pages web et les requêtes des internautes. Nous reviendrons en détail sur ces sujets au début du premier chapitre.

Mises à jour de Bing

Bing est un moteur de recherche assez innovant qui reprend dans les grandes lignes ce que propose Google, ce qui lui a d'ailleurs valu des attaques dans sa courte histoire tant les analogies étaient nombreuses. En pratique, voici les fonctionnalités proposées : recherche d'actualités, de produits (abandonnée en 2013), d'images et de vidéos, Bing Maps (cartographie), recherches associées, historique des recherches, météo en direct, traducteur et calculatrice en ligne, Bing Rewards (un système de crédit présent seulement aux États-Unis) ou encore Bing Voyages.

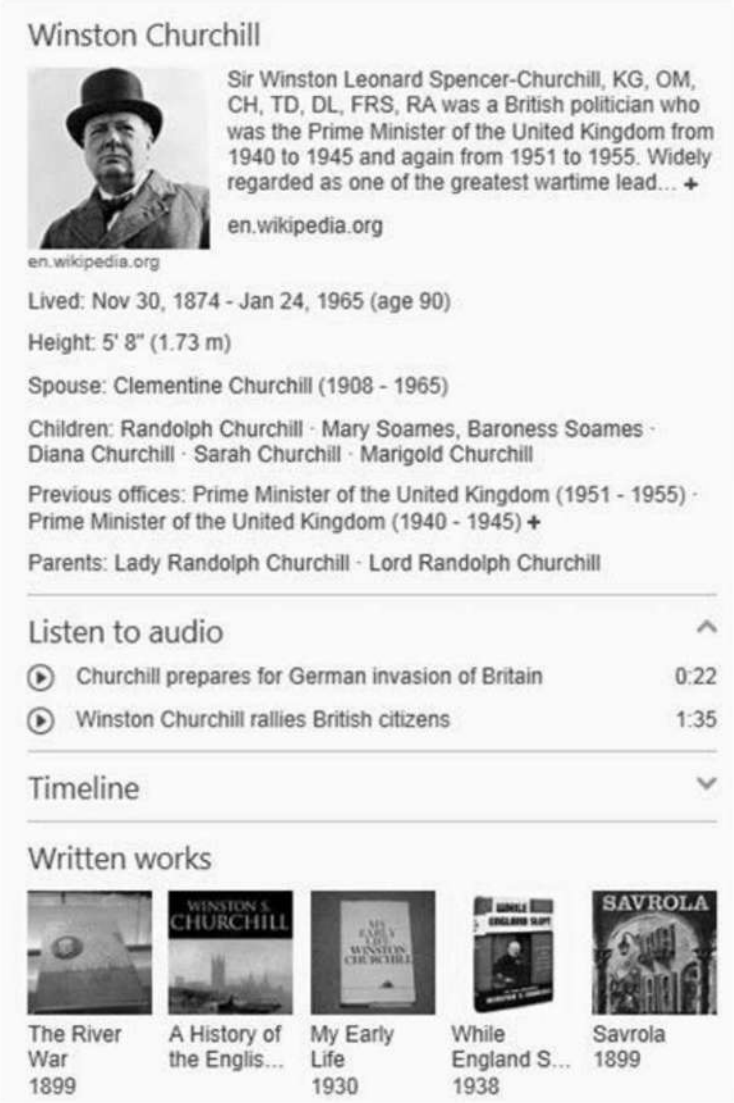
La technologie de recherche de Microsoft est axée autour d'algorithmes tels que le BrowseRank, créé dès 2008 (source : http://goo.gl/rdxuqr), et le StaticRank que nous détaillerons plus tard.

Microsoft communique peu autour de ses mises à jour, sauf si ces dernières sont majeures. Nous allons dresser un rapide historique des mises à jour récentes et marquantes du moteur de recherche.

- Juin 2009 : dans la foulée du lancement de Bing, Microsoft a lancé Bing Travel (ou Bing Voyages) après le rachat du service Farecast en avril 2008 (initialement aux États-Unis).
- Décembre 2009 : Bing Maps est lancé pour contrer Google Maps. Dans les faits, ce type de service existait déjà chez Microsoft depuis 2005 avec Windows Live Local basé sur la technologie Microsoft MapPoint mais le nom de Bing Maps a été attribué en décembre 2009 afin de coller à la nouvelle politique commerciale de la firme. Le service utilise Silverlight de Microsoft et a connu nombre de mises à jour de cartes et de technologies dans son histoire, notamment le 15 décembre 2010 avec une mise à jour graphique de l'outil autour d'un nouveau fond de cartes (après une autre refonte du 7 décembre 2010) et le 12 juin 2013 via l'ajout de 270 To de données.
- Septembre 2010 : Bing Rewards a été annoncé et déployé aux États-Unis afin d'offrir un système de crédits en fonction des recherches des internautes et du temps de diffusion des annonces. Cet ingénieux système économique n'a pourtant pas encore été mis en place partout dans le monde.
- 10 mai 2012 : lancement du Knowledge Graph de Bing appelé Bing Snapshot, soit six jours seulement avant celui de Google (essentiellement déployé aux États-Unis). Le 31 mars 2014, Richard Qian de l'équipe d'indexation et sémantique de Bing a indiqué que plus de 150 millions d'entités ont été ajoutées dans Bing Snapshot (source : http://goo.gl/pnjufr).

Figure I-4

Bing Snapshot, le graph de connaissances de Microsoft



- Mi-2013 : Bing déploie son système de produits sponsorisés dans la lignée d'un Google Shopping. Initialement, Microsoft avait déployé son comparateur de prix Bing Shopping (anciennement Microsoft Live Shopping) basé sur Ciao, racheté en 2008, mais ce service a été arrêté en octobre 2013 au profit des produits sponsorisés avec Bing Products Ads, accessibles par l'interface de gestion des liens payants du moteur.
- 11 octobre 2013 : Bing Snapshot accueille désormais les auteurs de contenus grâce au partenariat de Microsoft avec le service Klout (source : http://goo.gl/2xMlyA). L'outil permet donc de développer un AuthorShip à la manière de Google, nous détaillerons ce sujet dans les prochains chapitres.
- Mises à jour de l'algorithme de pertinence en mars 2012, fin août 2012, en décembre 2013 et le 5 mai 2014.

- 16 juin 2014 : Microsoft annonce qu'il travaille sur un projet intitulé Bing Catapult (source : http://goo.gl/sqgC3y), une nouvelle infrastructure complète des datacenters du moteur de recherche pour accélérer et améliorer l'indexation des pages web et la pertinence des résultats. Le déploiement a été réalisé début 2015 aux États-Unis, puis plus tard dans le monde entier.
- 14 mai 2015 : Bing annonce le déploiement d'un algorithme spécifique à la recherche mobile (source : http://goo.gl/AI08X3), un mois après avoir affiché un libellé « mobile-friendly » dans les SERP mobiles.
- 15 juin 2015 : Bing passe totalement en HTTPS avec SSL. Les URL du moteur de recherche sont désormais toutes sécurisées (source : https://goo.gl/5mSRIX).

Il est souvent reproché à Bing de copier Google. Bien que cela ne soit pas toujours vrai quand nous comparons les dates de lancement des services, des similitudes sont à déplorer telles que la dernière en date avec *Bing Rich Captions*, un système équivalent aux extraits enrichis de Google que nous détaillerons bientôt...

Figure I-5

Prix et disponibilité affichés avec Bing Rich Captions

Peut-on vivre sans Google ?

Nous considérons souvent que Google est seul au monde, mais il ne faut jamais enterrer les concurrents qui ont parfois un rôle à jouer. Il suffit de suivre les différents baromètres des parts de marché dans le monde pour s'en rendre compte. Google domine, écrase parfois, mais n'est pas toujours prédominant, comme c'est le cas en Russie, en République Tchèque ou encore en Chine.

Aux États-Unis, les parts de visites sont plus réparties que dans d'autres pays bien que Google domine avec 67 % devant près de 19 % pour Bing et 10 % pour Yahoo! (source : http://goo.gl/HbTEca).

Si notre marché est francophone, force est de constater que Google domine outrageusement et cela explique en partie pourquoi nous évoquons essentiellement ce moteur de recherche au sein de cet ouvrage. Par ailleurs, Google communique bien plus que Microsoft au sujet de son outil, ce qui ne nous permet pas toujours d'être exhaustifs à propos de Bing.

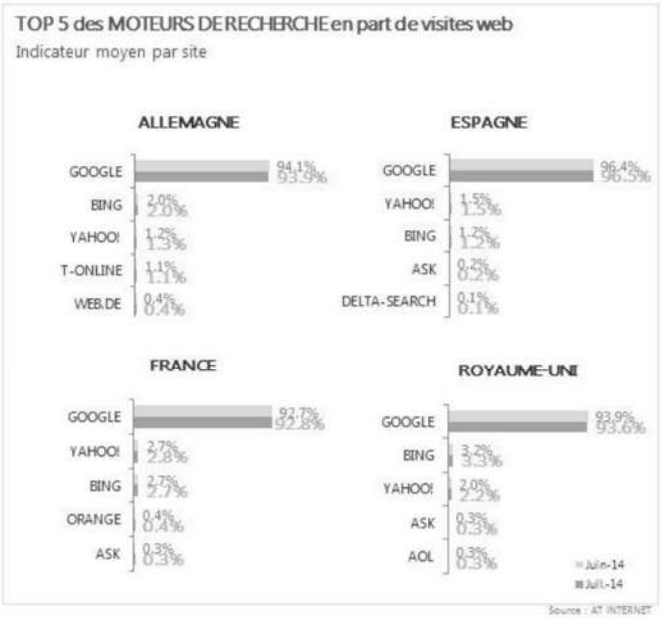
L'institut AT Internet suit l'évolution des parts de marché des moteurs de recherche et montre l'avance considérable de Google sur ses concurrents en Europe occidentale. Nous évoquerons toutefois des moteurs « secondaires » français tels que Qwant ou Exalead pour leurs différences techniques et leurs innovations bien qu'ils ne bénéficient pas d'une place prédominante dans l'esprit des internautes.

Néanmoins, si nous ciblons un marché mondial ou tout du moins marqué par les pays comme la Russie ou la Chine, il est certain que nous devrons mieux maîtriser des moteurs de recherche tels que Yandex ou Baidu notamment. La malice de Microsoft pour conquérir les résultats internationaux de Baidu en

Chine montre que nous devons à tout prix nous focaliser sur ce moteur si nous voulons toucher les visiteurs chinois.

Figure I-6

Répartition des parts de marché des moteurs en Europe



La programmation au service du référencement

Internet est un monde complexe dans lequel s'affrontent nombre de technologies et de services. Nous pensons souvent maîtriser le Web et ses spécificités, mais sa richesse est telle que nous ne faisons que rarement le tour de la question, et c'est peu dire…

Force est de constater que le référencement fait partie des disciplines en vogue à la portée de tout passionné disposant d'un socle de connaissances suffisant pour administrer des sites web. Malheureusement, trop de spécialistes se voient limités par la barrière du développement et du « code » qui pourrait leur permettre d'aller plus loin en imaginant des programmes adaptés à leurs besoins ou tout simplement en gagnant du temps de gestion. Certes, de nombreux développeurs réalisent cette tâche pour nous, mais cela implique souvent des frais que nous pourrions éviter si nos capacités en programmation étaient plus développées.

L'objectif de cette introduction est de fournir (ou rappeler) des bases en matière de développement, sans pour autant rentrer dans des détails obscurs puisque d'autres livres sont bien plus adaptés pour répondre à ce besoin. Néanmoins, nous allons nous remettre les idées en place voire apprendre des bases afin de pouvoir améliorer notre référencement en conséquence, comme nous le verrons au cours des divers programmes disponibles dans le livre.

Différencier les langages côté client et côté serveur

La diversité des langages est souvent le premier frein pour les non-initiés. Les choix sont si nombreux en informatique que nous ne savons jamais avec quel langage commencer et surtout pourquoi en utiliser un plutôt qu'un autre. En réalité, la Toile se limite à une quinzaine de langages spécifiques et n'est pas débordée comme la programmation informatique pure qui se compose de plusieurs dizaines d'écritures telles que C++, Ada, Cobol, PacBase, Visual Basic et tant d'autres. Tous ne sont pas obligatoires pour affronter la programmation. Dans cette infime collection, il est important de distinguer les langages initialisés côté client et côté serveur.

Le client correspond au navigateur web, c'est-à-dire l'ordinateur courant, tandis que le serveur est un logiciel installé sur une machine distante (souvent gérée par un hébergeur externe dans le cas des petites structures ou des particuliers).

Globalement, un langage qui s'exécute côté client peut afficher des contenus dans un navigateur et gérer des animations. *A contrario*, un langage côté serveur a une panoplie bien plus développée car il peut notamment s'interfacer avec des bases de données (comme PostgreSQL, MySQL, DB2, SQL Server, Oracle ou encore SQLite) mais aussi gérer des fichiers divers (PDF, CSV...), se connecter à des API (applications web), traiter des formulaires et permettre de créer des interfaces d'administration *(backoffice)*.

Les usages sont multiples et cette liste de fonctionnalités est non exhaustive. Il ne faut donc pas se tromper de langages selon les besoins, c'est pourquoi nombre de codes de ce livre seront basés sur des programmes exécutés côté serveur.

Côté client, les langages sont peu nombreux : HTML (et HTML 5) pour la mise en page, CSS pour la mise en forme (CSS 3 actuellement mais le CSS 4 est en développement depuis 2010), JavaScript pour les animations et scripts divers (ou par le biais de bibliothèques telles que jQuery, Rico, Prototype, Mootools...), VBS ou JScript pour être compatible avec les technologies de Microsoft, Flash et enfin les applets Java ou ActiveX (Microsoft).

Côté serveur, la liste peut être plus exhaustive mais si nous nous arrêtons aux langages purement web, ceux-ci sont rares. Dans les faits, plusieurs langages sont issus du monde informatique et ont proposé des modules web avec le temps (comme Python). Nous pouvons citer des systèmes de codage tels que PHP, ASP et ASP.Net de Microsoft, Java (JSP et servlets), ColdFusion, Python, Perl ou Ruby on Rails. Tous reprennent les grandes lignes des langages de programmation mais leur syntaxe et les techniques de développement peuvent totalement varier de l'un à l'autre.

Microsoft propose des technologies web

ASP et ASP.Net ne sont pas des langages de programmation, ce sont des technologies mises en place par Microsoft. Nous programmons en VBScript (« VBS » ou « Visual Basic Script ») et JScript pour ASP, ou en C# ou VB.Net en ASP.Net. Il faut maîtriser au moins un de ces langages pour coder sur Microsoft.

La société Tiobe[1] dresse la liste des langages de développement informatique préférés des usagers. Sur le Web, le langage qui sort vainqueur côté serveur est PHP tant il est efficace et à la portée de tous. D'autres langages peuvent aussi être utilisés comme VBS ou C# pour les utilisateurs de serveurs Microsoft (IIS). Python est à la mode actuellement, bien que ce ne soit pas un langage web à l'origine (seules des bibliothèques externes le rendent utile sur le Web).

Figure I-7

Liste des langages préférés des développeurs

Dec 2013	Dec 2012	Change	Programming Language	Ratings	Change
1	1		C	17.890%	-0.81%
2	2		Java	17.311%	-0.26%
3	3		Objective-C	10.202%	-0.91%
4	4		C++	8.268%	-0.94%
5	5		C#	5.620%	+0.07%
6	6		PHP	5.281%	-0.26%
7	7		(Visual) Basic	3.752%	-1.42%
8	8		Python	2.210%	-1.64%
9	21	⇈	Transact-SQL	1.877%	+1.30%
10	11	^	JavaScript	1.852%	+0.53%
11	15	⇈	Visual Basic .NET	1.688%	+0.80%
12	9	v	Perl	1.072%	-1.10%
13	10	v	Ruby	0.932%	-0.80%
14	17	^	MATLAB	0.708%	+0.10%
15	12	v	Delphi/Object Pascal	0.691%	-0.29%

Quel rôle pour le référencement ?

Connaître les différents langages, leur utilité et leur degré d'intérêt permet de sélectionner les technologies adaptées à chaque besoin. En référencement, il est important de distinguer tous ces langages car certains d'entre eux sont impitoyables et peuvent engendrer des pertes de trafic importantes. Il convient de rester méfiant pour toujours trouver le bon langage et le code idéal pour chaque usage.

Tout au long de notre périple, nous allons être confrontés à une multitude d'outils avec des méthodes de conception parfois bien différentes. En effet, nous n'utilisons pas un moteur de blog tels que WordPress ou DotClear comme un CMS tels que Drupal, Spip ou Django (en Python) ou encore un framework comme eZpublish, Symfony, CakePHP, Zend ou Play! (en Java et Scala). Il faut bien maîtriser le code pour se lancer dans l'utilisation de certains outils, c'est l'une des raisons qui explique le succès de WordPress, par exemple, tant son utilisation avancée reste accessible.

1. http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Lorsque les référenceurs doivent se plonger dans les codes pour optimiser les contenus, ils se limitent souvent aux balises HTML car cela reste le fondement essentiel de ce métier. Toutefois, il est parfois nécessaire d'aller plus loin pour automatiser des tâches, ou tout simplement pour gérer des contenus profonds dissimulés dans des morceaux de code écrits avec d'autres langages. Dans ce cas, il est important d'avoir au moins les bases de développement pour se repérer mais aussi pour créer nos propres fonctions utiles pour le référencement. C'est tout l'objet de notre lecture…

Afin que tout soit bien clair après cette introduction, sachez que nous ne rappellerons que le B.A.-ba de la programmation dans peu de langages et que nous ne proposerons pas de codes orientés objet car cette syntaxe, aussi qualitative soit-elle, n'est pas encore la plus courante (les développeurs les plus aguerris et jeunes se tournent vers cette syntaxe mais ce phénomène n'est pas généralisé).

Par conséquent, nous ne traiterons que des bases HTML-CSS, en PHP et JavaScript mais nous négligerons le reste, bien que ce livre présente des codes fondés sur la technologie ASP de Microsoft. Nous devons faire des choix et réduire ces explications à quelques lignes auraient peu de valeur, nous préférons que vous vous orientiez vers des ouvrages exhaustifs sur ces sujets si vous craignez de ne pas comprendre tous les programmes commentés présents tout au long de notre propos.

Différence entre développement procédural et POO

Il faut savoir qu'il n'existe pas de réelles différences en matière de performance selon que nous codons avec la méthode procédurale ou orientée objet (POO), contrairement à ce que pléthore de développeurs avancent pour vanter les mérites de telle ou telle technique. Chacun est libre de coder comme il le souhaite tant que le résultat est fonctionnel, compréhensible (bien commenté notamment) et réutilisable par un tiers.

WordPress est un bon exemple car son code Open Source est contrebalancé entre la programmation procédurale et orientée objet selon les développeurs qui interviennent à la source. Nous trouvons même des morceaux de codes en PHP 4 et en PHP 5 mélangés. Les bases sont essentielles pour ne pas être perdu…

Connaître la programmation web a toujours fait débat dans le milieu du référencement. D'un côté, les réfractaires se refusent à coder la moindre ligne car ils estiment que ce n'est pas toujours utile et que des spécialistes peuvent le faire à leur place. D'un autre côté, au contraire, les partisans voient dans la maîtrise du code un plus non négligeable et surtout un moyen de mieux gérer leur référencement.

Force est de constater qu'un non-développeur ne peut pas tout faire en matière de référencement tant ses limites techniques l'empêchent d'imaginer des programmes automatisés ou même de répondre à certains besoins complexes. En effet, comment gérer le fichier `.htaccess` pour optimiser le référencement si nous ne savons rien coder ? Comment améliorer un système multilingue si nous ne différencions pas la bonne méthode des mauvaises techniques ? Ce ne sont que des exemples, mais la liste pourrait facilement s'étendre si nous grattions un peu le sujet…

Désormais, les référenceurs sont nombreux sur le marché du travail et nombre d'agences préfèrent trouver la perle rare voire le couteau suisse plutôt que d'avoir un référenceur spécialiste uniquement de l'aspect webmarketing. Dans ce cas, avoir de bonnes connaissances techniques permet de sortir du lot et de trouver un poste plus polyvalent avec des possibilités accrues en matière d'optimisation.

L'autre problématique vient de la taille réduite des entreprises et des start-ups relatives à Internet. Nous trouvons beaucoup de PME voire de TPE dans ce secteur, et nous n'avons donc pas toujours des développeurs à portée de main pour coder à notre place. Il faut par conséquent mettre les mains dans le cambouis pour trouver des solutions.

Au fond, souhaitons-nous être éternellement limité et dépendant d'autres personnes ou préférons-nous contrôler le référencement de A à Z ? Bien entendu, il n'est pas utile d'être un génie du développement ou de tout maîtriser. D'une part, cela est impossible et d'autre part, seulement une infime partie serait directement utile pour gagner du temps et optimiser nos tâches. Malgré tout, retenons qu'une bonne maîtrise technique confère une réelle indépendance et une liberté dans l'exécution du métier, c'est pourquoi nous verrons diverses applications SEO dans la suite de notre lecture.

Socle technique pour programmer sur le Web

Langage HTML et mise en page

HTML est un langage de balisage commun qui se décompose en deux parties encadrées par des balises `<html>` et `</html>`.

- La « tête », balisée par `<head>` et `</head>`, contient les informations nécessaires au bon fonctionnement de la page, dont le titre de la page si cher au référencement ou encore la balise permettant de relier le document à des styles CSS.
- Le « corps » est encadré par les balises `<body>` et `</body>` et contient toutes les données visibles par les utilisateurs, à savoir les contenus, les images, etc.

Ces deux parties insérées entre les balises `<html>` et `</html>` sont surmontées d'un doctype, c'est-à-dire une déclaration de type de document qui permet d'indiquer au navigateur et aux robots quelle version du langage est utilisée. Par exemple, si vous choisissez HTML 5, le doctype est simpliste :

```
<!DOCTYPE html>
```

Une fois cette structure initiale réalisée, vous pouvez démarrer en remplissant la tête du document avec les éléments essentiels (jeu de caractères à utiliser, titre et description du document, mots-clés associés, lien vers les styles CSS, scripts JavaScript…).

Quel charset utiliser ?

Le choix du jeu de caractères est primordial en développement web. Les Européens sont toujours partagés entre l'ISO-8859-1 (caractères occidentaux) et l'UTF-8 (tous types de caractères), mais il est fortement conseillé d'opter pour le second choix, bien plus polyvalent et compatible à l'international.

Voici à quoi peut ressembler notre début de page web en HTML :

```
<!DOCTYPE HTML>
<html>
<head>
```

```
<meta charset="utf-8"/>
<title>Titre du document</title>
<meta name="description" content="Description du document"/>
<meta name="keywords" content="mot-clé 1, mot-clé 2, ..."/>
<link rel="stylesheet" type="text/css" href="style.css"/>
<script type="text/JavaScript">// Scripts potentiels...</script>
</head>
<body>
Contenu de la page (visible par les utilisateurs)
</body>
</html>
```

Il faut ensuite se concentrer sur le corps de la page qui contient les contenus des pages web à l'aide de balises « block » (qui forcent un retour à la ligne et prennent des dimensions si nécessaire) et « inline » (sans retour à la ligne ni largeur et hauteur configurables) plus ou moins nombreuses selon la version HTML utilisée.

Les balises de type « block » les plus connues sont `<div>` (bloc de contenu), `<h1>` (jusqu'à `<h6>`) et `<p>` (paragraphes de texte) tandis que les balises en ligne sont plus nombreuses, telles que `<span>` (balise pour styliser des parties de contenus), `<strong>` (mise en exergue représentée par une mise en gras), `<em>` (mise en avant avec affichage en italique) ou encore `<a>` (liens hypertextes).

Quel rôle pour le référencement naturel ?

La quasi-totalité de ces balises a un impact sur le référencement naturel, les mots-clés contenus entre ces éléments ont plus de valeur, à l'exception des éléments `<p>` et `<div>` qui sont « neutres ».

En HTML 5, nous trouvons davantage de balises structurelles telles que `<header>` (en-tête), `<footer>` (pied de page), `<nav>` (zone de navigation), `<aside>` (contenus annexes), `<hgroup>` (groupements de titres), `<section>` et `<article>` pour les contenus. Cette mouture offre plus de précision sémantique et permet de mieux hiérarchiser les textes. Nul doute que cela influencera le référencement et le positionnement des pages dans un avenir proche.

Enfin, avant de donner un exemple complet d'un corps de page, il faut savoir que les menus sont souvent réalisés à l'aide de balises de listes (`<ul>…</ul>`) dans lesquels chaque item de liste (`<li>…</li>`) contient un lien hypertexte.

```
<body>
<div id="bloc-general">
    <header>
        <h1>Logo du site</h1>
</header>
<nav>
    <ul>
        <li><a href="page1.html">Page 1</a></li>
        <li><a href="page2.html">Page 2</a></li>
        <li><a href="page3.html">Page 3</a></li>
```

```
    </ul>
</nav>
<section>
    <article>
        Texte avec <strong>mise en gras</strong> et <em>mise en italique</em>.
    </article>
</section>
<footer>
    <p>Pied de page</p>
</footer>
</div>
</body>
```

Nous n'avons pas pu traiter l'ensemble des spécificités des versions d'HTML ici. Vous pourrez également rencontrer des balises de formulaire (comme `<form>`, `<input />`...) ou de tableaux (comme `<table>` ou encore `<tr>` et `<td>`), par exemple. Il est fortement conseillé de maîtriser ce langage tant il est primordial pour le référencement. Son compère, le CSS, est utile pour réaliser la mise en forme de vos pages web mais il ne joue pas de rôle majeur en SEO.

Gérer les animations avec JavaScript, jQuery, Prototype et consorts...

Dans la lignée des langages exécutés côté client, JavaScript et toute sa panoplie de bibliothèques sont à connaître pour donner vie aux pages web tant les animations et les interactions générées par ces codes peuvent apporter un plus pour les utilisateurs. Néanmoins, il est important de garder en mémoire que le JavaScript et ses acolytes ne sont pas les plus grands amis de Google car une grande partie des codes sont bloquants pour les robots du moteur de recherche. Il convient donc de l'utiliser avec parcimonie.

Généralement, la programmation est fondée sur un principe simple : l'usage de « variables », à savoir des entités créées pour être réutilisées comme bon nous semble en cas de besoin. Les variables peuvent recevoir une valeur fixe ou dynamique, ce qui leur permet d'interagir avec les programmes en fonction des données captées ou directement proposées par l'utilisateur.

En JavaScript, tout comme dans les bibliothèques associées, une variable est définie en ajoutant le mot-clé `var` avant son nom (`var uneVariable`). Une variable peut prendre plusieurs types de valeurs, par exemple une chaîne de caractères, un numérique ou encore un booléen (`true`/`false`).

Le deuxième point qui a fait le succès de la programmation est le système conditionnel qui permet d'effectuer des actions précises en fonction d'hypothèses fixées au préalable. Ce mécanisme fonctionne avec la syntaxe `if (condition) {...} else {...}`.

L'autre facteur essentiel à comprendre lorsque nous développons sur le Web est le système des boucles, dont l'objectif est de pouvoir répéter des actions autant de fois que nous le souhaitons. Trois types de boucles se retrouvent dans la majorité des langages :

- `while` signifie « pendant que la condition est vrai, fais ceci » et s'écrit `while (condition) {...};`

- `do ... while` est à peu près équivalente, à la seule différence que le premier tour de boucle est réalisé, même si la condition n'est pas respectée à la fin du tour. Sa syntaxe est `do {...} while (condition);`
- `for` permet d'ajouter une incrémentation automatique et de parcourir des tableaux de données notamment. Elle s'écrit sous la forme `for (variable = valeur; condition; incrémentation) {...}`.

Hormis ces grandes règles de programmation, la force de JavaScript et des bibliothèques telles que jQuery ou Prototype (qui facilitent la vie des développeurs) est de pouvoir interagir avec les éléments HTML ou avec les propriétés CSS pour créer des animations ou modifier des éléments. Nous croiserons d'ailleurs quelques scripts au cours de notre lecture qui iront dans ce sens.

La fonction `getElementById();` résume la méthode la plus courante pour capter un contenu selon son identifiant unique. Voici un exemple de code en HTML et JavaScript :

```
<div id="bloc1">Texte écrit en vert avec CSS</div>
<a href="#" onclick="document.getElementById('bloc1').style.color = blue ;">Changer la
couleur du texte en bleu</a>
```

jQuery et les autres bibliothèques JavaScript élargissent encore davantage les possibilités tout en simplifiant l'écriture du code pour la majorité des modifications courantes. Par exemple, le changement de couleur s'écrirait ainsi en jQuery :

```
$("#bloc1").css('color','blue');
```

Enfin, le dernier concept à connaître avant de continuer notre parcours rapide des langages web est celui des fonctions. En programmation, tout ou presque n'est que fonction, c'est-à-dire des mini programmes qui permettent de gagner du temps et d'éviter de programmer plusieurs fois les mêmes scripts. La syntaxe est toujours de la forme suivante :

```
function NomDeLaFonction(arguments-optionnels) {
// code de la fonction
}
```

Il ne s'agit que de bases à approfondir mais elles vous donnent la teneur de la programmation en JavaScript natif ou à l'aide des bibliothèques (notamment jQuery). Terminons notre rapide introduction par PHP.

Bases de PHP

Le langage PHP est l'un des plus répandus sur la Toile tant il est puissant et permet d'effectuer une multitude de tâches côté serveur. Nous n'allons pas beaucoup développer ce langage puisque les mêmes principes que ceux cités précédemment s'y retrouvent, à savoir les variables, les boucles, les conditions mais aussi la création des fonctions.

Baliser les codes PHP

Les codes PHP sont toujours encadrés par des balises `<?php ... ?>` et se repèrent très vite dans le code.

En réalité, la différence majeure de syntaxe entre JavaScript et PHP réside dans les variables qui ne se basent pas sur le mot-clé `var` mais tout simplement avec le signe du dollar `$`. Ainsi, il suffit d'écrire `$uneVariable` pour créer une variable de notre choix. Pour le reste, les boucles conservent la même syntaxe, tout comme les systèmes conditionnels et les fonctions.

Avantage des fonctions en PHP

Les fonctions PHP présentent un avantage car elles peuvent accueillir des valeurs par défaut dans les paramètres de fonctions, ce qui s'avère très pratique.

Un autre aspect majeur de PHP concerne le traitement des formulaires puisque nous utilisons sans cesse des champs ou des zones de texte pour interagir avec les utilisateurs (commentaires, formulaires de contact, backoffice…), c'est pourquoi une bonne maîtrise des types de données et de leur traitement est primordiale.

Figure I-8

Exemple de méthode GET avec le paramètre « s »

Pour récupérer les informations issues des champs de formulaire, PHP met à disposition deux variables dites « super-globales » : `$_GET['name']` et `$_POST['name']`. Entre les crochets, il suffit d'indiquer la valeur de l'attribut `name` des champs de formulaire en HTML pour récupérer la valeur (`value`). Le choix entre `GET` et `POST` correspond à l'attribut `method` précisé dans la balise `<form>` en HTML.

Différences entre GET et POST

`GET` fait passer des paramètres dans les URI tandis que `POST` dissimule les informations. Nous préférons `GET` pour tout ce qui contient une pagination en général (galeries, moteur de recherche…) et la méthode `POST` lorsque nous envoyons des messages ou que nous passons des données sensibles (module d'identification avec mot de passe, formulaire de contact…).

```
<form method="get" action="">
    <input type="text" name="recherche" />
    <input type="submit" name="bouton" value="Rechercher" />
</form>
<?php
// Si le bouton est cliqué
if(isset($_GET['bouton'])) {
    // On affiche la requête saisie
    echo "Requête de recherche : ".$_GET['recherche'];
}
?>
```

Les formulaires présentent de nombreux risques en matière de sécurité. Il faut veiller à bien protéger les variables et leurs données dans la grande majorité des cas avec des fonctions spécifiques telles que `htmlspecialchars()`, `isnumeric()`, `addslashes()`, etc.

Enfin, terminons notre tour d'horizon des concepts avec celui des tableaux, disponibles aussi bien en PHP qu'en JavaScript, par exemple. Les tableaux jouent un rôle majeur car nous les utilisons sans cesse, notamment lorsque nous parcourons des bases de données.

En PHP, nous créons les tableaux « scalaires » en faisant appel à la fonction `array()`. Les items des tableaux sont séparés par des virgules comme dans l'exemple suivant :

```
$tableau = array('item1', 'item2', 'item3', '…');
```

Et les tableaux en JavaScript ?

En JavaScript, le principe est identique mais nous devons ajouter le mot-clé `new` pour créer l'objet `tableau` comme dans `var tableau = new Array('item1', 'item2…');`

Lorsqu'il s'agit de tableaux simples comme celui présenté ici, nous faisons appel aux valeurs en les ciblant à l'aide de leur « identifiant » dans le tableau. Le premier item est toujours « 0 », le second « 1 », et ainsi de suite. Si nous voulons récupérer la valeur `"item2"`, par exemple, le fait qu'elle soit en seconde position dans le tableau implique que nous devrons écrire ce qui suit :

```
echo $tableau[1];
```

Il existe également des tableaux dits « associatifs » qui permettent de personnaliser les clés reliées à des valeurs, plutôt que d'être limité à des numéros sans lien logique avec les données. Dans ce cas, les items sont toujours séparés par des virgules, mais les clés et les valeurs sont distinguées par une « flèche », comme ici :

```
$tableau = array('cle1'=>'valeur1', 'cle2'=>'valeur2', '…'=>'…');
```

Les bases de données ou tableaux complexes se présentent comme des tableaux placés dans des tableaux. Ils sont appelés « multidimensionnels » et peuvent avoir des tailles démesurées.

```
$tableaumulti = array(
    0 => array('nom'=>'Martin', 'prenom'=>'Alexandra'),
    1 => array('nom'=>'Chartier', 'prenom'=>'Mathieu'),
    2 => array('nom'=>'Andrieu', 'prenom'=>'Olivier')
);
echo $tableaumulti[0]['prenom']; // Affiche Alexandra
```

Enfin, sachez qu'il existe une boucle particulière (`foreach`) qui permet de parcourir les tableaux rapidement. Elle se traduit par : « pour chaque item du tableau, parcourir les données ». Sa syntaxe est un peu particulière, comme le montre les exemples suivants :

```
$tab = array('nom'=>'Chartier', 'prenom'=>'Mathieu');
// Syntaxe : foreach($nom-tableau as $valeur-tableau) { … }
foreach($tab as $valeur) {
    echo $valeur; // Affiche "ChartierMathieu"
}
// 2e écriture : foreach($nom-tab as $cle-tab => $valeur-tab) { … }
foreach($tab as $cle => $valeur) {
    echo $cle; // Affiche "nom" puis "prenom"
    echo " : "; // Sépare les clés des valeurs par " : "
    echo $valeur; // Affiche "Chartier" puis "Mathieu"
}
```

Nous n'avons pas pu traiter tous les concepts intéressants de PHP ou de JavaScript tels que la concaténation, l'arithmétique ou encore la gestion des objets mais nous vous invitons à vraiment intégrer ces concepts pour progresser. Il est fortement conseillé de se référer à la documentation officielle de PHP pour aller plus loin avec ce langage (http://www.php.net/manual/fr/).

Conclusion sur la programmation

D'autres langages peuvent être croisés lorsque nous faisons du Web tels que le Python avec son écriture très segmentée et imbriquée, ses `if… elif…` ou encore ses tuples et listes de données. Java est un langage orienté objet courant et très typé également, il peut se croiser de temps en temps, tout comme les technologies ASP voire ASP.Net si vous travaillez avec des environnements Microsoft.

Nous ne pouvons pas traiter de tous ces langages en quelques lignes, ce n'est pas le sujet du livre. Comme nous l'avons dit au début de ce chapitre, l'objectif est d'utiliser les forces de la programmation au profit du référencement, et bien que cela demande une mise à niveau ou quelques rappels, chacun sera libre d'approfondir les langages cités précédemment.

Retenons que nous devrons maîtriser HTML pour gérer notre référencement et que tous les autres langages qui gravitent autour de lui pourrons avoir un impact indirect sur notre travail, soit en améliorant l'ergonomie générale des sites web avec CSS, JavaScript ou même PHP, soit en nous permettant de créer nos propres outils pour automatiser des tâches et gagner en cadence. Une fois les techniques acquises, nos seules réelles limites sont notre imagination et notre créativité, sans quoi nous pourrions développer une multitude d'outils utiles pour les référenceurs.

Enfin, n'oublions pas non plus la suite Office de Microsoft et VBA (ou VBE pour Excel) qui peuvent aussi nous permettre de créer des macros puissantes pour classer et administrer des données ou encore générer des rapports, bien que cela soit indépendant du Web mais aide surtout le travail en amont et en aval du référenceur.

Nous allons désormais étudier plusieurs sujets connus dans le milieu SEO en allant dans les détails et en tentant de créer des programmes clés en main (dont certains peuvent être améliorés) pour nous aider dans nos tâches rébarbatives ou complexes.

1

Maîtriser les techniques d'indexation

Rappel des fondamentaux

L'indexation est une composante fondamentale du référencement dont l'objectif est de faciliter l'enregistrement des pages dans les bases de données des moteurs. Souvent, les gestionnaires de sites ont tendance à optimiser le positionnement des pages avant de penser à les indexer, ne tombons pas dans cette mauvaise stratégie.

Pour rappel, les principales étapes successives à respecter dans une stratégie SEO sont les suivantes.

1. Analyse concurrentielle et de la faisabilité du marché : elle consiste à vérifier les sites concurrents existants sur le marché, à étudier leur stratégie de communication et à mesurer les capacités à lutter dans le même secteur d'activités.
2. Préparation du référencement : recherche d'expressions et de mots-clés, analyse des termes usités par les concurrents, étude de la longue traîne et de faisabilité sur les mots-clés sélectionnés.
3. Amélioration de l'indexation : optimisation des pages et de certains facteurs pour faire en sorte que les moteurs de recherche indexent le plus possible de pages du site web.
4. Optimisation du positionnement : amélioration des contenus, des codes sources et développements techniques spécifiques (PageSpeed…). Nous traiterons ces points dans le prochain chapitre.
5. Audit et suivi des efforts consentis : ces étapes permettent de jauger la qualité du travail réalisé mais aussi de prévoir des ajustements pour améliorer encore les résultats. Il s'agit de phases majeures pour les référenceurs car une fois le gros du travail effectué, c'est le suivi continuel qui permet d'optimiser

encore davantage l'indexation et le classement des pages web. Cette étape permet aussi de réaliser des analyses statistiques et de calculer des retours sur objectifs ou sur investissements.

Dans ce chapitre, nous détaillerons quelques méthodes d'indexation parmi les plus connues afin d'optimiser ce maillon essentiel dans la chaîne du référencement. Toutes les techniques ne seront pas présentées mais parfois uniquement rappelées par commodité. N'hésitez pas à vous référer à d'autres ouvrages pour en savoir plus sur le sujet si vous manquez de connaissances.

Rôle et importance de l'ergonomie

Qualité du code pour le crawl

L'indexation des pages web n'est qu'une histoire de crawl, c'est-à-dire de parcours des codes sources par les robots. En effet, les spiders (ou crawlers, bots, robots...) scrutent les pages, récupèrent dynamiquement les contenus ainsi que les liens internes et externes, puis suivent ces connexions pour passer de sites en sites.

Le traitement des contenus leur permet de calculer la pertinence des pages et de savoir s'ils doivent les conserver ou non dans l'index final. Néanmoins, c'est essentiellement le maillage interne (les liens) qui les intéressent car c'est ainsi qu'ils peuvent naviguer de sites en sites et trouver sans cesse de nouvelles données à indexer.

Les spécialistes considèrent souvent que la qualité du code source des pages n'est pas primordiale car elle ne permet pas d'être mieux positionné dans les pages de résultats. Certes, il ne s'agit pas d'un critère de valeur à part entière en matière de positionnement mais son rôle est en revanche indispensable pour l'indexation des pages.

Un robot n'aime pas être freiné dans sa course, il aime les pages claires, structurées et bien conçues afin de pouvoir crawler avec aisance et trouver de nouvelles pages. Il persiste des langages et des facteurs bloquants qui peuvent empêcher partiellement voire totalement le parcours des robots, ce qui constitue un véritable drame dans une phase d'indexation car les pages concernées seront pour la majorité ignorées et non retenues.

Nous reviendrons sur les facteurs bloquants dans le troisième chapitre mais retenez bien qu'un code propre, ergonomique et structuré est le meilleur moyen d'être apprécié par les robots...

Gérer les URL et le Bot Herding

Nous avons évoqué précédemment la qualité des codes sources, il va de soi que la gestion des URL constitue un maillon majeur de la chaîne pour aider les spiders à passer de page en page. De nos jours, les spécialistes tentent même d'attirer les robots dans un sens de circulation jugé adéquat pour optimiser l'indexation, cette méthode d'appât s'appelle le *Bot Herding*[1], voir chapitre 2, section « Optimiser le Rank Sculpting et le Bot Herding ».

1 Le Bot Herding est une technique qui permet aux webmasters de mieux contrôler le parcours des robots à l'intérieur d'un site web.

Retenez qu'il est essentiel de construire des pages hiérarchisées et structurées, c'est-à-dire avec des menus lisibles par les moteurs de recherche et une bonne gestion des niveaux de profondeur du site. S'il existe des impasses dans un site, le robot ne peut plus effectuer son travail d'indexation et n'apprécie guère d'être stoppé dans sa démarche, il convient donc de prévoir des échappatoires dans chaque page pour rediriger le robot à notre guise vers les pages issues des mêmes thématiques ou importantes à nos yeux.

Le Bot Herding ainsi que la gestion des URL passent par un plan du site dessiné et détaillé pour voir comment réaliser la structure la plus ergonomique et efficiente possible. Si le squelette du site est adapté à un humain et facilite la navigation des visiteurs, dites-vous qu'il en sera de même pour les robots d'indexation...

Il n'existe pas de méthodes miracles pour faire des sites structurés. À la fin de ce livre, nous reviendrons en détail sur les points à respecter dans l'audit SEO, ce qui vous permettra d'envisager plusieurs possibilités. Mais pour ne pas entretenir trop de suspense, voici quelques rappels intéressants :

- réaliser des menus non bloquants avec des liens classiques en HTML est la meilleure solution pour permettre le crawl ;
- ajouter un plan du site constitué de nombreux liens vers les pages internes majeures (voire toutes les pages si le site n'est pas de trop grande envergure) facilite le crawl des robots lorsqu'ils découvrent cette page (il s'agit un peu de leur Saint Graal tant nous leur donnons de quoi manger...) ;
- utiliser des systèmes de tags optimisés (nuages de tags, hashtags...) permet de faciliter l'indexation mais aussi d'améliorer le positionnement. Nous étudierons certaines méthodes dans le prochain chapitre ;
- insérer un fil d'Ariane dans les pages web permet aux visiteurs de mieux se situer dans le site mais aussi aux robots d'avoir toujours des liens à parcourir pour rebondir de pages en pages. Un fil d'Ariane s'impose presque lorsqu'il s'agit de sites profonds et peut vraiment aider à l'indexation pour les moteurs de recherche ;
- installer des flux RSS ou Atom dans le site web afin de permettre un crawl de ces fichiers qui redirigent en général vers une dizaine d'articles ou de contenus. Comme ces flux de syndication se mettent à jour au fur et à mesure que des nouvelles actualités et de nouveaux articles sont publiés, les robots ont toujours des liens à suivre ;
- éviter d'utiliser trop de redirections ou tout du moins de mauvaises redirections entre les pages, cela peut engendrer des pénalités mais aussi freiner voire perdre les robots si elles ne sont pas bien maîtrisées.

Méthodes d'indexation

Comme ce chapitre ne va pas rentrer dans le fond du sujet de tous les facteurs d'optimisation de l'indexation et que nous ne voulons pas vous laisser comme des âmes en peine, voici un rapide rappel des méthodes diverses et variées qui nous permettent de mieux enregistrer et afficher les pages web dans les index des moteurs.

- Créer un site structuré et ergonomique pour faciliter le crawl et permettre aux moteurs d'attribuer une pertinence maximale aux pages. Cette étape constitue un avantage pour les divers visiteurs et

utilisateurs du site, son impact est donc double. Un bon maillage interne rend l'indexation plus simple et assure de bien meilleurs résultats, à condition d'éviter les facteurs ou langages bloquants (traités ultérieurement, voir chapitre 3).

- Obtenir un maximum de liens entrants de pages déjà indexées. Comme les bots scrutent les sites web de pages en pages, le fait d'obtenir des liens de sites déjà connus permet de se faire remarquer plus rapidement. La méthode est surtout intéressante quand il s'agit d'un site jeune car elle permet de faire connaître les pages aux divers crawlers du Web en peu de temps.
- Soumettre les nouvelles pages (pas nécessairement toutes) dans les formulaires d'indexation des moteurs de recherche. En effet, certains outils présentent des formulaires dédiés à l'indexation pour proposer des pages à crawler et indexer. Google, Bing et consorts proposent ce type de service, cela peut être intéressant pour les nouveaux sites web afin de se faire connaître, bien que la technique ne soit pas une garantie en termes de résultats.
- Utiliser des flux de syndication (RSS ou Atom) ainsi que des parseurs (ou *scrapers*) pour proposer des portions dynamiques dans les pages web et fournir de nouveaux liens à suivre. Pour faciliter l'indexation, l'idéal est d'utiliser des annuaires ou des agrégateurs afin de se faire repérer plus rapidement par les robots. Si possible, n'hésitez pas à utiliser le protocole Push (PubSubHubbub) pour accélérer l'indexation des articles et actualités sur Google (implanté par défaut dans WordPress...).
- Réaliser des fichiers `sitemap.xml` et les indiquer aux divers moteurs de recherche pour les inciter à crawler de nombreuses pages et surtout en retenir un maximum dans l'index. Il s'agit de la meilleure alternative pour implanter les pages dans les moteurs de recherche, nous la détaillerons dans ce chapitre...
- Créer un fichier `robots.txt` pour choisir les pages à indexer ou non. Ce fichier a pour objectif de limiter l'indexation et d'éviter que des pages non souhaitées apparaissent dans les résultats de recherche (SERP). Cette méthode sera détaillée dans la section « Créer un fichier robots.txt » de ce chapitre.
- Optimiser l'affichage des pages web grâce aux extraits enrichis *(rich snippets)* afin d'occuper davantage d'espace dans les pages de résultats. Cette étape est souvent négligée par manque de technique ou de temps mais si nous le faisons, nous pouvons nettement améliorer la visibilité de nos pages au sein des SERP et améliorer le taux de clics.

Les moteurs gardent la mainmise sur l'indexation

N'oublions pas que les moteurs de recherche demeurent les seuls décideurs en matière d'indexation, ce qui signifie que les robots peuvent très bien crawler des pages sans jamais les afficher dans les SERP. Il faut donc faire le maximum pour que beaucoup de pages soient retenues mais rien ne garantit une totale indexation des sites web...

Maîtriser l'évolution de l'indexation

Google Caffeine, MayDay et Bing Catapult

Nous avons rapidement évoqué Google Caffeine et MayDay dans l'introduction. Ces deux mises à jour ont considérablement modifié l'indexation et la lecture des pages web par Googlebot, au même titre de Bing Catapult pour le moteur de Microsoft.

Google MayDay a été le premier changement marquant dans la compréhension de pages web et de leurs contenus. D'autres algorithmes avaient amélioré ces points précis, mais cela faisait de nombreuses années que l'infrastructure de Google stagnait en la matière. MayDay, dont le nom provient certainement du mois d'officialisation (mai 2010), a permis de mieux comprendre les requêtes issues de la longue traîne (requêtes à rallonge de plus de 3 voire 4 mots-clés).

Cela a affecté essentiellement le classement des pages, mais MayDay agissait en réalité dès l'indexation pour mieux comprendre les contenus page par page. Ce point est important car, auparavant, Google avait tendance à analyser les pages par « groupes » ou par ensembles. Les analyses individuelles ont commencé avec Google MayDay, très liée à la sortie de Google Caffeine dans la foulée.

Google Caffeine est une infrastructure née en 2009, mais qui n'a été officialisée dans le monde que le 8 juin 2010 (source : https://goo.gl/jzeJLD). Toute la technique d'indexation du moteur de recherche a été revue et corrigée, tant sur le plan des codes sources que sur l'infrastructure matérielle. Google a indiqué à l'époque que le rafraîchissement de l'index serait amélioré d'environ 50 % par rapport à l'ancienne structure. Caffeine a permis à Google de mieux indexer les ressources web (pages, images, vidéos, PDF...) et surtout beaucoup plus rapidement (de l'ordre de quelques secondes pour le *Minty Fresh Indexing* notamment).

Bing Catapult est moins connue que Google Caffeine mais reprend les grandes lignes de son infrastructure. Il faut dire que la communication de Microsoft est souvent discrète et que l'annonce n'a pas été une révolution le 16 juin 2014 (source : http://goo.gl/Azryvc), plusieurs années après la nouveauté de Google. Cependant, Bing Catapult a vraiment changé en profondeur l'indexation menée par Bingbot puisque ce ne sont pas seulement des programmes qui ont amélioré le crawl des robots, mais aussi l'infrastructure technique et matérielle *(hardware)*. Cela a permis au moteur de recherche de multiplier par deux voire par trois sa capacité et sa vitesse d'indexation.

Google Hummingbird

Hummingbird (traduit par « colibri » en français) est l'un des derniers algorithmes de pertinence activés par Google pour améliorer les résultats organiques, principalement sur les requêtes larges et complexes de type conversationnel. La mise à jour a été activée fin août 2013, mais annoncée seulement le 26 septembre. La mise à jour Google Hummingbird a été déployée dans le monde entier et pourrait toucher une majorité des requêtes à l'avenir.

La légende veut que Google ait choisi ce nom en référence à la précision et la rapidité d'un colibri. L'algorithme Hummingbird est donc rapide et précis comme l'oiseau, selon Google, et améliore les résultats de recherche.

Cet algorithme cible une meilleure compréhension des requêtes des utilisateurs. Il ne s'agit pas vraiment d'un algorithme visant à améliorer l'indexation ni même le positionnement des pages web ; il a pour rôle de mieux associer les pages web indexées avec des requêtes dites « conversationnelles », c'est-à-dire en langage naturel.

Prenons un exemple concret : quand un internaute recherchait « Qu'est-ce qu'un moteur de recherche ? » dans Google, le moteur découpait la requête en mots séparés (ou expressions connues) et proposait les résultats répondant à l'ensemble de ces termes clés selon le ranking correspondant. Dorénavant, Google analyse la requête, détecte qu'il s'agit d'une question et en détermine le sens. Par conséquent, il comprend que la partie « qu'est-ce que » n'est pas un mot-clé mais juste la question « quoi ? » à laquelle il va devoir associer l'analyse sémantique de l'expression « moteur de recherche ». Pour simplifier, Google déduit qu'il s'agit d'une question à laquelle il doit répondre, et non d'une simple requête composé d'un enchaînement de termes.

Bon anniversaire Google

Google s'est offert Hummingbird pour ses 15 ans. La mise à jour à été annoncée par Amit Singhal (vice-président senior du Search) lors d'une conférence de presse pour l'anniversaire du moteur. Un cadeau plutôt original pour le géant américain.

Ici, il ne s'agit pas de pénaliser les sites mais bien d'une mise à jour axée autour de la recherche sémantique pour améliorer les résultats fournis par le moteur, surtout lorsqu'un internaute tape une requête en « langage naturel » *(conversational search)*, c'est-à-dire en posant des questions ou en faisant des phrases dans le champ de recherche.

Bien que cet algorithme ne concerne pas des pénalités, il est à noter que Google Panda et Penguin existent toujours et n'ont pas été retirés du moteur. En réalité, le nouvel algorithme prend lui-même en compte des critères analysés par les deux mises à jour puisqu'il se situe à un niveau supérieur dans la hiérarchie de l'algorithme. Hummingbird est le plus gros changement technique que Google ait subi après l'infrastructure Caffeine en 2009, à laquelle il peut être apparenté sur certains points.

Un pas vers la recherche sémantique d'envergure ?

Hummingbird constitue une révolution dans le monde du Web car, pour la première fois, un moteur de recherche comprend des phrases humaines (même les plus complexes parfois) et tente de retourner les réponses les plus qualitatives. Après la mise en œuvre du Knowledge Graph, cette mise à jour est la preuve qu'une forme de recherche sémantique est en place sur Google.

Comme nous l'avons évoqué, Hummingbird n'est pas un filtre ou un algorithme pénalisant, mais sa liaison étroite avec l'algorithme principal et Google Panda nous incite à nous méfier encore plus qu'auparavant. En effet, si Google était déjà capable de repérer des contenus de mauvaise qualité, cela sera encore plus vrai à l'avenir.

Partant de ce postulat, plusieurs solutions doivent être envisagées par le référenceur pour améliorer ses contenus et tenter de répondre aux requêtes visées par Hummingbird.

- Travailler sa longue traîne avec précision. La recherche par mot-clé unique est quasiment finie, l'internaute tapant désormais des expressions de 3 à 4 mots, voire plus, ou de vraies questions.
- Travailler l'univers sémantique des contenus (synonymes, hyperonymes…) car la recherche va être de plus en plus souvent qualifiée sémantiquement.

- Envisager la création d'articles qui répondent à d'éventuelles questions. Comment faire ceci ? Pourquoi faire cela ? Quelle est la méthode pour faire ceci ? Ne négligeons jamais que les internautes ont souvent des intentions variées lors des recherches. Ils peuvent aussi bien chercher une information, un contenu, un produit, un service, un lieu... et tout cela passe par des requêtes textuelles simples ou par des questions. Notre objectif est d'être là pour répondre à leurs interrogations…
- Introduire les recherches conversationnelles au sein des mots-clés, c'est-à-dire des parties textuelles qui pourraient coller parfaitement avec des questions posées ou des requêtes tapées.
- Orienter les contenus vers leur cible principale, à savoir l'utilisateur, et non pas vers le moteur de recherche, tout cela en soignant encore plus les contenus et en les rendant riches, pertinents, variés et de qualité.
- Songer à mettre à jour fréquemment les contenus pour toujours répondre à l'actualité et aux éventuelles nouvelles questions.
- Accorder de l'importance à la recherche universelle par l'inclusion de vidéos, photos, cartes... au sein des contenus ; cela augmente la pertinence des pages mais permet aussi de répondre à d'autres types de demandes.
- User des extraits de code enrichis pour ressortir sur des requêtes de type conversationnel (auteurs, avis, produits, événements, lieux…) comme « Quel est le prix d'une tablette tactile ? ».

Hummingbird tient-il compte des critères sociaux ?

Selon plusieurs sources, il paraîtrait que Google Hummingbird permet enfin au moteur de recherche de prendre en compte des critères issus des réseaux sociaux pour l'établissement de son classement. Toutefois, cette information est à prendre au conditionnel tant les études et avis divergent à ce sujet.

En réalité, il semblerait que la firme accorde de l'importance à son réseau social Google+ mais très peu aux autres concurrents tels que Twitter ou Facebook. Pour ce faire, un algorithme basé sur l'AuthorShip ou l'AuthorRank semble bien plus plausible. Lorsque nous manquons de données, il arrive souvent que des rumeurs soient lancées et se propagent sur la Toile. Cela semble être le cas une nouvelle fois par manque de communication de la part de Google…

La promesse de Google avec cet algorithme est de proposer des SERP encore plus propres et pertinentes, sans liens farfelus ou sans rapport avec la recherche initiale de l'internaute. Désormais, Google développe la recherche sémantique et la compréhension des chaînes de caractères (nous parlons de phrases, mais un robot reste un robot et ne remplacera pas un humain) au point d'offrir des résultats de plus en plus aboutis pour les internautes.

Depuis le 16 novembre 2015, Google a annoncé de fortes améliorations de la recherche sémantique et de la compréhension des requêtes en langage naturel (source : http://goo.gl/yK9YGo). La première phase de développement de Google Hummingbird s'intéressait essentiellement aux tournures de phrases et questions simples. Maintenant, l'algorithme est plus puissant et comprend mieux les superlatifs, les questions contenant des données chronologiques ou encore des questions plus complexes (avec des combinaisons de mots-clés). Ces progrès se multiplient et signifient que la recherche sémantique ne va faire que s'améliorer dans les années à venir, notamment autour d'algorithmes comme Hummingbird ou RankBrain (intelligence artificielle de Google).

Nous pouvons considérer que des moteurs performants tels que Google et Bing sont assez puissants pour lire et comprendre les textes, ce qui limite encore plus le spam et favorise les sites dont la qualité des contenus mérite de ressortir dans les SERP. Retenons surtout que Hummingbird ne pénalise pas les sites ou les pages comme Panda ou Penguin ; il ne fait que changer l'approche de Google et des visiteurs vis-à-vis des recherches effectuées.

La peur des mises à jour…

Beaucoup de spécialistes se sont affolés à l'annonce du nouvel algorithme en pensant qu'il s'agissait d'un énième filtre ou d'une autre pénalité. Fort heureusement, l'algorithme Hummingbird devrait plutôt être considéré comme une évolution logique et utile au sein du moteur. Avec Hummingbird, Google devient encore plus intelligent et efficace ; nous ne pouvons pas nous plaindre…

RankBrain, l'intelligence artificielle et le machine learning de Google

Lors de la présentation des résultats financiers du troisième trimestre 2015, le PDG de Google, Sundar Pichai, a révélé que l'intelligence artificielle et le machine learning étaient deux points importants sur lesquels Google travaille activement. Ceci a été confirmé le 26 octobre 2015 avec l'annonce de l'algorithme RankBrain, un système intelligent utilisé pour 15 % des requêtes du moteur de recherche (source : http://goo.gl/y4lvG9).

L'objectif principal de RankBrain est de mieux comprendre les requêtes rares tapées par les internautes, en analysant plus profondément leur sémantique et leur sens. Si l'algorithme améliore les résultats pour 15 % des requêtes, c'est surtout parce qu'il s'agit d'intelligence artificielle et de machine learning (apprentissage automatique), des technologies qui travaillent automatiquement pour Google.

Un lancement de RankBrain dès 2014…

Comme pour Google Hummingbird, déployé un mois avant son officialisation, Google a affirmé que RankBrain est déjà utilisé depuis fin 2014, mais personne ne semble s'en être rendu compte. Cela peut s'expliquer parce que ces algorithmes ciblent avant tout des requêtes spécifiques et rares, voire de la longue traîne. Par conséquent, les fluctuations des résultats de recherche marquent moins les esprits que si les requêtes principales étaient affectées directement.

Greg Corrado, spécialiste du machine learning, a fait partie de l'équipe de développement de RankBrain (la Google Brain team), ce qui explique l'évolution automatique du système via une intelligence artificielle progressive. Il s'agit certainement du premier système de cette envergure mis en place dans un moteur de recherche.

Du machine learning open source chez Google

Le 9 novembre 2015, Google a rendu open source sa technologie TensorFlow, une bibliothèque complète développée en Python et C++ consacrée entièrement à l'intelligence artificielle et au machine learning. Cette annonce révèle les progrès de Google en la matière (TensorFlow semble être son ancienne technologie), mais surtout sa volonté de propager l'usage de l'apprentissage automatique et de l'intelligence artificielle partout dans le monde. RankBrain est donc un énième algorithme utilisant ce type de procédés déjà exploités par Google depuis quelques années.

Sur le plan technique, RankBrain n'est pas si simple à comprendre. L'algorithme est censé représenter les concepts repérés dans les textes (parmi les mots et expressions) en « vecteurs », notés automatiquement par le système selon leur importance. Google utilise notamment des logiciels comme Word2Vec (source : https://goo.gl/F9Q3Td) pour transformer automatiquement les mots et entités en vecteurs, analysés et notés par un réseau de neurones artificiels.

Tout est une question de contexte avec RankBrain. Un terme est transformé en vecteur puis analysé en fonction de son contexte sémantique. Ce sont donc les relations entre les mots (comme les co-occurrences souvent évoquées par les spécialistes des cocons sémantiques) et leur probabilité de se retrouver proches d'autres mots qui vont permettre à l'algorithme de déterminer leur valeur. Il s'agit en réalité d'analyser les vecteurs pour comparer leurs propriétés statistiques, et donc leurs « équivalences ».

Google fournit un exemple avec des noms de pays dans sa documentation pour Word2Vec. Il indique par exemple que le vecteur le plus proche du mot France est « Spain » en langue anglaise, comme le montre la capture suivante avec la notion de « distance » entre les vecteurs.

Figure 1-1

Distance entre les vecteurs

A simple way to investigate the learned representations is to find the closest words for a user-specified word. The *distance* tool serves that purpose. For example, if you enter 'france', *distance* will display the most similar words and their distances to 'france', which should look like:

Word	Cosine distance
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

La force de RankBrain est son intelligence artificielle mais aussi, comme nous l'avons indiqué, le machine learning. En effet, une fois que Google arrive à déterminer la distance entre les mots avec son analyse statistique des vecteurs, il est capable d'aller plus loin automatiquement en comprenant des expressions, puis des portions de textes. La conséquence est simple : Google devrait être capable rapidement de déterminer si des phrases ont le même sens, voire si l'orthographe et la grammaire de ces dernières sont bien respectées.

Il s'agit donc d'une réelle évolution pour la compréhension des requêtes des internautes, mais aussi des contenus des pages web. C'est bien toutes les indexations et les recherches qui risquent d'être affectées à terme par ce type d'algorithmes aussi puissants qu'évolutifs. Nous pouvons donc considérer RankBrain comme une énorme amélioration de Google Hummingbird, avant de nouveaux systèmes plus évolués à l'avenir…

Vers une recherche prédictive ?

Parallèlement à RankBrain et au développement de l'apprentissage automatique, il faut savoir que Google réfléchit également à mettre en œuvre de la recherche prédictive, essentiellement fondée sur une personnalisation des résultats de recherche en fonction de sa bonne connaissance des internautes. De nombreux brevets ont déjà été déposés à ce sujet (source : http://goo.gl/41RDy9).

Avec une meilleure compréhension sémantique des textes, via RankBrain et Hummingbird notamment, il ne fait aucun doute que Google sera bientôt capable de mieux associer les goûts et les recherches de chaque internaute pour personnaliser les SERP lors des recherches…

Maîtriser les Sitemaps XML

Origines et usages

Le protocole Sitemap a été lancé dès 2005 par Google afin de faciliter le travail d'indexation des pages web. Très rapidement, ce projet placé sous licence libre *(Creative Commons)* a été repris par d'autres moteurs tels que Bing, Exalead, Baidu et Yandex pour ne citer qu'eux. Créer son propre fichier Sitemap présente donc un réel intérêt pour optimiser l'enregistrement des pages. Si vous souhaitez en savoir davantage, toute la documentation officielle autour du protocole Sitemap est disponible à l'adresse suivante : http://www.sitemaps.org/fr/.

Le fichier Sitemap est un document XML qui recense la totalité des pages web à indexer, URL par URL. En réalité, il s'agit davantage d'un fichier XML de définition comme nous en rencontrons parfois dans certains scripts (galeries photo…). En d'autres termes, le fichier Sitemap se comporte plutôt comme un fichier texte balisé en XML, un peu comme le sont les pages web en HTML.

L'indexation par le biais de fichiers Sitemap s'est considérablement améliorée avec le temps. Presque tous les formats peuvent être indiqués dans ces documents, que ce soit de simples pages web en passant par des images, des fichiers PDF ou encore des vidéos.

Un Sitemap assure-t-il l'indexation des pages ?

La présence et la soumission d'un fichier Sitemap ne garantit pas que toutes les pages sont enregistrées dans l'index des moteurs, il s'agit juste d'une aide très complète pour favoriser l'indexation mais aucunement pour l'obliger...

Il est important de noter que le protocole Sitemap ne s'applique pas uniquement aux fichiers XML de définition tels que nous allons les optimiser. Il est également possible d'indiquer les URL des pages web via des flux de syndication RSS et Atom pour faciliter l'indexation (source : http://goo.gl/loorr), bien que cette méthode ne permette pas d'indiquer l'ensemble des pages. Bing utilise parfois cette technique mais elle donne simplement la possibilité aux moteurs d'accéder aux liens affichés dans les flux. Son avantage est d'offrir une rapidité d'indexation grâce à la syndication.

Il se peut que des fichiers Sitemap se retrouvent eux-mêmes indexés et soient donc accessibles via les SERP. Certes, les pirates du Web n'attendent pas l'indexation pour tester la lecture d'un fichier Sitemap mais cela signifie qu'il convient de ne pas indiquer les pages à risque sous peine de divulguer des informations vulnérables pour d'éventuelles attaques. L'idéal est d'indiquer dans les interfaces pour webmasters que les fichiers Sitemap doivent être retirés des SERP, cela limite les risques de visibilité.

Quel nom donner aux fichiers ?

Le nom des fichiers Sitemap est totalement libre, nous attribuons très souvent l'intitulé `sitemap.xml` mais il peut être complètement différent. Pour des raisons de sécurité et pour tromper l'ennemi, il convient même de modifier totalement cette règle de nommage pour limiter sa lecture par des tiers.

Étapes de création

Pour créer un fichier Sitemap manuellement, suivez une méthode simple.

1. Créer les pages web et leur attribuer un nom définitif. Si vous utilisez une réécriture d'URL, il faut évidemment prendre en compte les adresses web renommées.
2. Créer un fichier Sitemap de définition (ou plusieurs si un site en nécessite davantage) avec un éditeur de texte et l'enregistrer en prenant soin de modifier l'extension en .xml.
3. Le soumettre aux moteurs concernés via les interfaces pour webmasters (disponibles sur Google, Bing, Yandex et Baidu) ou directement dans un fichier `robots.txt` comme nous le verrons dans la sous-section « Ajout de fichiers sitemap.xml ».
4. Attendre que les robots parcourent et intègrent les données du plan de site envoyé, puis indexent les pages jugées pertinentes.

Nous l'avons dit précédemment, plusieurs fichiers XML peuvent être créés conjointement lorsque les besoins s'en ressentent. Cette technique est intéressante à bien des égards, elle permet notamment de ne pas mélanger les informations propres à l'indexation des pages, des PDF, des images et vidéos, etc. Une fois ces différents fichiers créés, il suffit de relier l'ensemble au sein d'un fichier d'index qui indique le chemin menant vers chaque plan de site.

Deux règles essentielles sont à respecter dans ces fichiers XML :

- aucun d'entre eux ne doit contenir plus de 50 000 URL. Il est rare d'atteindre ce chiffre mais si tel est le cas, il convient de créer plusieurs fichiers distincts ;
- leur poids est limité à 10 Mo maximum (10 485 760 octets pour être totalement précis).

La création manuelle de fichiers Sitemap est de plus en plus rare tant les développeurs se sont habitués à utiliser des outils ou des générateurs. Mais il est primordial de savoir concevoir ce type de document, notamment si nous développons nous-mêmes notre site web et un générateur dynamique de Sitemap.

Lutter contre les surpoids…

Les fichiers Sitemap peuvent être compressés si nécessaire (au format Gzip). Dans ce cas, c'est le poids de l'archive qui est pris en compte et qui ne doit donc pas dépasser les 10 Mo. En revanche, la limitation des 50 000 URL est toujours valable.

Le site officiel *sitemaps.org* indique qu'il est autorisé de noter chaque URL ligne par ligne au sein d'un fichier texte, il s'agit alors de lister les pages à indexer.

Soumettre des fichiers Sitemap

Avant d'entrer dans le détail de la création des fichiers, nous allons voir comment soumettre le fichier aux moteurs de recherche. Il s'agit de la dernière tâche à accomplir pour que les robots prennent en compte les plans de site XML.

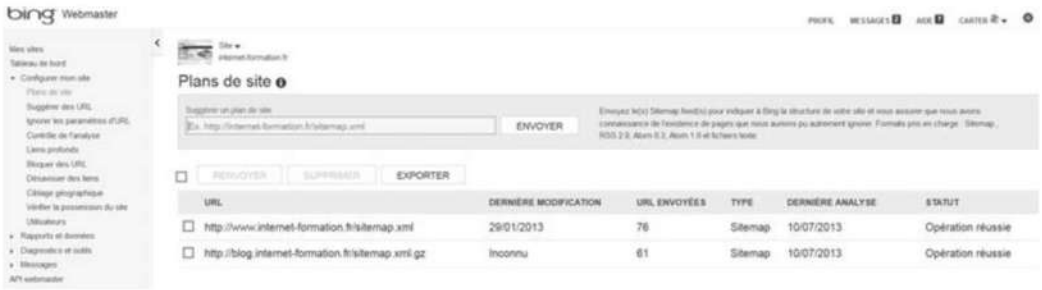
Deux solutions s'offrent à nous.

- Envoyer le fichier `sitemap.xml` ou `sitemap.xml.gz` via un client FTP tel que FileZilla, puis le soumettre à l'aide des Webmasters Tools. Cette méthode est limitée car les moteurs compatibles avec le protocole n'ont pas toujours d'interface propre aux webmasters, elle donc empêche l'optimisation de l'indexation dans ces cas précis, comme sur Exalead.
- Ajouter une ligne de code pour indiquer l'URL d'un fichier Sitemap au sein d'un fichier `robots.txt` et l'envoyer à la racine du serveur. Cette technique est conseillée car elle offre l'avantage d'être lue par tous les moteurs.

Dans un premier temps, intéressons-nous aux interfaces pour webmasters. Google, Bing, Baidu et Yandex possèdent leur propre outil, ce qui n'est pas le cas de moteurs comme Exalead, Qwant ou Ask à ce jour... Le principe est simple avec Bing Webmaster Center. Si ce n'est déjà fait, il convient tout d'abord de se créer un compte Microsoft, puis de suggérer son site à Bing via le compte Webmaster Center (http://www.bing.com/toolbox/webmaster). Ensuite, il faut cliquer sur *Configurer mon site*, puis sur *Plans de site* (ou *Sitemaps* en anglais) et indiquer l'URL complète menant vers le ou les fichier(s) Sitemap.

Figure 1-2

Ajout d'un fichier sitemap.xml avec la Bing Toolbox



Chez Google, nous devons nous rendre dans l'interface Google Search Console, puis cliquer sur *Exploration>Sitemaps* dans le menu de gauche. Le bouton *Ajouter/Tester un Sitemap* permet d'effectuer l'opération en quelques secondes. La totalité des fichiers qui ont été ajoutés pour chaque site est affichée à l'écran.

Figure 1-3

Soumettre un Sitemap dans la Google Search Console

Le même procédé s'applique avec Yandex Webmaster Tools et Baidu Webmaster Platform et suit la même logique bien que les interfaces soient moins complètes que celles de leurs concurrents.

La seconde méthode consiste à indiquer une ou plusieurs URL menant vers des fichiers Sitemap en XML ou compressés au format Gzip via un fichier `robots.txt`. Cette méthode présente l'avantage d'autoriser la lecture de ces plans de site à tous les moteurs utilisant le protocole.

Dans notre fichier, il faut ajouter la commande `Sitemap:` pour chaque URL pointant vers des plans de site. Si vous possédez plusieurs fichiers, il suffit donc de multiplier ces lignes de commande. Voici un exemple concret de fichier `robots.txt` :

```
Sitemap: http://www.test.com/plansite.xml
Sitemap: http://www.test.com/plansitevideo.xml.gz
```

Créer un Sitemap index

Si un site impose l'usage de plusieurs fichiers Sitemap, il convient de réaliser un Sitemap index regroupant les URL de chaque fichier XML. La syntaxe est relativement simple et se limite à l'ajout d'un doctype XML et à quatre balises :

- `<sitemapindex>...</sitemapindex>` encadrent l'ensemble des informations du fichier d'index, à savoir la totalité des URL ;
- `<sitemap>...</sitemap>` encadrent les données relatives à chaque fichier Sitemap ;
- `<loc>...</loc>` sont placées entre les balises `<sitemap>` et indiquent l'adresse web du fichier Sitemap ciblé ;
- `<lastmod>...</lastmod>` sont optionnelles et sont placées entre les balises `<sitemap>`. Elles indiquent la dernière date de mise à jour du Sitemap ciblé. Deux formats de dates anglaises sont autorisés : `AAAA/MM/JJ` ou `AAAA-MM-JJThh:mm:ss+GMT` (exemple : 2013-12-25T20:15:53+00:00).

Voici un exemple concret avec un fichier Sitemap compressé au format Gzip et un fichier classique en XML :

```
<?xml version="1.0" encoding="utf-8"?>
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
    <sitemap>
        <loc>http://www.test.com/sitemap-1.xml.gz</loc>
        <lastmod>2013-12-25</lastmod>
    </sitemap>
    <sitemap>
        <loc>http://www.test.com/sitemap-2.xml</loc>
    </sitemap>
</sitemapindex>
```

L'usage de multiples Sitemaps XML reliés à un index général est très intéressant en termes d'optimisation. Si cela ne change rien sur le plan de l'indexation, cela aide à mieux contrôler et suivre les pages indexées. Idéalement, il faudrait créer un fichier par page et l'ensemble serait lié à un Sitemap index (mais vous

pouvez aussi faire de petits groupes d'URL). Ainsi, il serait possible de suivre les pages indexées, dans la Search Console par exemple, car l'interface affiche pour chaque fichier le nombre d'URL recensées et le nombre d'URL indexées. Un découpage minutieux en divers Sitemaps constitue une solution astucieuse pour suivre l'indexation.

Concevoir un Sitemap XML

La création de fichiers Sitemap ressemble à peu de choses près à celle des fichiers d'index. Nous renouvellerons donc certaines pratiques pour aboutir au résultat escompté. Plusieurs étapes permettent de concevoir le fichier XML de définition :

- doctype : `<?xml version="1.0" encoding="UTF-8"?>` ;
- ajout du bloc englobant tout le Sitemap avec les balises `<urlset>…</urlset>`, sachant que la première doit recevoir l'attribut `xmlns` (pour la version du protocole) sous la forme `<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">` ;
- intégration des balises XML `<url>…</url>` utiles pour chaque page web à indexer à l'intérieur du bloc `<urlset>…</urlset>`.

L'étape majeure est celle qui consiste à fournir les informations relatives à chaque URL du site. Quatre balises peuvent s'inscrire entre les balises `<url>` et `</url>` de chaque page web :

- `<loc>…</loc>` sont les seules balises obligatoires, elles encadrent l'adresse web de la page à indexer ;
- `<lastmod>…</lastmod>` précisent la date de dernière mise à jour de la page à indexer. Comme pour le fichier d'index, la date s'inscrit au format ISO 8601, sous la forme générique inversée `AAAA-MM-JJ` ou détaillée `AAAA-MM-JJThh:mm:ss-GMT` ;
- `<changefreq>…</changefreq>` indiquent au robot des moteurs la fréquence habituelle de modification de la page à indexer. Plusieurs valeurs fixes sont proposées par les concepteurs du protocole : `always`, `hourly`, `daily`, `weekly`, `monthly`, `yearly`, `never`. Il ne s'agit que d'une indication rarement suivie par les robots, mais évitez de noter des fréquences surréalistes ou illogiques. Par exemple, une page d'actualités sera mise à jour quotidiennement (`daily`) voire plusieurs fois par jour (`hourly`) tandis qu'une page de contact sera certainement statique plusieurs mois (`monthly`) ;
- `<priority>…</priority>` précisent la priorité d'indexation pour que les robots se concentrent davantage sur ces pages ciblées. Il s'agit d'affecter une valeur décimale entre 0 et 1, sachant que 0.5 est la valeur par défaut. Attention, il faut utiliser un point et non une virgule pour les valeurs décimales. Une priorité de 0.8 s'écrit `<priority>0.8</priority>` par exemple. En général, nous attribuons la valeur 1 aux pages principales, 0.8 au second niveau d'arborescence et jusqu'à 0.4 ou 0.5 pour les pages les moins importantes comme le plan de site ou les mentions légales.

Limitation des balises <loc>

Entre les balises `<loc>` et `</loc>`, les informations ne doivent pas dépasser 2 048 signes et le protocole utilisé doit être inscrit. Ainsi, l'indexation d'une page web implique que l'adresse commence par http:// ou https:// par exemple.

Toutes les informations apportées en complément des URL absolues des pages web doivent respecter la réalité. Ne vous amusez pas à entrer des données totalement faussées, cela n'a pas d'intérêt et les robots sauront repérer ce type de procédé. Les crawlers restent les seuls décideurs à tout point de vue, que ce soit pour indexer une page ou pour respecter les conditions que nous fixons dans le fichier Sitemap, telles que la fréquence ou la priorité.Voici un exemple détaillé de fichier Sitemap (avec trois pages) :

```
<?xml version="1.0" encoding="utf-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
    <url>
        <loc>http://www.test.com</loc>
        <priority>1</priority>
        <lastmod>2013-12-25</lastmod>
        <changefreq>daily</changefreq>
    </url>
    <url>
        <loc>http://www.test.com/categorie-principale.asp</loc>
        <priority>0.8</priority>
        <changefreq>monthly</changefreq>
    </url>
    <url>
        <loc>http://www.test.com/notice-legale.asp</loc>
        <priority>0.5</priority>
        <changefreq>yearly</changefreq>
    </url>
</urlset>
```

Encodage des fichiers Sitemap

Notons que le fichier Sitemap est encodé en UTF-8 dans nos exemples, ce qui facilite la reconnaissance des caractères spéciaux tels que les accents français. Si vous optez pour un encodage en ISO-8859-1 (ou ISO-Latin-1), il faudra encoder manuellement tous les caractères spéciaux qui peuvent se trouver dans des URL dynamiques. Ainsi, l'esperluette (&) devient `&` tandis que des guillemets se transforment en `&quot;`.

Autres types de fichiers Sitemap

Le protocole Sitemap ne se limite pas seulement aux pages web classiques. En effet, nombre de formats peuvent être indexés. La liste suivante présente les formats actuellement autorisés :

- les images peuvent être ajoutées au Sitemap d'origine ou dans un fichier différent à condition de ne pas dépasser 1 000 URL par fichier ;
- les Sitemaps pour les sites mobiles sont autorisés et permettent d'indexer des URL spécifiques aux versions mobiles des sites web, comme http://m.test.com. Ces adresses peuvent s'ajouter dans le Sitemap d'origine, il suffit d'ajouter l'attribut `xmlns:mobile="http://www.google.com/schemas/sitemap-mobile/1.0"` dans la balise ouvrante `<urlset>` et d'ajouter le marqueur `<mobile:mobile/>` dans chaque bloc `<url>` qui contient une adresse vers une page web mobile ;

- les vidéos doivent être indiquées dans un fichier XML distinct et de ce fait, seuls 50 000 blocs de données sont autorisés au maximum. Seuls les formats suivants sont tolérés : `.mpg`, `.mpeg`, `.mp4`, `.m4v`, `.mov`, `.wmv`, `.asf`, `.avi`, `.ra`, `.ram`, `.rm`, `.flv`, `.swf` ;
- les URL issues de pages d'actualités peuvent être indiquées dans un fichier à part. Ce Sitemap permet notamment d'indexer les articles dans Google News si le moteur les juge pertinents ;
- les fichiers Sitemap spécifiques à des géolocalisations sont en suspens depuis des mois. Ils permettaient notamment d'intégrer des URL pointant vers des fichiers au format KML ou GeoRSS contenant des coordonnées précises.

La diversité des Sitemaps

Google lit très bien tous les formats que nous venons de citer mais c'est loin d'être le cas de tous les moteurs de recherche. Nous devons donc créer des fichiers spécifiques pour les formats de fichiers non tolérés sur Bing, Yandex, Baidu et consorts...

Si vous désirez concevoir un Sitemap personnalisé pour les images, retenez qu'il est tout à fait possible d'ajouter les images au sein du fichier recensant les pages, cette méthode est même recommandée pour exploiter pleinement le fichier. Dans ce cas, vous devez ajouter l'attribut `xmlns:image` avec la valeur `"http://www.google.com/schemas/sitemap-image/1.1"` dans la balise ouvrante `<urlset>`, en parallèle de l'autre attribut `xmlns`. Ensuite, vous devez obligatoirement ajouter deux couples de balises à l'intérieur des blocs `<url>…</url>` :

- `<image:image>…</image:image>`, elles contiennent toutes les balises relatives à l'indexation des images, nous devons avoir autant de ces blocs que nous avons d'images ;
- `<image:loc>…</image:loc>`, elles indiquent l'URL de l'image.

Quatre autres marqueurs optionnels peuvent compléter les blocs :

- `<image:caption>…</image:caption>` pour ajouter une légende ;
- `<image:geo_location>…</image:geo_location>` pour indiquer une éventuelle zone géographique relative à l'image ;
- `<image:title>…</image:title>` pour donner un titre à l'image ;
- `<image:license>…</image:license>` pour afficher les droits relatifs à l'image.

Voici un très court exemple de fichier Sitemap d'images :

```
<?xml version="1.0" encoding="utf-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
xmlns:image="http://www.google.com/schemas/sitemap-image/1.1">
    <url>
        <loc>http://www.test.com/services.php</loc>
        <image:image>
        <image:loc>http://www.test.com/image1.jpg</image:loc>
        <image:title>Titre de l'image</image:title>
        <image:caption>Légende de l'image</image:caption>
        <image:geo_location>Nantes, France</image:geo_location>
```

```
        </image:image>
    </url>
</urlset>
```

Suivre l'avancement du protocole

Chaque format possède son propre mode de fonctionnement, aussi nous devons régulièrement suivre les avancées du protocole via le site officiel ou grâce à la documentation technique fournie par Google (source : http://goo.gl/AQLFYL). Force est de constater que les fichiers XML les plus courants sont ceux qui recensent les pages web voire les Sitemaps pour les images ou les actualités, les autres formats restent encore en retrait ou utilisés sur des sites spécifiques. Par exemple, un détenteur d'une plate-forme multimédia a tout intérêt à créer des Sitemaps dédiés aux formats vidéo.

Exemples d'outils d'aide à la création de fichiers Sitemap

Les outils, extensions ou encore modules permettant de faciliter le travail des développeurs qui souhaitent référencer leur site sont légion sur le Web. Certains CMS (« Content Management System », des logiciels en ligne de gestion de pages web) bénéficient du travail des contributeurs pour proposer des extensions et modules de qualité qui créent parfaitement les fichiers Sitemap dont nous avons besoin, ce qui nous évite souvent de passer par l'étape manuelle. Toutefois, tous les sites n'ont pas cette chance et il faut parfois recourir à cette solution plus fastidieuse.

Citons quelques exemples d'extensions pour chaque CMS :

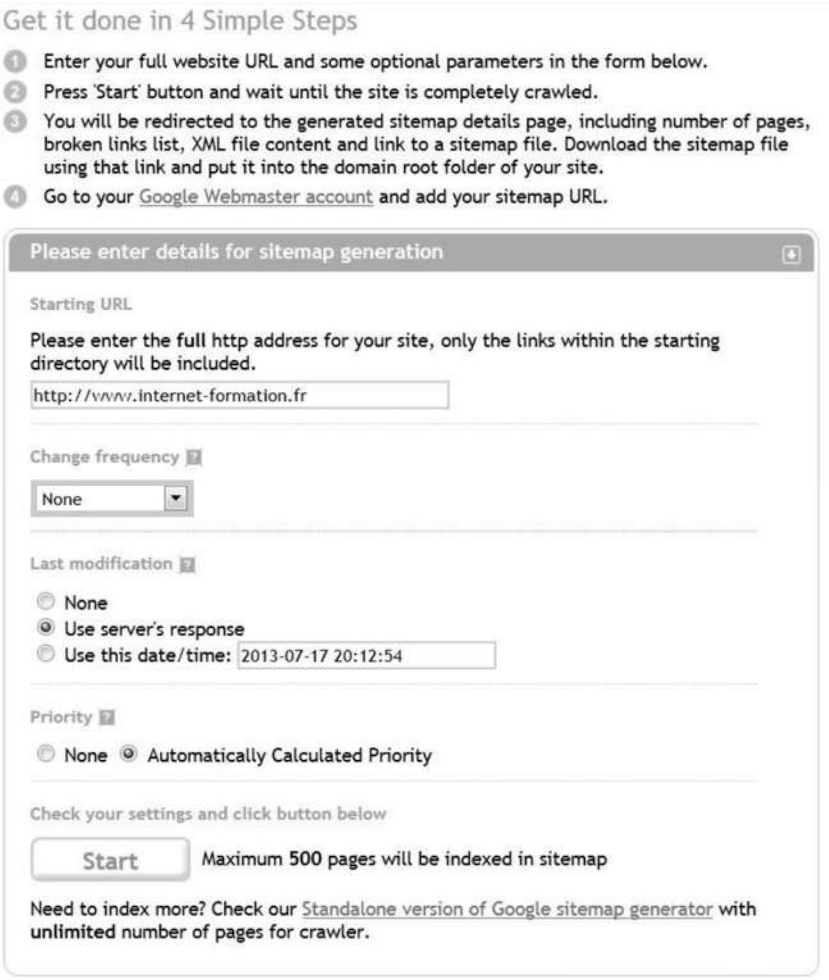
- Prestashop possède son propre générateur de fichiers Sitemap, lequel est installé par défaut dans le module appelé Google Sitemap ;
- Google XML Sitemaps (source : http://goo.gl/dr2rCo) ou encore Google Sitemap (source : http://goo.gl/tQ4qlU) pour Wordpress ;
- JCrawler (source : http://goo.gl/id2LKY) pour Joomla 1.5 et Xmap (source : http://goo.gl/wC3M4i) pour les versions récentes ;
- XML Sitemap pour Drupal ;
- SiteMap (source : http://goo.gl/WA5VMk) sur Spip ;
- Google Sitemap (source : http://goo.gl/9DPvIf) ou encore Weeaar Sitemap pour Typo3 ;
- Advance Sitemap (source : http://goo.gl/G6y7nY) ou Extended Sitemap (payant) pour Magento ;
- Dynamic Sitemap ou Google Sitemap Generator (source : http://goo.gl/WEyLtG) pour osCommerce.

Parallèlement à ces modules et extensions pour CMS, nous trouvons également des services en ligne ou des scripts pour faciliter la conception des fichiers d'indexation. Parmi eux, nous pouvons citer par exemple :

- XML-Sitemaps : http://www.xml-sitemaps.com ;
- SitemapDoc : http://www.sitemapdoc.com ;
- My Sitemap Generator : http://www.my-sitemap.com ;
- Free Sitemap Generator : http://www.freesitemapgenerator.com.

Figure 1-4

Paramétrage de l'outil XML-Sitemaps



Tous ont leurs propres qualités mais il faut admettre que la majorité des outils ne permet pas de créer de fichiers pour certains formats comme les PDF ou vidéos, ces derniers doivent souvent être écrits manuellement ou avec un code personnel. Toutefois, un service en ligne comme XML-Sitemaps est efficace pour générer les fichiers Sitemap courants car il utilise un robot pour parcourir tous les liens accessibles à partir d'une URL donnée. Ainsi, la création d'un fichier de moins de 500 URL ne prend que quelques minutes...

Figure 1-5

Génération automatique, puis téléchargement du fichier dans différents formats

Créer son propre générateur avec PHP et MySQL

Nous avons vu précédemment que les outils à notre disposition ainsi que certaines extensions ne permettaient pas de générer tous les fichiers Sitemap souhaités. Nous allons maintenant nous intéresser à la création de notre propre script en PHP pour lister toutes les pages web et documents contenus dans les sites web.

Les deux exemples de code présentés peuvent être améliorés et complétés mais l'objectif est avant tout de montrer comment mettre la technique au service du référencement, notamment pour indexer des fichiers tels que des PDF et des images. Les lignes de code qui suivent sont toutes commentées pour faciliter la lecture des scripts mais les plus néophytes pourront se sentir totalement perdus. Il est recommandé dans ce cas de se documenter à propos du couple de langages PHP/MySQL via d'autres ouvrages spécialisés.

SitemapCrawler.php

Le premier script, `SitemapCrawler.php`, ne fonctionne qu'avec un site statique sans base de données. Son rôle est de lister tous les fichiers présents dans des dossiers ciblés afin de recenser leur nom et leur adresse pour les ajouter à la volée dans un fichier Sitemap généré automatiquement.

En d'autres termes, il s'agit d'une sorte de crawler interne qui comptabilise les fichiers à indexer selon la configuration réalisée, puis le script crée le fichier Sitemap correspondant avec le nom désiré. L'avantage du script est d'être rapide et de pouvoir ajouter ou ignorer tous les fichiers qui peuvent déranger. En contrepartie, il n'est fonctionnel que pour les sites statiques dont les noms de pages sont définis par les fichiers tels que `faq.php`, `index.html`, `contact.php`, `plan-de-site.html`, etc.

Pour le bon fonctionnement du script, nous devons régler l'URL d'origine à partir de laquelle le script va se lancer, ainsi que le nom du fichier Sitemap que nous souhaitons obtenir mais aussi l'URL qui sera

affichée devant le nom des fichiers (par défaut, il s'agit de l'adresse `http` du site). Trois variables sont donc à définir :

- `$cheminBase` pour l'URL d'origine sachant qu'un point (valeur par défaut) équivaut au dossier courant dans lequel est placé le fichier `SitemapCrawler.php` (si nous le plaçons à la racine, il faut donc laisser la valeur par défaut) ;
- `$fichierSitemap` pour indiquer le nom du fichier Sitemap de destination (par défaut, `sitemapCrawler.xml`) ;
- `$URLSource` pour afficher le protocole ainsi que le nom de domaine dans le fichier XML de sortie. Nous devons veiller à placer le protocole `http` ou `https` selon les besoins.

De plus, nous devons paramétrer quelques facteurs afin de ne pas enregistrer des pages que nous ne souhaiterions pas voir indexées :

- `$extensionsJustes` est un tableau PHP qui contient toutes les extensions à lister dans le fichier Sitemap. Par exemple, si nous voulons indexer uniquement les fichiers HTML, PHP, ASP ainsi que les images en JPEG, nous entrons par défaut :

```
$extensionsJustes = array('html','php','asp','jpg','jpeg');
```

- `$dossiersJustes` fonctionne de la même manière que le paramètre précédent sauf que son rôle est d'indiquer les dossiers que nous autorisons à crawler (en plus du dossier où se situe le script). Par exemple, si nous désirons uniquement indexer les dossiers d'images, nous indiquons par défaut :

```
$dossiersJustes = array('img', 'image', 'images');
```

- `$fichiersIgnores` est également un tableau mais ce dernier a un rôle inverse puisqu'il interdit l'inscription des fichiers intégrés dans la variable. Si nous ne souhaitons pas lister les pages d'erreur, par exemple, nous pouvons noter :

```
$fichiersIgnores = array('SitemapCrawler.php', '404.php', '403.php', '500.php');
```

Une fois la configuration initiale effectuée, il ne nous reste plus qu'à envoyer le script à la racine du répertoire contenant notre site web, puis à lancer le script dans le navigateur à l'aide de la syntaxe http://www.monsite.com/SitemapCrawler.php. L'action est rapide et le navigateur affichera le nombre de fichiers ajoutés dans le fichier Sitemap XML ainsi que leur adresse propre.

La version du script dont voici le code complet génère automatiquement le Sitemap classique et celui des images, il faudrait ajouter quelques fonctions complémentaires pour aller plus loin et générer d'autres types de Sitemaps tolérés par le protocole.

```
<?php
// Dossier de départ ('.' par défaut pour la racine, './NOM-DOSSIER' pour commencer dans un
// dossier spécifique)
$cheminBase = '.';

// URL de base à afficher dans le Sitemap (sans slash final)
$URLSource = 'http://'.$_SERVER['HTTP_HOST'];
```

```
// Nom à donner au fichier Sitemap et création du fichier
$fichierSitemap = 'sitemapCrawler.xml';
$sitemapXML = fopen($fichierSitemap,"w");

// Listes des extensions autorisées, des dossiers à scanner (en plus de $chemin) et des
// fichiers à ne pas indexer
$extensionsJustes = array('php', 'jpg', 'pdf');
$dossiersJustes = array('img');
$fichiersIgnores = array('404.php', '403.php', '500.php', '.htaccess', 'robots.txt',
'analyticstracking.php');
array_push($fichiersIgnores, substr($_SERVER['PHP_SELF'],1));

// On ajoute le doctype XML ainsi que la balise ouvrante <urlset>
fputs($sitemapXML, '<?xml version="1.0" encoding="utf-8"?>'."\n");
fputs($sitemapXML, '<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
xmlns:image="http://www.google.com/schemas/sitemap-image/1.1">'."\n");

// Fonctions qui permettent de lister les fichiers d'images
function is_Images($ext) {
    $listeExt = array('jpg', 'jpeg', 'png', 'gif', 'bmp', 'svg');
    if(in_array($ext,$listeExt)) {
        return true;
    }
}
// Fonction d'affichage de l'extension des fichiers dans le résumé
function is_Format($ext) {
    $listeWebSafe = array('html', 'htm', 'php', 'asp', 'aspx');
    if(!in_array($ext,$listeWebSafe)) {
        return true;
    }
}

// On liste tous les fichiers à partir d'un dossier donné
function CrawlFichier($chemin = '.', $URLBase = '', $extensionsOK = array(), $dossiersOK =
array(), $fichiersInterdits = array()) {
// On ouvre le répertoire
$repertoire = opendir($chemin);
// On compte le nombre de fichiers ajoutés dans le Sitemap
$nb = 0; // On initialise à 0
$nb = $nb + $nb;
// Création de variables globales
global $sitemapXML, $nb;

// On boucle pour lister tous les dossiers et fichiers
while($fichier = readdir($repertoire)) {
// On récupère l'extension des fichiers listés
$extensions = strtolower(pathinfo($fichier,PATHINFO_EXTENSION));
```

```
// On exclut les répertoires inutiles './' et '../'
if($fichier != '.' && $fichier != '..' && is_dir($chemin.'/'.$fichier) && in_
array($fichier,$dossiersOK)) {

// On lance la fonction récursive jusqu'à la fin du crawl complet
CrawlFichier($chemin.'/'.$fichier, $URLBase, $extensionsOK, $dossiersOK,
$fichiersInterdits);

} elseif(in_array($extensions,$extensionsOK) && !in_array($fichier,$fichiersInterdits)) {

if($chemin == '.') {

// On ajoute les balises <url>...<url> avec <loc>URL</loc>
fputs($sitemapXML, "\t<url>\n");
fputs($sitemapXML, "\t\t<loc>".$URLBase.'/'.$fichier."</loc>\n");

if(is_Images($extensions)) {
fputs($sitemapXML, "\t\t<image:image>\n");
fputs($sitemapXML, "\t\t\t<image:loc>".$URLBase.'/'.$fichier."</image:loc>\n");
fputs($sitemapXML, "\t\t</image:image>\n");
echo "<small><em>".$URLBase.'/'.$fichier."</em> (image ".$extensions.")</small><br/>";
} elseif(is_Format($extensions)) {
echo "<small><em>".$URLBase.'/'.$fichier."</em> (fichier ".$extensions.")</small><br/>";
} else {
echo "<small><em>".$URLBase.'/'.$fichier."</em></small><br/>";
}
fputs($sitemapXML, "\t</url>\n");
$nb++; // on ajoute 1 au comptage

} elseif(!is_dir($fichier)) { // On vérifie qu'il s'agit d'un fichier
$cheminPropre = substr($chemin,1); // On nettoie l'URL de base
// On ajoute les balises <url>...<url> avec <loc>URL</loc>
fputs($sitemapXML, "\t<url>\n");
fputs($sitemapXML, "\t\t<loc>".$URLBase.$cheminPropre.'/'.$fichier."</loc>\n");

if(is_Images($extensions)) {
fputs($sitemapXML, "\t\t<image:image>\n");
fputs($sitemapXML, "\t\t\t<image:loc>".$URLBase.$cheminPropre.'/'.$fichier."
</image:loc>\n");
fputs($sitemapXML, "\t\t</image:image>\n");
echo "<small><em>".$URLBase.$cheminPropre.'/'.$fichier."</em>
(image ".$extensions.")</small><br/>";
} elseif(is_Format($extensions)) {
echo "<small><em>".$URLBase.$cheminPropre.'/'.$fichier."</em> (
fichier ".$extensions.")</small><br/>";
} else {
echo "<small><em>".$URLBase.'/'.$fichier."</em></small><br/>";
}
```

```
fputs($sitemapXML, "\t</url>\n");
$nb++; // On ajoute 1 au comptage
        }
    }
}
}

// Lancement de la fonction récursive de crawl :
CrawlFichier('CHEMIN','TABLEAU-EXTENSIONS-IGNOREES')
CrawlFichier($cheminBase, $URLSource, $extensionsJustes, $dossiersJustes, $fichiersIgnores);

// On affiche le nombre de pages inscrites dans le fichier XML
global $nb;
echo "<h2>".$nb." pages ajoutées dans ".$fichierSitemap."</h2>";

// On finalise le fichier XML en fermant la balise </urlset>
fputs($sitemapXML, '</urlset>');
?>
```

Figure 1-6

Génération automatique du fichier sitemapCrawler.xml pour un site statique

http://www.mathieu-chartier.com/activites.php
http://www.mathieu-chartier.com/index.php
http://www.mathieu-chartier.com/img/orientations-magazine-2010-mathieu-chartier.jpg (image jpg)
http://www.mathieu-chartier.com/img/Chartier-Mathieu-centre-presse-22-03-2013-referencement-web.jpg (image jpg)
http://www.mathieu-chartier.com/img/guide-du-referencement-web-chartier-mathieu-first.jpg (image jpg)
http://www.mathieu-chartier.com/img/guide-du-referencement-web-chartier-mathieu.jpg (image jpg)
http://www.mathieu-chartier.com/img/orientations-magazine-2010-interview-mathieu-chartier.jpg (image jpg)
http://www.mathieu-chartier.com/menu.php
http://www.mathieu-chartier.com/Contrat de partenariat formateur V5.pdf (fichier pdf)
http://www.mathieu-chartier.com/footer.php
http://www.mathieu-chartier.com/competences-mathieu-chartier.php
http://www.mathieu-chartier.com/contact.php
http://www.mathieu-chartier.com/experience-mathieu-chartier.php
http://www.mathieu-chartier.com/livre-mathieu-chartier.php

14 pages ajoutées dans le fichier sitemapCrawler.xml

SitemapCrawlerBDD.php

Ce même type de procédé est réalisable avec des sites dynamiques. Dans ce cas, nous devons disposer d'une base de données bien conçue qui récupère toutes les informations importantes telles que l'URL des pages et fichiers ajoutés, voire également la date de création et de modification des documents et leur type. Dans notre exemple pour `SitemapCrawlerBDD.php`, nous allons utiliser une base de données type comme celle de WordPress. Certes, le script sera adapté en partie à ce CMS mais il peut allègrement être modifié pour des sites dynamiques divers. Il était plus simple de partir d'une base de données existante et massivement utilisée. Pour le reste, le fonctionnement est identique au premier code, il suffit de placer le fichier à la racine d'un site WordPress et de le lancer via le navigateur. Plusieurs informations doivent être recueillies dans le cas d'un site dynamique sur WordPress :

- l'URL de la page d'accueil ;
- les adresses des catégories quand elles existent ;

- les adresses de chaque article affiché et chaque page publiée ;
- les URL des fichiers attachés (images et PDF notamment).

Comme précédemment, il faut configurer quelques facteurs pour rendre le script fonctionnel (les deux derniers sont optionnels) :

- les identifiants de base de données pour pouvoir se connecter ;
- l'URL de base avec le protocole `http` ou `https` ;
- le nom du fichier Sitemap de sortie ;
- les extensions des fichiers d'images autorisées ;
- les noms de tables ou les requêtes SQL si nous souhaitons utiliser le script avec WordPress ou d'autres sites dynamiques ;
- l'écriture des URL dans le fichier Sitemap et dans le rendu affiché par le navigateur.

Par défaut, WordPress a tout prévu. La table `posts` stocke toutes les données relatives aux pages et aux fichiers associés. Par exemple, les URL non réécrites sont stockées dans la colonne `guid`, la date de création dans `post_date`, le type de contenu dans `post_type` ou encore l'alias (ou *slug*) des URL dans `post_name`. L'idéal est de jeter un œil dans une base de données WordPress ou de lire la documentation officielle pour ne pas être perdu. En réalité, seule l'architecture des catégories est complexe et il faut faire de multiples jointures de tables pour récupérer les informations. Nous trouvons cette requête dans la variable `$categorieSQL` du script.

Figure 1-7

Génération du fichier sitemapCrawlerBDD.xml pour un site WordPress ou dynamique

Page d'accueil
http://blog.internet-formation.fr

Catégories
http://blog.internet-formation.fr/category/referencement/
http://blog.internet-formation.fr/category/infos-en-stock/
http://blog.internet-formation.fr/category/prospective-web/
http://blog.internet-formation.fr/category/programmation/
http://blog.internet-formation.fr/category/webmarketing/

Groupe 1
http://blog.internet-formation.fr/faire-son-fichier-sitemap/ (11/06/2009 à 11:06:27)
http://blog.internet-formation.fr/wp-content/uploads/2009/06/xml3.png (fichier lié)

Groupe 2
http://blog.internet-formation.fr/parts-de-marche-navigateurs-avril-2009/ (11/06/2009 à 11:06:31)
http://blog.internet-formation.fr/wp-content/uploads/2009/06/navigateurs-200904-1.jpg (fichier lié)
http://blog.internet-formation.fr/wp-content/uploads/2009/06/navigateurs-200904-2.jpg (fichier lié)

Groupe 3
http://blog.internet-formation.fr/barometre-des-moteurs-de-recherche-avril-2009/ (12/09/2009 à 10:09:01)
http://blog.internet-formation.fr/wp-content/uploads/2009/06/moteurs-200905-1.png (fichier lié)
http://blog.internet-formation.fr/wp-content/uploads/2009/06/moteurs-200905-2.png (fichier lié)

Architecture des URL de WordPress

Dans l'exemple du script, les URL des pages et des articles reprennent le format `%postname%` des permaliens (nom de l'article dans WordPress) tandis que les catégories se basent sur la structure *http://domaine.ext/category/slug-categorie*. Il faudra modifier les résultats selon les besoins et la base de données utilisée.

`SitemapCrawlerBDD.php` offre l'avantage d'être plus précis et dynamique que le premier code que nous avons détaillé. Il sait tirer profit des bases de données pour obtenir plus d'informations. C'est pourquoi le Sitemap XML de sortie est plus détaillé dans ce second code. Il permet notamment de classer les images en fonction de leur liaison avec une page ou un article. Il contient également des balises `<priority>` et `<lastmod>` quand cela est possible, ce qui lui confère une plus grande souplesse.

Voici le script commenté et détaillé dans son ensemble. Il peut bien sûr être complété et amélioré, mais il donne déjà une base de ce qu'il est possible de faire avec des sites conçus manuellement.

```
<?php
// Informations de connexion et d'identification
$serveur = 'localhost';
$BDD = 'crawler';
$utilisateur = 'root';
$motdepasse = '';
$serveurBDD = 'mysql:host='.$serveur.';dbname='.$BDD.'';

$connexion = new PDO($serveurBDD, $utilisateur, $motdepasse);
$connexion->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// Requête SQL à personnaliser
$tables = "wp_posts";
$requeteSQL = "SELECT * FROM $tables WHERE (post_type='page' OR post_type='post' OR
post_type='page') AND post_status='publish'";

// URL de base à afficher dans le Sitemap (sans slash à la fin)
$URLSource = 'http://'.$_SERVER['HTTP_HOST'];

// Nom du fichier Sitemap et création du fichier
$fichierSitemap = 'sitemapCrawlerBDD.xml';
$sitemapXML = fopen($fichierSitemap,"w");

// Listes des extensions d'images pour le Sitemap Images
$fichiersJoints = array('jpg', 'jpeg', 'png', 'gif', 'bmp');

// Points de départ des comptages
$nb = 1;
$tour = 1;

// On ajoute le doctype XML ainsi que la balise ouvrante <urlset>
fputs($sitemapXML, '<?xml version="1.0" encoding="utf-8"?>'."\n");
fputs($sitemapXML, '<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
xmlns:image="http://www.google.com/schemas/sitemap-image/1.1">'."\n");
```

```
// On ajoute la page d'accueil du site en début de fichier
fputs($sitemapXML, "\t<url>\n");
fputs($sitemapXML, "\t\t<loc>".$URLSource."</loc>\n");
fputs($sitemapXML, "\t\t<priority>1</priority>\n");
fputs($sitemapXML, "\t</url>\n");
echo "<strong>Page d'accueil</strong><br/>";
echo "<small><em>".$URLSource."</em></small><br/>";

// On ajoute les catégories si elles existent (requête complexe)
$categorieSQL = "SELECT * FROM wp_terms AS wterms INNER JOIN wp_term_taxonomy AS wtaxonomy ON
(wterms.term_id = wtaxonomy.term_id) WHERE wtaxonomy.taxonomy = 'category' AND wtaxonomy.
parent = 0 AND wtaxonomy.count > 0";

if(count($categorieSQL) > 0) {
    echo "<br/><strong>Catégories</strong><br/>";
    foreach($connexion->query($categorieSQL) as $categorie) {
        echo "<small><em>".$URLSource.'/category/' .$categorie['slug']."/</em></small><br/>";
        fputs($sitemapXML, "\t<url>\n");
        fputs($sitemapXML, "\t\t<loc>".$URLSource.'/category/'
        .$categorie['slug']."/</loc>\n");
        fputs($sitemapXML, "\t\t<priority>0.8</priority>\n");
        fputs($sitemapXML, "\t</url>\n");
        $nb++;
    }
}

// Boucle de récupération des données
foreach($connexion->query($requeteSQL) as $rangee) {

    $extensions = strtolower(pathinfo($rangee['guid'],PATHINFO_EXTENSION));

    if(!in_array($extensions,$fichiersJoints)) {
        echo "<br/><strong>Groupe ".$tour."</strong><br/>";
        echo " <small><em>".$URLSource."/".$rangee['post_name']."/ </em></small><br/>";
        fputs($sitemapXML, "\t<url>\n");
        fputs($sitemapXML, "\t\t<loc>".$URLSource."/".
        $rangee['post_name']."/</loc>\n");
        fputs($sitemapXML, "\t\t<lastmod>".date('Y-m-d',
        strtotime($rangee['post_modified'])).'T'.date('h:m:s+00:00',
        strtotime($rangee['post_modified']))."</lastmod>\n");
        $tour++;
        $nb++; // On incrémente à chaque tour de boucle
    }

    // On associe les URL avec leurs fichiers associés (.jpg, .png...)
    $requeteSQL2 = "SELECT $colonnes FROM $tables WHERE post_parent='".$rangee['ID']."' AND
    (post_type='$condition1' OR post_type='$condition2')";
```

```
    foreach($connexion->query($requeteSQL2) as $url) {
    echo "<small><em>".$url['guid']."</em></small> (fichier lié)<br/>";
    fputs($sitemapXML,"\t\t<image:image>\n");
    fputs($sitemapXML,"\t\t\t<image:loc>".$url['guid']."</image:loc>\n");
    fputs($sitemapXML,"\t\t\t<image:caption>".$url['post_title']."</image:caption>\n");
    fputs($sitemapXML,"\t\t\t<image:title>".$url['post_name']."</image:title>\n");
    fputs($sitemapXML,"\t\t</image:image>\n");
    $nb++; // On incrémente à chaque tour de boucle
    }

    if(!in_array($extensions,$fichiersJoints)) {
        fputs($sitemapXML, "\t</url>\n");
    }
}
echo "<h2>".$nb." pages ajoutées dans ".$fichierSitemap."</h2>";

// On finalise le fichier XML en fermant la balise </urlset>
fputs($sitemapXML, '</urlset>');
?>
```

La présentation de ces deux scripts s'est avérée assez technique. Les plus débutants doivent avant tout les tester pour comprendre le fonctionnement général car cela fait énormément d'informations à ingurgiter. Ne soyez pas inquiets, il n'est pas toujours nécessaire de se mettre à l'ouvrage pour faire un bon référencement…

En définitive, il serait intéressant de développer des scripts spécifiques aux formats vidéo, par exemple, pour générer des fichiers Sitemap précis ou encore pour les sites mobiles. Actuellement, ce n'est pas encore suffisamment développé mais sur le même principe que les deux codes précédents, nous pourrions générer tous les sites de Sitemap que nous souhaitons.

Créer un fichier robots.txt

Principe général de fonctionnement

La désindexation des pages et des fichiers résulte généralement de plusieurs méthodes conjointes mais la plus efficace consiste à créer un fichier `robots.txt`. Véritable fichier texte déposé à la racine du serveur, il a pour vocation d'indiquer aux robots quelles pages doivent être suivies et surtout lesquelles doivent être indexées ou non. Son rôle peut être double puisqu'il permet de déréférencer des pages ou des documents que nous jugeons peu intéressants, et offre aussi la possibilité de nettoyer des URL présentes en doublon pour contrer d'éventuels contenus dupliqués.

Généralement, un crawler lit d'abord le fichier `.htaccess`, puis il s'intéresse au fichier `robots.txt` afin d'avoir une liste de paramètres à respecter avant de procéder à l'indexation et à l'enregistrement des données. Si le fichier est absent, il continue sa lecture et indexe tout ce qui lui semble pertinent lors de ses parcours successifs.

Le fichier `robots.txt` impose des contraintes pour être pleinement fonctionnel ; veillez à les respecter. Tout d'abord, le fichier doit impérativement se nommer `robots.txt` (au pluriel et avec l'extension d'un fichier texte). Par ailleurs, pensez à placer le fichier uniquement à la racine du serveur.

La moindre faute dans le nom du fichier ou un mauvais placement du document fera qu'il sera ignoré par les robots. Enfin, assurez-vous également que le fichier `robots.txt` ne contient pas de lignes vides, car cela peut entraîner un dysfonctionnement dans certains cas. L'idéal est de placer une instruction par ligne, de vérifier si des sauts de ligne ont été effectués par mégarde et de bien les supprimer avant l'envoi du fichier sur le serveur.

Parmi les moteurs de recherche, Google possède une telle qualité d'indexation qu'il arrive à notifier de nombreuses pages web, même celles que nous ne souhaitons pas voir apparaître. Le déréférencement agit ainsi en amont pour pallier ces éventuels problèmes. Néanmoins, sachez que la présence d'un fichier `robots.txt` n'implique pas toujours une désindexation idéale des URL indiquées. Les moteurs restent les seuls maîtres et il arrive fréquemment que quelques pages passent au travers des mailles du filet que nous avons conçu grâce à notre fichier.

Voici plusieurs cas dans lesquels les moteurs de recherche peuvent outrepasser le fichier `robots.txt` :

- le moteur a indexé la page web avant que le fichier `robots.txt` ne soit mis en place ;
- le fichier est incorrect ou présente des erreurs d'écriture ;
- le robot ignore des commandes de son propre chef ;
- les pages web à déréférencer sont jugées pertinentes par le robot pour différentes raisons ou parce que des liens de qualité pointent vers ces pages.

Généralement, quand une page ou un document ne doit pas être indexé, il faut passer par l'interface Google Search Console ou Bing Webmaster Center pour supprimer ces adresses superflues grâce à l'option URL à supprimer. Il s'agit d'une méthode connexe pour obtenir des résultats satisfaisants sur le long terme.

Encodage du fichier robots.txt

Le fichier `robots.txt` doit être encodé en UTF-8 pour être fonctionnel, avec ou sans BOM *(Byte Order Mark)*, c'est-à-dire avec ou sans marqueur qui indique l'ordre de lecture des octets en début de fichier (caractère invisible par défaut).

Étapes de création d'un robots.txt

Nous allons créer entièrement un fichier `robots.txt` pour comprendre les différentes étapes et manipulations à effectuer. Nous verrons par la suite des spécificités souvent méconnues et des outils pour nous faciliter la tâche, mais dans un premier temps, nous devons mettre la main à la pâte pour savoir comment fonctionne ce fichier si important pour le référencement.

Tout d'abord, sachez que peu d'instructions sont disponibles dans les fichiers `robots.txt`. Les plus courantes qui nous intéressent sont les suivantes :

- `user-agent:` pour indiquer le ou les robots qui devront prendre en compte les règles à suivre ;
- `allow:` pour autoriser l'indexation des pages, sachant que cette option est celle par défaut dans les moteurs de recherche puisqu'ils crawlent et indexent si le fichier est inexistant ;
- `disallow:` pour limiter l'enregistrement et le suivi de certains documents ou pages. C'est cette fonctionnalité qui nous intéresse pour le déréférencement.

Sachez que la casse est sans importance pour les directives, vous pouvez donc librement écrire avec ou sans majuscules. Toutefois, respectez la casse dans les adresses web à bannir ou à autoriser car elle est prise en compte dans ce cas. Ainsi, il n'y a pas de différence entre `Disallow:` et `disallow:`. En revanche, `Fichier.html` et `fichier.html` sont deux documents dissociés aux yeux des robots, nous devons donc respecter leur intitulé exact.

Après chacune de ces instructions, il suffit d'inscrire les données qui nous intéressent. Commençons par le cas de la directive `user agent:` qui est la plus simple à comprendre. Nous devons préciser le ou les robots qui prendront en compte les règles de bonne conduite du fichier `robots.txt`. Il est important de bien connaître leur nom pour apporter de la précision. Voici quelques exemples :

- Googlebot pour les résultats classiques de Google ;
- Feedfetcher-Google pour les flux de syndication de Google ;
- Googlebot-News et Googlebot-Image respectivement pour les actualités et images ;
- Yandexbot pour le moteur russe Yandex ;
- Gigabot pour Gigablast ;
- Bingbot pour Bing de Microsoft ;
- Teoma pour Ask Jeeves ;
- Yahoo! Slurp pour Yahoo! ;
- Baiduspider pour le leader de la recherche chinoise Baidu ;
- Exabot pour Exalead ;
- Lexxebot pour le moteur de recherche Lexxe…

Il existe aussi des robots spécifiques à certains programmes mis en place par des applications ou scripts, donc la liste peut être infinie. Nous avons cité ici les robots principaux que nous rencontrerons dans notre démarche de référencement. Ainsi, pour bloquer l'accès de certaines adresses web à Google ou à Bing, il faudra écrire ceci :

```
user-agent: Googlebot
{ bloc d'instructions }
user-agent: Bingbot
{ bloc d'instructions }
```

Cet exemple montre que les robots parcourent le fichier `robots.txt` de haut en bas, comme pour les fichiers HTML classiques. Il suffit dans ce cas de placer les blocs d'instructions reliés à un robot précis les uns après les autres. La seule recommandation de Google est de porter attention à l'ordre des blocs et de ne pas générer de confusion entre les instructions, bien que ce type de procédé soit rarissime.

L'autre technique pour se faciliter la tâche est d'appliquer certaines actions à tous les robots d'un seul tenant. Pour ce faire, il suffit d'utiliser le caractère * qui permet d'englober la totalité des moteurs existants. Ainsi, la ligne suivante indique que tous les robots ne devront pas suivre ni indexer les pages indiquées dans les instructions qui suivront :

```
user-agent: *
```

En réalité, nous utilisons généralement un seul bloc d'instructions pour les pages car il serait étonnant d'accepter l'enregistrement de documents sur Bing et non sur Google par exemple. C'est pour cette raison que le caractère * est régulièrement utilisé par les référenceurs en matière de déréférencement web.

Le seul cas qui nous incite à utiliser divers groupes successifs d'instructions est Google car il s'agit d'un des rares moteurs à avoir des robots distincts pour certaines plates-formes. Ainsi, vous pouvez refuser l'indexation d'un dossier d'images à Googlebot-Image sans pour autant que Googlebot soit interdit d'accès. Il en va de même pour les actualités avec Googlebot-News ou les flux de syndication avec Feedfetcher. Si vous êtes dans ces cas précis, il est recommandé de procéder à un bloc d'instructions génériques, puis à des blocs spécifiques pour les robots complémentaires.

Une fois les robots ciblés, nous devons paramétrer les adresses web à interdire ou à autoriser. Dans notre cas, nous mettrons de côté les URL à indexer puisque ce cas est celui par défaut dans les moteurs (le fonctionnement est identique à ce qui va suivre donc si `allow:` vous intéresse, vous ne serez pas perdu(e)). Il est essentiel de retenir que chacune des URL inscrites dans le fichier `robots.txt` ne doit pas contenir le protocole d'origine (`http`, `https`...) et doit être précédée du caractère `/` (qui indique la racine du serveur).

Les robots d'indexation vont automatiquement déduire le protocole utilisé et appliquer le schéma d'adresse web suivant : `protocole://www.nom-de-domaine.ext/url-ou-dossier-pris-en-compte`. Le caractère `/` correspond au nom de domaine dans l'URL. Nous pouvons ensuite bloquer l'accès à des dossiers complets, à des pages web ou à des documents spécifiques si les moteurs les prennent en compte, tels que des fichiers PDF ou DOC, par exemple.

Spécificité des protocoles du Web

Si un fichier `robots.txt` interdit l'accès à un fichier nommé `exemple.php` sur un serveur FTP comme `ftp://www.monsite.com`, il ne bloquera pas l'indexation d'un fichier du même nom si ce dernier est situé sur un serveur HTTP ou HTTPS.

Il existe peu de méthodes pour bloquer l'accès des fichiers, nous allons toutes les présenter. Nous pouvons nommer un fichier par son vrai nom comme dans l'exemple suivant :

```
disallow: /un-fichier.html
```

Il est également possible de bloquer un répertoire complet, mais n'oubliez pas d'ajouter le caractère `/` à la fin de l'URL pour préciser que vous souhaitez déréférencer son contenu.

```
disallow: /repertoire-bloque/
```

Attention aux slashs

L'oubli du caractère / à la en fin de répertoire pose problème car cela indique aux robots que les fichiers commençant par la chaîne de caractères indiquée doivent être bloqués. Dans l'exemple, cela signifierait que des fichiers ou des pages comme `repertoire-bloque.html` ou `repertoire-bloque/un-fichier.php` seraient bloqués. Attention également à ne jamais écrire l'instruction `disallow: /` car elle bloque l'ensemble de l'indexation !

Vous pouvez bien entendu mélanger les deux instructions pour limiter l'accès à des fichiers situés dans des répertoires donnés comme dans l'exemple suivant :

```
disallow: /repertoire/fichier-bloque.html
```

Nous pouvons limiter l'indexation avec ce type de procédé, mais d'autres méthodes plus pointues sont également disponibles. En effet, certains caractères spéciaux, que nous retrouvons souvent dans les expressions régulières en développement web ou sur Google Analytics, précisent les adresses web à bloquer ou à autoriser. Concernant les fichiers `robots.txt`, peu de caractères sont accessibles mais ils offrent des avantages certains.

- Le caractère * signifie « tout » en langage informatique. Il permet d'indiquer une suite de caractères autorisés dans une instruction. Par exemple, la chaîne `fichie*` signifie que toutes les pages ou fichiers qui commencent par `fichie` et qui sont suivis ou non par d'autres caractères doivent être indexés ou déréférencés.
- Le caractère $ marque la fin d'une chaîne de caractères. Il convient de ne l'utiliser qu'après le nommage d'une extension si vous ne voulez pas que l'instruction soit obsolète. Ainsi, l'instruction `disallow: /fchie*.php$` signifie que tous les fichiers contenant la chaîne `fichie` et se terminant uniquement par `php` sont bannis.
- Le caractère # permet d'ajouter des commentaires dans les fichiers `robots.txt`, comme dans d'autres langages web ou en Shell. Tout ce qui suit ce caractère dans une ligne est interprété comme un commentaire textuel destiné à apporter des précisions jugées utiles.

Nous pouvons aller encore plus loin pour construire des blocages significatifs et qui limitent l'indexation des URL portant des paramètres gênants comme c'est souvent le cas avec des langages web tels que PHP ou C#. Voici quelques exemples d'instructions que nous pouvons insérer dans un fichier `robots.txt` :

```
user-agent: *
# Interdire l'accès aux paramètres dans une URL en HTML
disallow: /*.html$
# Interdire l'accès aux sessions PHP (avec la chaîne SESSID)
disallow: /*SESSID*
```

Comment masquer les sessions dans les URL ?

Écrivez l'instruction `SetEnv SESSION_USE_TRANS_SID 0` dans un fichier `.htaccess` pour faire disparaître les variables de session dans les URL, c'est propre et cela évite les problèmes d'indexation et de contenus dupliqués.

```
# Bloquer tous les paramètres d'URL (méthode généraliste)
```

```
disallow: /*?*
# Bloquer l'accès aux images pour tous les moteurs
disallow: /images/

user-agent: Googlebot-Image
# Bloquer les images GIF pour Gooblebot-Image
disallow: /*.gif$
```

Enfin, il est à noter que les URL peuvent être inscrites dans le fichier `robots.txt` sous leur forme originelle avec l'adresse IP sur le serveur. Les URL portant un numéro de port dans un but de sécurisation sont également tolérées. Dans ces deux cas précis, l'indexation sera limitée aux fichiers accessibles uniquement avec l'adresse IP mentionnée ou le numéro de port indiqué.

Outils et spécificités des fichiers robots.txt

Google fournit de nombreuses informations sur les fichiers `robots.txt` dans sa documentation officielle (source : http://goo.gl/FL5Bs), ce qui s'avère intéressant car quelques subtilités sont mentionnées. Intéressons-nous à quatre d'entre elles qui peuvent être importantes pour le référencement :

- la propriété `crawl-delay:` ;
- l'instruction `sitemap:` ;
- la directive `noindex:` ;
- le cas de Google AdSense et du robot Mediapartners-Google.

Limiter la surcharge serveur avec crawl-delay

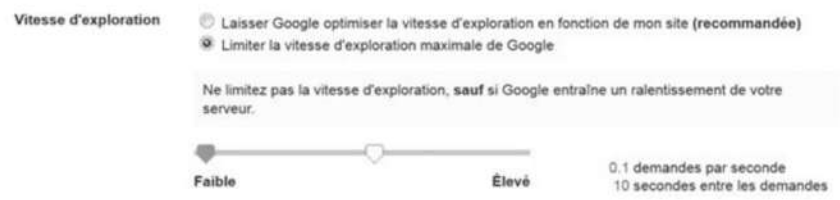
L'instruction `crawl-delay` est prise en compte sur Bing, MSN et Ask mais pas sur Google. Cette fonction permet d'indiquer aux robots un délai à respecter entre deux requêtes afin de décharger le serveur. En effet, il est possible qu'une indexation massive et simultanée par divers robots engendre des pertes de vitesse de chargement pour les internautes à cause d'une surcharge serveur. Dans ce cas, indiquer un délai d'indexation plus lent peut s'avérer très utile, comme dans l'exemple suivant :

```
user-agent: Bingbot
# Indique un délai de crawl de 5 secondes entre chaque requête
crawl-delay: 5
```

Google ne prend pas en compte l'instruction `crawl-delay`. Pour maîtriser la surcharge, connectez-vous à votre compte Google Search Console, cliquez sur la flèche située à droite de l'icône représentant une roue crantée (menu *Configuration*) et sélectionnez *Paramètres du site*. Dans la rubrique *Vitesse d'exploration*, paramétrez la vitesse d'exploration maximale des pages ou laissez Google gérer l'option.

Figure 1-8

Paramétrage de la vitesse d'exploration dans la Google Search Console



Mais où est passée l'option de réglages ?

Il peut arriver qu'aucune option ne soit disponible dans la rubrique *Vitesse d'exploration*. Si votre serveur est assez puissant et que Google a été configuré automatiquement, vous n'aurez pas à paramétrer le délai d'indexation.

Ajouter des fichiers sitemap.xml

L'instruction `sitemap:` peut s'avérer très utile pour indiquer aux moteurs de recherche le chemin d'accès direct vers un fichier `sitemap.xml`. Certains moteurs ont opté pour ce protocole, comme Google, Bing, Yahoo!, Ask et Exalead. Avec cette instruction, il n'est plus nécessaire de passer par les interfaces pour webmasters pour préciser l'adresse du fichier `sitemap.xml`.

Il s'agit même de la seule technique viable pour indiquer à des moteurs secondaires, tels qu'Exalead, le chemin d'accès à ce fichier. En effet, certains moteurs ne disposent pas d'interface pour les webmasters donc le fichier `robots.txt` reste la seule alternative. Voici comment procéder :

```
# Indique l'adresse du fichier sitemap.xml
sitemap: http://www.nom-de-domaine.ext/sitemap.xml
```

Où placer le fichier Sitemap ?

Il n'est pas obligatoire de placer le fichier `sitemap.xml` à la racine du site, vous pouvez aussi l'enregistrer dans un répertoire. Dans ce cas, il suffit alors d'indiquer l'adresse précise.

Si vous possédez plusieurs fichiers Sitemap, vous pouvez ajouter autant d'instructions que nécessaire pour atteindre le résultat escompté. Les moteurs gèrent bien ce procédé dans un fichier `robots.txt`.

Ne pas interdire l'accès aux fichiers Sitemap

Il arrive que les fichiers Sitemap soient indexés par les robots et affichés dans les SERP des moteurs. Il ne faut pas appliquer la syntaxe `disallow: /sitemap.xml` pour autant car cela bloquerait la lecture du fichier et contredirait l'instruction `sitemap:`.

Directive noindex:

En novembre 2007, le webmaster du site Sebastian's Pamphlets a annoncé une découverte intéressante pour une bonne gestion de l'indexation (source : http://goo.gl/PY29U). En effet, il a mis en exergue la prise en compte d'une directive `noindex:`, semblable à celle que nous pouvons retrouver dans les balises meta

robots. Google n'a jamais fait d'annonce officielle au sujet de cette directive mais les différentes études menées ont prouvé qu'elle fonctionne parfaitement.

L'avantage de l'instruction `noindex:` est d'autoriser la lecture des pages et le suivi des liens internes, et de bloquer l'indexation des adresses web spécifiées dans ces liens. Par exemple, nous pouvons désirer le suivi des liens dans une page Plan de site sans pour autant souhaiter que cette page apparaisse dans les SERP.

La structure de cette directive est identique à celle de l'instruction `disallow:`. L'exemple suivant présente un blocage de l'indexation du fichier `non-indexe.html` sans pour autant empêcher le robot de lire le fichier :

```
user-agent: *
noindex: /non-indexe.html
```

Intérêt de la directive noindex:

La directive `noindex:` s'avère intéressante si vous voulez cacher certains fichiers présents dans les SERP. En effet, nous ne souhaitons pas toujours voir certains documents tels que le fichier `sitemap.xml` ou le fichier `robots.txt` dans les résultats. Avec une bonne utilisation de l'instruction, nous pouvons bloquer l'indexation de ces fichiers sans limiter pour autant leur accès aux robots. Toutefois, John Mueller, porte-parole de Google, déconseille l'usage de cette directive non officielle (source : http://goo.gl/BK1NCh).

Rien n'assure à ce jour que la directive `noindex:` fonctionne sur d'autres moteurs que Google car les tests effectués sur Bing n'ont pas été probants. Il semblerait même que le moteur de Microsoft ignore totalement le fichier `robots.txt` dans certains cas. Certes, nous ciblons surtout Google car sa capacité d'indexation est bien plus évoluée que celle de sa concurrence. Si vous souhaitez appliquer cette directive, sachez que son impact est limité.

Cas des ressources CSS et JavaScript

Google recommande de ne pas bloquer l'accès aux ressources CSS et JavaScript, quelle qu'en soit la finalité pour le site. Le moteur a fait de grands progrès en 2015 dans le crawl des fichiers CSS et JavaScript, notamment pour améliorer l'indexation et la lecture des pages web en Ajax. Par conséquent, il préfère avoir accès à un maximum d'informations pour mieux comprendre les pages, mieux les afficher et mieux respecter leur structure. Rien ne dit que cela affecte le positionnement, mais les pages web composées avec de l'Ajax sont aidées par ces nouveaux procédés de crawl.

Dans le cas d'un blocage total des ressources dans un fichier `robots.txt`, cela joue un mauvais tour aux robots de Google puisqu'ils n'ont plus accès aux contenus comme il le faudrait. Ce même procédé peut s'appliquer aussi pour les fichiers `.htaccess` ou `web.config`, qui ne doivent pas non plus bloquer l'accès aux feuilles de styles et aux scripts externes.

L'exemple de blocage le plus courant est celui de Google Adsense, dont le robot est Mediapartners-Google. La méthodologie est très simple à mettre en œuvre (source : http://goo.gl/SelEM), elle consiste à supprimer au début du fichier `robots.txt` les deux lignes suivantes :

```
user-agent: Mediapartners-Google
disallow: /
```

Cette spécificité de Google AdSense n'est pas négligeable tant la régie est utilisée et constitue un exemple type de ressources bloquées. Google souhaite vraiment accéder au site comme si les robots constituaient un utilisateur lambda, il faut donc bloquer uniquement les fichiers ou répertoires inutiles, mais pas les ressources qui font fonctionner un site web.

Outils de création de fichiers robots.txt

Nous savons désormais comment construire manuellement des fichiers `robots.txt` optimisés et performants pour réussir notre déréférencement et tolérer la lecture de certains fichiers. Les petits sites web ne demandent pas de gros efforts en termes de désindexation, mais dès que les pages commencent à se multiplier, la donne change et se complexifie. Pour répondre aux problèmes de création d'un fichier `robots.txt`, nous disposons de nombreux outils sur la Toile, dont voici une liste non exhaustive :

- Robot Control Code Generation Tool : http://goo.gl/wa5x9t ;
- SeoBook : http://goo.gl/IIXF9D ;
- Générateur Internet MarketingNinjas : http://goo.gl/DyRAA ;
- Webophil : http://goo.gl/YLo7FI ;
- YellowPipe : http://goo.gl/ER6Ach ;
- générateur du site Aspirine : http://goo.gl/hNuZCI ;
- générateur du site HowRank : http://goo.gl/oc0ZHN.

Les différents CMS du marché disposent également d'extensions de ce type comme iRobots.txt SEO pour WordPress, RobotsTxt pour Drupal ou encore JCrawler pour Joomla notamment. D'autres CMS comme Prestashop mettent nativement un générateur de fichiers `robots.txt` à disposition des webmasters pour leur faciliter la tâche.

Limitation des générateurs

Notez que ces générateurs constituent une aide mais ne sont pas toujours parfaits. Qui plus est, ils n'offrent pas toutes les fonctionnalités que nous venons de découvrir. Par exemple, vous devrez ajouter manuellement les directives `noindex:` qui vous intéressent.

Autres techniques d'optimisation

Maîtriser les rich snippets

Présentation des extraits enrichis

Les extraits de code enrichis *(rich snippets)* permettent d'ajouter une surcouche sémantique dans le code des pages afin d'apporter un certain nombre de précisions utiles pour les robots. Ces données complémentaires ont pour rôle de mieux qualifier les contenus et faire comprendre la logique et la structure du code aux moteurs de recherche.

Prenons un exemple : si nous créons une fiche produit en xHTML, nous mettons en place plusieurs « champs » comme le bloc pour les photos du produit, le prix, la disponibilité, la description ou encore le titre. Tous seront souvent qualifiés sémantiquement par les mêmes balises en HTML, à savoir les `<div>`.

Par conséquent, comment Google peut-il différencier les blocs et mieux comprendre les contenus ainsi que notre structure de page ? Certes, les contenus sont une indication mais cela ne suffit pas. C'est en ce sens qu'interviennent les extraits de code enrichis, qui aident à la compréhension du code par les robots.

L'autre avantage des rich snippets nous concerne directement puisque certains balisages spécifiques améliorent l'affichage des résultats dans les SERP, ce qui octroie une visibilité accrue pour ces pages au détriment des concurrents. Certes, le positionnement web n'est pas augmenté par ce biais, mais si l'affichage est meilleur et que le résultat occupe plus de place à l'écran, il est fort probable que le taux de clics soit amélioré.

Tout l'intérêt des extraits de code enrichis se retrouve ici pour les référenceurs, bien que nous puissions éventuellement envisager un impact sur le positionnement dans les années à venir si Google trouve cette démarche valable.

De nombreuses analyses ont montré que les extraits de code enrichis amélioraient le taux de clics, réduisaient le taux de rebond dans les pages web visitées voire favorisaient le taux de conversion dans les boutiques en ligne.

En effet, les rich snippets sont nombreux et ont tous leur rôle à jouer, mais tous ne permettent pas d'afficher des informations supplémentaires dans les résultats de recherche. Tous les sites ne sont donc pas directement impactés par ces données sémantiques.

Souvent, ce sont les sites e-commerce qui en profitent ou les sites utilisant des systèmes de notation (blogs, sites de cuisine...) mais aussi les sites musicaux. Voici les informations les plus souvent affichées à l'aide des extraits de code enrichis :

- étoiles de notation et/ou nombre de votes ;
- nom, tarif et/ou disponibilité d'un produit ;
- titre, durée et album de musique ;
- numéro de téléphone et/ou adresse ;

- logo d'entreprise lorsque Google estime que cela est pertinent (possible à l'aide de l'attribut `rel="publisher"` placé dans une balise `<link />`).

Figure 1-9

Exemples de rich snippets affichés dans les SERP

La documentation de Google explique comment mettre en place des rich snippets (source : http://goo.gl/rpcPZ7).

1. Choisir un format de balisage parmi ceux autorisés : microdonnées avec Schema.org 1. ou JSON-LD (recommandé par Google), microformats (très complets et complexes à l'aide d'un système de classes spécifiques en HTML) ou RDFa (relativement simple à mettre en place autour d'attributs `property` notamment).
2. Effectuer le balisage des contenus à partir du format choisi.
3. Vérifier et tester l'exactitude du balisage à l'aide de l'outil en ligne mis à disposition par Google à l'adresse suivante : http://www.google.com/webmasters/tools/richsnippets

Faut-il absolument intégrer des extraits de code enrichis ?

Les rich snippets sont utiles mais pas nécessairement pour tous les types de sites. Il faut savoir que Google peut pénaliser des sites qui abusent des extraits de code enrichis ou qui les utilisent à outrance alors que les pages ne le nécessitent pas forcément. Aussi, soyons vigilants et utilisons-les avec intelligence.

Nous allons étudier comment mettre en place des extraits de code enrichis selon les trois formats disponibles en nous basant sur trois exemples simples de données qui peuvent être affichées dans les SERP : une fiche produit, des coordonnées d'entreprise et un système de notation.

Figure 1-10

Exemple de test avec l'outil de Google

Microdonnées et Schema.org

Schema.org (https://schema.org) et les microdonnées représentent le format le plus conseillé à ce jour. S'il n'est pas meilleur que ces concurrents directs, il offre l'avantage d'avoir des attributs et des propriétés tolérées par le W3C et la spécification HTML 5. Nous verrons que les autres formats, notamment les microformats, ne présentent pas forcément les mêmes avantages.

Globalement, les microdonnées se basent sur trois attributs HTML.

- `itemscope` indique qu'un balisage Schema.org va être mis en place dans un bloc. Il faut ajouter cet attribut dans tous les blocs qui contiennent une sémantisation du code (`<div>`, `<header>`, `<article>`...).
- `itemtype` précise le type de sémantisation mis en place. Cela diffère en fonction des informations que nous souhaitons mettre en avant. Par exemple, le type pour présenter la fiche d'un film est `itemtype "http://schema.org/Movie"`, celui pour présenter une personne est `itemtype "http://schema.org/Person"` et celui pour afficher des informations concernant un produit est `itemtype="http://schema.org/Offer"`. La liste des attributs `itemtype` est disponible sur le site de Schema.org. Attention toutefois car il existe des sous-types intermédiaires, notamment pour les fiches produits, etc.
- `itemprop` correspond à la liste des propriétés à indiquer pour chaque information que nous souhaitons préciser. Il existe des listes de propriétés selon les types sélectionnés mais toutes ne sont pas utiles si vous souhaitez uniquement voir des informations affichées dans les SERP de Google.

Nous pouvons désormais envisager la mise en place de nos trois exemples simples pour visualiser comment utiliser les extraits de code enrichis. Pour ce faire, il suffit de suivre les indications et les tableaux de données fournis par le site Schema.org.

Pour les fiches produits, voilà ce qu'il est possible de réaliser :

```
<div itemscope itemtype="http://schema.org/Product">
    <section itemprop="offers" itemscope itemtype="http://schema.org/Offer">
        <img itemprop="image" src="image-produit.jpg" />
        <h2 itemprop="name">Nom du produit</h2>
        <article itemprop="description">Description du produit</article>
        <div>
            <p itemprop="price">1000€</p>
            <p itemprop="sku">En stock</p>
        </div>
    </section>
</div>
```

Nous voyons ici que les attributs `itemtype` peuvent se confronter. En réalité, chaque type d'information contient des propriétés propres et des sous-propriétés accessibles via d'autres sous-types, c'est notamment le cas entre le type `Product` et le sous-type `Offer` qui permettent d'afficher le prix des produits et leur disponibilité.

Prenons maintenant l'exemple de la sémantisation de coordonnées pour des sociétés, associations ou entreprises en tout genre.

```
<div itemscope itemtype="http://schema.org/Person">
    <h2 itemprop="name">Mathieu Chartier</h2>
    <img src="photo-profil.jpg" itemprop="image" />
    <p itemprop="jobTitle">Formateur et référenceur</p>
    <div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    <p itemprop="streetAddress">Avenue de la Fraternité</p>
    <p>
        <span itemprop="postalCode">86000</span> 
        <span itemprop="addressLocality">Poitiers</span>
    </p>
    <p itemprop="addressRegion">Poitou-Charentes</p>
    </div>
    <p itemprop="telephone">06 xx xx xx xx</p>
    <p>E-mail : <a href="mailto:contact@internet-formation.fr"
    itemprop="email">contact@internet-formation.fr</a></p>
    <p>Site web : <a href="http://www.internet-formation.fr"
    itemprop="url">www.internet-formation.fr</a></p>
</div>
```

Nous voyons que beaucoup d'informations peuvent être précisées à l'aide des propriétés `itemprop`. Il convient également de mélanger le type `Person` et le sous-type `PostalAdress` si nous souhaitons afficher toutes les données relatives à un lieu et une personne.

Pour les personnes morales, le type `PostalAdress` peut suffire. Il faudra ajouter la propriété `name` pour indiquer le nom de la société ou de l'association, par exemple.

Enfin, nous allons étudier comment mettre en place la sémantisation d'un système de notation sur un blog, par exemple, même si ce principe peut s'appliquer à presque tous les sites web.

```
<article itemscope itemtype="http://data-vocabulary.org/Review">
    <hgroup>
    <h2 itemprop="itemreviewed">Titre de l'article</h2>
    <h3>Publié par <span itemprop="reviewer">Alexandra Martin</span> le 5 mai.</h3>
    <p>Texte de l'article</p>
    </hgroup>
<div>
<p><span itemprop="count">10</span> personnes ont voté.</p>
<p>
    <span itemprop="rating" itemscope itemtype="http://data-vocabulary.org/Rating">
    <span itemprop="average">4.2</span>/
    <span itemprop="best">5</span>
</p>
</div>
</article>
```

Intégrer des systèmes de notation enrichis

Il existe d'autres méthodes pour afficher les notes, n'hésitez pas à vous appuyer sur les exemples fournis par Google dans ses documentations (source : http://goo.gl/78mnsM) ou sur le site officiel des microdonnées (source : http://schema.org).

Microformats

Les microformats (http://microformats.org) représentent une autre forme de sémantisation basée sur des listes de groupes de données (`hCalendar`, `hCard`, `hReview`…) contenant une multitude de classes placées dans le code HTML des pages web. Le seul inconvénient de ce système est d'ailleurs l'usage de l'attribut `class` car il est généralement consacré aux classes CSS, ce qui peut porter à confusion voire ajouter des doublons dans les codes sources.

Ici, il faut définir le type de donnée sémantique, puis utiliser les classes associées pour apporter des précisions destinées aux robots. Nous pouvons reprendre les trois exemples précédents à l'identique en les adaptant aux microformats.

Commençons par le cas particulier des fiches produits :

```
<section class="hproduct">
    <img class="photo" src="image-produit.jpg" />
    <h2 class="fn">Nom du produit</h2>
    <article class="description">Description du produit</article>
    <div>
        <p class="price">1000€</p>
        <p class="availability">En stock</p>
    </div>
</section>
```

Notons que la section `hProduct` des microformats n'a jamais été confirmée et reste à l'état de brouillon à l'heure actuelle. Il est donc conseillé d'utiliser plutôt Schema.org dans ce cas.

Pour les personnes physiques ou morales, nous pouvons reprendre le modèle suivant :

```
<div class="vcard">
    <h2 class="fn">Mathieu Chartier</h2>
    <img src="photo-profil.jpg" class="photo" />
    <p>Formateur et référenceur</p>
    <div>
        <p class="street-address">Avenue de la Fraternité</p>
        <p>
        <span class="postal-code">86000</span> 
        <span class="locality">Poitiers</span>
        </p>
        <p class="region">Poitou-Charentes</p>
    </div>
```

```
    <p class="tel">06 xx xx xx xx</p>
    <p>E-mail : <a href="mailto:contact@internet-formation.fr" class="email">
    contact@internet-formation.fr</a></p>
    <p>Site web : <a href="http://www.internet-formation.fr" class="url">
    www.internet-formation.fr</a></p>
</div>
```

Pour les entreprises et associations, il ne faut pas utiliser la classe `fn` seule mais lui ajouter la classe `org`. Ainsi, pour le nom d'une société, il faudrait plutôt écrire :

```
<h2 class="fn org">Nom de l'entreprise</h2>
```

Pour le système de notation, les différences avec les microdonnées sont relativement faibles en définitive, si ce n'est que l'attribut `itemprop` est souvent remplacé par `class`.

```
<article class="hreview-aggregate">
    <hgroup class="item">
        <h2 class="fn">Titre de l'article</h2>
        <h3>Publié par <span class="reviewer">Alexandra Martin</span> le 5 mai.</h3>
    </hgroup>
    <p class="summary">Texte de l'article</p>
    <div>
        <p><span class="count">10</span> personnes ont voté.</p>
        <p>
            <span class="rating">
            <span class="average">4.2</span>/
            <span class="best">5</span>
        </p>
    </div>
</article>
```

Un plug-in WordPress adéquat…

Si vous souhaitez mettre en place un système de notation, le plug-in gratuit GD Star Rating pour WordPress permet d'obtenir ce type de fonction sans effort.

RDFa

Le dernier procédé pour ajouter de la sémantique dans les codes HTML est le RDFa (*Resource Description Framework attributes*, http://rdfa.info), un des plus anciens à avoir été créé. Le code est bien plus marqué ici car de nouveaux espaces de noms ont été créés en XML à l'origine pour développer cette technique.

Le système RDFa est beaucoup plus complexe que les autres car il peut faire appel à plusieurs types de balisages différents comme celui de Dublin Core (`dc` dans les codes) ou tout simplement le vocable RDFa (source : http://rdf.data-vocabulary.org/rdf.xml). Selon le langage préféré, certaines différences sont notables.

Globalement, voici les attributs utiles en HTML pour mettre en place le balisage RDFa :

- `xmlns` indique le type d'espace de noms utilisé, par exemple Dublin Core ;
- `typeof` indique le type d'information sémantiquement décrite ;
- `rel` précise des relations avec d'autres documents ou ressources ;
- `property` indique les propriétés sémantiques ;
- `content` accompagne parfois `property` pour fournir le contenu sous la forme valide tolérée par le système RDFa.

Nous allons tenter de reprendre nos trois exemples dans ce format sémantique en commençant tout d'abord par les fiches produits.

```
<section prefix="foaf:http://xmlns.com/foaf/0.1/ gr:http://purl.org/goodrelations/v1#"
typeof="gr:Offering">
    <img rel="foaf:depiction" src="image-produit.jpg" />
    <h2 property="gr:name">Nom du produit</h2>
    <article property="gr:description">Description du produit</article>
    <div property="gr:hasPriceSpecification" typeof="gr:UnitPriceSpecification">
        <p><span property="gr:hasCurrencyValue">1000</span>
        <span property="gr:hasCurrency">€</span></p>
        <p>En stock></p>
    </div>
</section>
```

Les sociétés et personnes peuvent aussi être décrites avec RDFa :

```
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Person">
    <h2 property="v:name">Mathieu Chartier</h2>
    <img src="photo-profil.jpg" class="photo" />
    <p property="v:title">Formateur et référenceur</p>
    <div rel="v:adress" typeof="v:Adress">
        <p property="v:street-address">Avenue de la Fraternité</p>
        <p>
            <span property="v:postal-code">86000</span> 
            <span property="v:locality">Poitiers</span>
        </p>
        <p property="v:region">Poitou-Charentes</p>
    </div>
    <p rel="v:tel">06 xx xx xx xx xx</p>
    <p>E-mail : <a href="mailto:contact@internet-formation.fr"
    property="v:email">contact@internet-formation.fr</a></p>
    <p>Site web : <a href="http://www.internet-formation.fr"
    property="v:url">www.internet-formation.fr</a></p>
</div>
```

Pour les personnes morales, il faut ajouter un sous-type avec `typeof="Organization"`, puis avec l'attribut `property="v:name"` pour indiquer la raison sociale.

La notation reprend un peu le même principe avec le type `v:review` :

```
<article xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Review">
    <hgroup>
        <h2 property="v:itemreviewed">Titre de l'article</h2>
        <h3>Publié par <span property="v:reviewer">Alexandra Martin</span>
                        le <span property="v:dtreviewed" content="2015-05-05">5 mai</span>.</h3>
    </hgroup>
    <p>Texte de l'article</p>
    <div>
        <p><span property="v:count">10</span> personnes ont voté.</p>
        <p rel="v:rating">
            <span typeof="v:Rating">
            <span property="v:average">4.2</span>/
            <span property="v:best">5</span>
        </p>
    </div>
</article>
```

JSON-LD

JSON-LD est un format de balisage sémantique récent, dérivé de la notation standard en JSON (source : http://json-ld.org). Le suffixe « LD » signifie « Linked Data » ; cette écriture est donc utilisée pour servir de marqueur de données sous la forme de groupes « propriété : valeur ».

Google utilise, voire favorise, le format JSON-LD pour mettre en œuvre les extraits structurés. Aussi, tous les types ainsi que toutes les propriétés et valeurs de `Schema.org` sont compatibles avec JSON-LD. Ce n'est donc qu'un choix d'écriture en définitive.

L'usage des microdonnées est souvent maîtrisé par les webmasters car cela fait plusieurs années que `Schema.org` est implanté, mais l'insertion des propriétés spécifiques au sein des balises HTML est parfois fastidieux et chronophage. Le format JSON-LD se présente alors comme une alternative pratique, car tout est concentré entre des balises `<script>`, souvent calées en bas des codes sources. Il faut juste respecter le type `"application/ld+json"` pour que cela soit fonctionnel pour les moteurs de recherche.

Dans la documentation de Google au sujet des données structurées (source : https://goo.gl/l647ld), de nombreux exemples sont fournis pour chaque type de donnée. Les propriétés et valeurs possibles sont indiquées, qu'elles soient obligatoires ou non, et il ne reste qu'à personnaliser le rendu. Voici un exemple complet en JSON-LD pour une fiche-produit sur un site e-commerce :

```
<script type="application/ld+json">
{
  "@context": "http://schema.org/",
  "@type": "Product",
  "name": "Nom du produit",
  "image": "http://www.boutique.com/image-produit.jpg",
  "description": "Description du produit",
  "brand": {
```

```
    "@type": "Thing",
    "name": "Nom de la marque du produit"
  },
  "offers": {
    "@type": "Offer",
    "priceCurrency": "EUR",
    "price": "99.99",
    "priceValidUntil": "2020-12-31",
    "availability": "InStock",
    "seller": {
      "@type": "Organization",
      "name": "Nom de la boutique"
    }
  }
}
</script>
```

L'avantage de la notation JSON-LD est de regrouper toutes les données structurées dans un même endroit, sans avoir à modifier les balises HTML. En outre, ce format reste assez lisible puisque toute la logique structurelle des marqueurs sémantiques est mise en avant par les tabulations et les types/sous-types de données (représentés par la propriété spécifique `"@type"`).

Outil d'aide au balisage des extraits de code enrichis

Google a récemment mis en place un outil d'aide à la réalisation du balisage sémantique des pages web. En effet, nous ne sommes pas tous des techniciens hors pair et il peut s'avérer utile de disposer d'outils pour faciliter le travail. Certes, les sites officiels de chaque format présentent parfois des outils de création, mais le Markup Helper de Google semble être encore bien plus fiable (source : http://goo.gl/9YlFuB). L'outil est accessible à l'adresse http://goo.gl/Vjjl9a ou via la Google Search Console dans la section dédiée aux données structurées. L'outil met à disposition plusieurs solutions pour réaliser son balisage sémantique :

- création des balises sémantiques à partir d'une URL existante ou d'un code HTML donné pour les sites web ;
- création du balisage pour des e-mails HTML.

Une liste de types d'informations est proposée pour affiner le balisage au fur et à mesure. Cela peut prendre un peu de temps quand les pages sont complexes, mais il suffit en réalité de répéter la démarche suivante :

- entrez l'URL de votre site web ;
- sélectionnez le type de données parmi les choix disponibles ;
- cliquez sur les zones à qualifier dans votre site et indiquez le type d'information décrite ;
- affichez le code HTML généré par Google Markup Helper ;
- copiez ou téléchargez le code HTML généré pour remplacer votre page existante.

Figure 1-11

Outil d'aide au balisage fourni par Google

Problème du marqueur en ligne

Un petit bémol concernant cet outil : le code généré contient parfois des erreurs HTML ou des invalidités avec la spécification W3C (par exemple, usage de balises de type `block` à l'intérieur de balises `inline`, ce qui n'est pas autorisé en théorie...). Il convient donc de vérifier l'exactitude du code avant de se précipiter à modifier et enregistrer définitivement la page web.

Figure 1-12

Mise en place du balisage avec Google Markup Helper

L'outil d'aide au balisage de Google est une bonne surprise qui ravira les moins techniciens d'entre nous mais restons toutefois vigilants en raison des erreurs engendrées. De plus, l'outil ne permet pas d'utiliser les microformats ou RDFa, il se contente uniquement des microdonnées avec Schema.org.

Retenons aussi que les SERP de Google affichent des informations complémentaires qui proviennent au moins aussi souvent des microdonnées que des autres formats. Il ne faut donc pas nécessairement privilégier Schema.org, les autres systèmes de balisage ont un réel rôle à jouer dans l'affichage de données dans les résultats du moteur de recherche.

Bien utiliser les URL canoniques

Les URL canoniques sont des adresses web préférentielles indiquées aux robots des moteurs de recherche lorsque plusieurs pages aux contenus similaires ou quasi identiques se retrouvent confrontées entre elles et risquent de poser des problèmes de DUST (voir chapitre suivant).

Prenons un exemple simple qui permet d'expliquer l'intérêt des URL canoniques : si une boutique en ligne contient des URL distinctes pour des fiches produits qui proposent des attributs divers, cela signifie que plusieurs adresses mènent vers la même fiche produit, à peu de choses près. Nous risquons alors de générer des contenus dupliqués par inadvertance voire de nous faire pénaliser. Voici à quoi pourraient ressembler des URL dupliquées :

```
http://www.example.com/produits?cat=robes&couleur=rouge&col=v
http://www.example.com/produits?cat=robes&couleur=vert&col=v
```

Ici, nous voyons un exemple de deux adresses web quelque peu différentes qui pointent en réalité vers la même page avec seulement quelques paramètres différents. Malheureusement, cela peut causer deux indexations et donc un ajout de contenus dupliqués dans les moteurs de recherche, ce qui peut vite être préjudiciable si des sanctions tombent…

L'attribut `rel "canonical"` a été prévu pour pallier ce type de problème. Il permet d'indiquer au robot la page mère (URL canonique) à indexer afin que les doublons soient ignorés par les moteurs.

La balise `<link />` est utilisée pour indiquer les URL canoniques dans toutes les pages doublonnées, elle doit absolument être placée dans la section `<head>…</head>` des pages web (ou envoyée dans les en-têtes HTTP). Le principe est simple, il suffit d'ajouter l'attribut `rel="canonical"` et l'attribut `href` contenant l'adresse de la page mère, comme dans l'exemple suivant :

```
<link href="http://www.example.com/produits?categorie=robes" rel="canorical" />
```

Il est important d'indiquer ces mêmes adresses dans un fichier Sitemap XML pour que les moteurs puissent déterminer les pages jugées comme importantes. Il convient donc de ne pas notifier les doublons dans ce listing de pages web.

Avant de se lancer dans la mise en place des URL canoniques, il est recommandé de bien choisir les pages canoniques et de vérifier leur bon fonctionnement. Par exemple, contrôlez la forme de l'URL indiquée (forme absolue ou relative) dans l'attribut `href`, l'exactitude de l'adresse (une page vers une page, une catégorie vers une catégorie, etc.) et que la balise n'est pas placée dans la balise `<body>` du code source HTML.

L'usage des URL canoniques impose quelques contraintes pour être pleinement fonctionnel :

- l'URL canonique notée dans la balise doit absolument exister et ne pas être une page de redirection ou redirigée ;
- la page mère (canonique) ne doit absolument pas contenir la balise meta `<meta content="noindex,nofollow" name="robots" />`, qui empêche l'indexation de la page ;
- les pages ne doivent contenir qu'une seule balise `<link rel="canonical" href="URL" />` par page. S'il en existe plusieurs, toutes les adresses précisées seront ignorées. Il convient donc de se méfier

des extensions dans les CMS qui génèrent des URL canoniques si vous en avez également mis en place par vous-même…

Comme nous l'avons vu, il existe deux méthodes pour indiquer des URL canoniques aux moteurs de recherche : soit avec la balise `<link />` en xHTML (`<link>` en HTML 5), soit par l'envoi d'en-têtes HTTP. Nous allons donc étudier cette seconde méthode rapidement car elle peut avoir son utilité dans certains cas.

Voici à quoi peut ressembler l'en-tête HTTP d'une page web une fois chargée :

```
HTTP/1.1 200 OK
Host: www.site.com
Date: Fri, 05 Sep 2014 15:31:05 GMT
Content-Type: text/html; charset=utf-8
Server: Apache
X-Powered-By: PHP/5.3.16
Vary: Accept-Encoding,User-Agent
```

Il suffit de trouver le moyen d'ajouter une ligne supplémentaire sous la forme suivante :

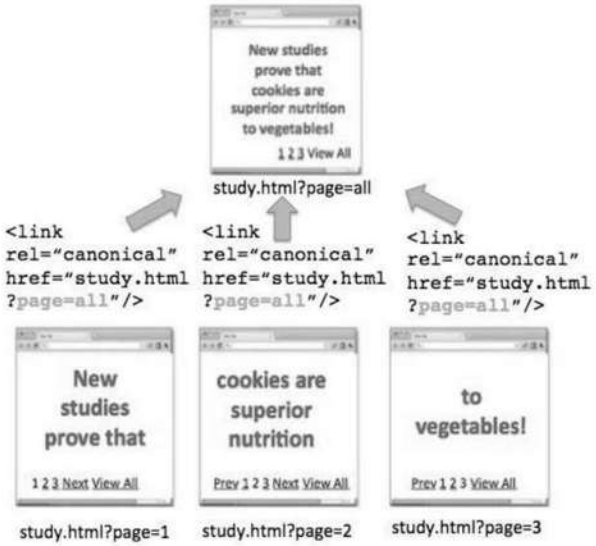
```
Link: <URL_DE_LA_PAGE_CANONIQUE>; rel="canonical"
```

Pour envoyer des informations dans les en-têtes HTTP, il faut utiliser la fonction `header();` en PHP. Dans notre cas, il convient d'indiquer l'URL canonique à envoyer, ce qui peut s'avérer utile et parfois plus simple à mettre en place lorsque nous créons notre propre CMS. Voici l'utilisation de base :

```
< ?php
$URL = "http://www.site.com/page.php";
header('Link:<'.$URL.'>; rel="canonical"');
?>
```

Figure 1-13

Exemple d'adresses web paginées dirigées vers une page contenant tous les résultats



Le cas le plus fréquent d'URL en doublon provient des paginations mises en place sur les sites comme dans le cas de galeries multimédias, de listing de liens (annuaires, par exemple) ou encore de liste de produits (sites e-commerce en général). Tous ces exemples font l'objet de contenus dupliqués sanctionnés par Google, il convient alors d'utiliser les URL canoniques avec la bonne méthode (source : http://goo.gl/YyKJAU) :

- ne pas pointer vers la première page paginée lorsqu'il existe plusieurs pages de résultats mais plutôt vers une page qui affiche tous les résultats ou qui est « neutre » ;
- utiliser les attributs `rel="prev"` et `rel="next"` dans des balises `<link />` pour indiquer les pages précédentes et suivantes en cas de pagination car elles permettent d'indiquer aux robots la relation qui les concernent.

Google a précisé que les valeurs `prev` et `next` de l'attribut `rel` peuvent suffire pour indiquer la présence d'une pagination et d'éventuels contenus dupliqués (source : http://goo.gl/EAJdBf), cette méthode est souvent la plus adéquate pour éviter tout risque…

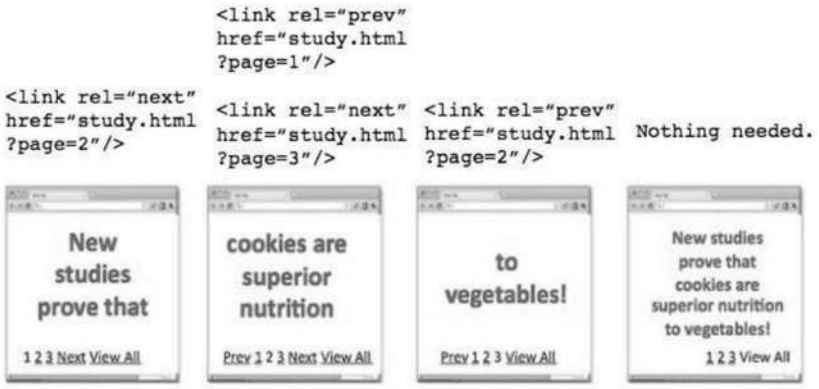
Prenons un exemple de mise en place avec `rel="prev"` et `rel="next"` pour les pages *n* (ici, 1 à 3) de l'adresse http://www.site.com/?page=n :

```
<!-- Page 1 -->
<link rel="next" href=" http://www.site.com/?page=2" />
<!-- Page 2 -->
<link rel="prev" href=" http://www.site.com/?page=1" />
<link rel="next" href=" http://www.site.com/?page=3" />
<!-- Page 3 -->
<link rel="prev" href=" http://www.site.com/?page=2" />
```

Il est également possible d'utiliser la variante `rel="previous"`, souvent méconnue mais valide.

Figure 1-14

Utilisation des valeurs next et prev sur Google



Bien que les systèmes de gestion de contenus tels que WordPress, Drupal et autres disposent souvent de moyens pour gérer les pages précédentes et suivantes, il est parfois nécessaire d'implanter soi-même les attributs `rel` manuellement. Si vous gérez vos paginations dynamiquement avec PHP, vous pouvez utiliser la fonction suivante codée pour l'occasion. Il suffit de la lancer dans la section `<head>…</head>` de la page PHP contenant la pagination en la paramétrant à votre guise.

```
function managePrevNext($param="page", $nbPages = 1) {
$parametre = htmlspecialchars($_GET[$param]);

// Récupération dynamique de l'URL (et des paramètres s'ils existent)
preg_match_all('#([^=])+([^?&\#])+#i', $_SERVER['QUERY_STRING'], $valueArgs);
$urlPage = $_SERVER['PHP_SELF'].'?';
foreach($valueArgs[0] as $arg) {
    $urlPage .= $arg;
    $urlPage = str_replace("&".$param."=".$parametre, "", $urlPage);
}
$urlPage .= "&".$param."=";
$urlPage = str_replace("?".$param."=".$parametre."&", "?", $urlPage);

// Vérifie et sécurise la fonction contre d'éventuelles failles
if(is_numeric($parametre) && $nbPages > 1 && $parametre < ($nbPages+1)) {
    // Balises rel="prev" et rel="next" dynamiques
    $prev = '<link rel="prev" href="'.$urlPage.($parametre-1).'" />'."\n";
    $next = '<link rel="next" href="'.$urlPage.($parametre+1).'" />'."\n";

    // S'il s'agit de la première page
    if($parametre == 1) {
        echo $next;
    }
    // S'il s'agit de la dernière page
    if($parametre == $nbPages) {
        echo $prev;
    }
    // S'il s'agit des autres pages
    if($parametre > 1 && $parametre < $nbPages) {
        echo $prev;
        echo $next;
    }
}
}
```

Il suffit ensuite de lancer la fonction dans la section `<head>...</head>` en ajoutant les deux arguments demandés (voir code suivant). Tout d'abord, il faut préciser le nom du paramètre d'URL qui contient le numéro de la page (`p` ou `page` en général). Ensuite, il faut calculer le nombre total de pages de résultats et l'afficher en second argument de la fonction. La fonction affichera alors les balises `<link />` adéquates en fonction du numéro de page.

```
<?php managePrevNext(PARAMETRE_NAME_PAGE, NOMBRE_DE_PAGES); ?>
```

Sachez également qu'il est tout à fait possible de conjuguer l'utilisation des relations canoniques et celles des pages précédentes ou suivantes mais il faut absolument veiller à ne pas se tromper dans les URL.

En définitive, retenons que la mise en place des balises canoniques, précédentes ou suivantes n'est pas toujours aisée mais peut s'avérer déterminante si nous souhaitons éviter des sanctions causées par du *duplicate content* ou des problèmes de DUST (voir chapitre 3).

Les CMS courants proposent souvent des systèmes natifs d'installation de ces balises (notamment WordPress dans ses versions récentes) ou disposent généralement d'extensions de qualité qui permettent de réduire les risques avec le moins d'efforts possible. Il n'est donc pas toujours nécessaire d'effectuer le travail manuellement pour que l'ensemble fonctionne.

Multilinguisme avec hreflang

Optimiser le référencement des sites multilangues n'est pas simple en règle générale. Nous ne rentrerons pas dans les détails obscurs de ces spécificités, mais il est bon de rappeler les grands principes.

- Il est recommandé de créer un sous-domaine ou un répertoire par langue, voire d'acheter un nom de domaine avec des extensions de pays pour chaque langue d'un site. Chaque option a ses avantages et inconvénients mais, en général, ce sont les sous-domaines qui remportent la mise pour des raisons de coûts et de facilité de gestion.
- Il convient de ne jamais mélanger de contenus en plusieurs langues au sein des pages web, car c'est le meilleur moyen pour noyer les mots-clés de chaque langage. Dans le même esprit, il est fortement déconseillé de créer des liens vers différentes langues sans avoir encore traduit les contenus ; sinon, c'est la porte ouverte aux contenus dupliqués.

Pour aller plus loin et améliorer la compréhension des sites multilangues, Google a créé le 5 décembre 2011 une balise `<link/>` avec un nouvel attribut `hreflang` (source : https://goo.gl/PTwht5). Cette balise HTML indique aux robots d'indexation les versions alternatives de chaque page, mais en fonction de leur langue d'origine.

Le système fonctionne un peu comme si le moteur possédait des versions canoniques de pages, dont chaque version est indexable mais bien différenciée dans l'index de Google. L'attribut n'est pas accompagné pour autant de l'attribut `rel="canonical"`, mais par la valeur `rel="alternate"` ainsi qu'un attribut `href` indiquant l'URL de chaque version de site (documentation : https://goo.gl/SMYP97).

Voici un exemple de code utilisant les attributs `hreflang` :

```
<link rel="alternate" hreflang="fr" href="http://www.site.com"/>
<link rel="alternate" hreflang="fr-ca" href="http://ca.site.com"/>
<link rel="alternate" hreflang="nl-be" href="http://be.site.com"/>
<link rel="alternate" hreflang="en" href="http://en.site.com"/>
<link rel="alternate" hreflang="zh-TW" href="http://zh.site.com"/>
```

Comme vous le constatez, le code de langue se décompose en deux parties. La première représente la langue ciblée et la seconde la composante régionale (optionnelle). Par exemple, si nous ciblons des contenus en français pour les Suisses, il faut écrire `"fr-CH"`.

Si vous utilisez un outil de sélection de langues ou une redirection automatique en fonction de la source géographique des visiteurs, il faut ajouter une autre balise `<link/>` avec un attribut `hreflang` portant la valeur `"x-default"` (langue par défaut), comme ceci :

```
<link rel="alternate" hreflang="x-default" href="http://www.site.com"/>
```

L'App Indexing : indexer des liens profonds d'applications mobiles

Qu'est-ce que l'App Indexing ?

L'App Indexing est un mécanisme créé par Google pour indexer les liens profonds d'applications mobiles, à savoir des URL de sections ou de pages internes à une application. L'officialisation du système date du 31 octobre 2013 (source : https://goo.gl/h0RBGr) mais n'était consacrée qu'à Android dans les premiers temps, avant de s'ouvrir peu à peu à iOS 7 puis iOS 8. Depuis la fin 2015, l'App Indexing est compatible uniquement avec Android et iOS 9 (et suivants).

L'objectif de cette fonctionnalité est d'indexer des contenus internes à une application pour les proposer dans les résultats mobiles quand cela est pertinent sur une requête donnée. Ainsi, le système affiche un lien organique avec un titre, une courte description, le nom de l'application, voire d'autres informations (notes et autres extraits enrichis éventuels). Souvent, l'application est accompagnée d'un bouton, qui peut prendre deux formes :

- *Installer* si l'app mobile n'est pas déjà présente dans la tablette ou le smartphone ;
- *Ouvrir* pour rejoindre la page profonde de l'application si cette dernière est déjà installée.

Les liens profonds d'applications ne sont affichés que lorsque les requêtes sont assez centrées sur des applications existantes voire sur leur nom, mais cela s'améliore de mois en mois, offrant de plus en plus régulièrement des liens de ce type dans les SERP.

Figure 1-15

Exemple de lien profond d'app mobile affiché dans les SERP

Le 30 septembre 2015, Marisa Moeva, Webmaster Trends Analyst, a affirmé que Google accorde un faible boost SEO aux utilisateurs de l'API App Indexing (source : http://goo.gl/fazNzz). En réalité, cela est déjà le cas depuis le mois d'avril 2015 et permet aux liens web d'être mieux positionnés grâce à l'utilisation de l'App Indexing.

Pas de boost SEO pour iOS ?

Le 30 octobre 2015, Marisa Moeva et John Mueller ont indiqué que le boost SEO accordé depuis avril aux liens profonds d'apps mobiles sur Android ne devrait pas être appliqué pour iOS (source : http://goo.gl/O0iiSP). Ce choix est étonnant et rien ne permet de confirmer que la revalorisation ne finira pas par être également accordée aux concepteurs d'applications mobiles pour les systèmes Apple.

Il ne faut pas nécessairement mettre en place l'App Indexing pour obtenir un meilleur positionnement des liens profonds d'applications dans les SERP des mobiles. La finalité est surtout de mieux indexer les pages internes des apps pour gagner de la visibilité et de nouveaux biais d'entrée afin d'obtenir des téléchargements d'applications ou d'augmenter le nombre d'utilisateurs.

Google met en place l'App Streaming

Google a annoncé le 18 novembre 2015 le développement de l'App Streaming, un système permettant au moteur d'afficher des apps mobiles via les SERP mobiles sans avoir à télécharger l'application (source : http://goo.gl/xxtWzm). L'App Indexing sert d'origine à l'App Stream car Google consolide une application intermédiaire (en « streaming ») à partir de sa connaissance des liens profonds et des contenus. Le premier usage officiel date du 3 décembre 2015 quand Google a lancé les Trial Run Ads, c'est-à-dire un premier système d'App Streaming pour les jeux vidéo sur mobile (source : http://goo.gl/XFbl0k). Ainsi, les mobinautes peuvent tester pendant 60 secondes un jeu avant de télécharger éventuellement l'application mobile.

L'App Streaming est un système très ingénieux et pratique pour les mobinautes, qui tire donc ses sources grâce à l'App Indexing et au crawl des liens profonds d'applications. Un bémol peut néanmoins être apporté à ce système, car si ce type de pré-rendu se multiplie, le nombre de téléchargements d'applications risque de chuter fortement (certains utilisateurs s'arrêteront aux tests voire trouveront l'information directement sans avoir à télécharger l'application complète)…

Procédure d'indexation des liens profonds d'applications mobiles

La documentation sur l'App Indexing fournie par Google est assez complète et devrait vous aider à mieux indexer les liens profonds d'applications. Comme tout cela est encore assez changeant depuis les dernières nouveautés de l'API et l'intégration d'iOS 9, nous ne rentrerons pas dans des détails techniques qui pourraient vite être obsolètes ou faussés. Voici les liens vers la documentation pour suivre les informations officielles à jour : https://goo.gl/DxyUcG et https://goo.gl/7Y7yV7.

Il faut savoir qu'à l'origine, Google n'indexait les liens profonds d'applications que s'ils avaient une équivalence sur un site web. Ainsi, il pouvait faire le lien entre les deux versions et mettre en avant la version des applications dans les SERP mobiles. Cela s'est ouvert et depuis la fin 2015, Google accède aussi aux liens inédits présents uniquement dans les apps mobiles. Pour ce faire, il faut remplir un formulaire en ligne et donner des informations sur les URL uniques : https://goo.gl/Df8lLs.

La documentation indique qu'il faut respecter plusieurs étapes pour mettre en place l'App Indexing et bien faire fonctionner l'ensemble du système. Prenons comme exemple la méthodologie pour les applications Android, qui profitent de l'indexation profonde depuis des années et bénéficient du petit bonus de positionnement :

- ajouter des `intent filters` pour les URL en HTTP au sein du fichier `AndroidManifest.xml` de l'application Android ;
- déclarer et valider le site officiel de l'application. Il s'agit en général d'indiquer à Google l'URL du site initial pour qu'il fasse la liaison entre les pages web et les pages d'application existantes. Cela peut se faire avec la Search Console ou la Developer Console de Google Play (https://goo.gl/ubYxID) ;
- effectuer un suivi, faciliter l'indexation avec l'API et corriger les éventuelles erreurs rencontrées.

Voici un exemple de fichier `AndroidManifest` partiel avec une mise en œuvre d'App Indexing :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.AppliExemple">
<application>
    <activity android:name="com.example.android.AppliExemple">
    <!—Bloc intent-filter pour l'App Indexing -->
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="http" />
        <data android:scheme="https" />
        <data android:host="www.appliexemple.fr" />
    </intent-filter>
    </activity>
          <!-- Metadonnée pour indiquer un fichier noindex.xml -->
          <meta-data android:name="search-engine" android:resource="@xml/noindex"/>
</application>
...
</manifest>
```

Enfin, sachez qu'il est possible de gérer l'indexation des contenus avec un fichier `noindex.xml`, une sorte d'équivalent du fichier `robots.txt` pour les applications Android, dont voici un exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<search-engine xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Exclusion des notifications -->
    <noindex android:value="notification"/>
    <!-- Exclusion d'une page précise -->
    <noindex uri="http://site.com/page-masquee"/>
    <!-- Exclusion des URL commençant par un préfixe defini -->
    <noindex uriPrefix="http://site.com/prefixe-masque"/>
</search-engine>
```

Une fois l'implantation de l'App Indexing effectuée (voire celle de son API), vous pouvez tester la bonne marche du système avec un outil disponible à cette URL : https://goo.gl/JyXzs2. L'année 2016 devrait marquer un tournant dans l'indexation des liens profonds d'applications mobiles. Bien que nous n'ayons pas donné tous les détails dans cet ouvrage, il est recommandé de s'intéresser à ce sujet si vous possédez des apps nomades.

Faire du SEO local

Particularités du SEO local

La recherche locale est un des aspects les plus importants des moteurs de recherche actuels. Aux origines du Web, Internet était considéré comme l'ouverture sur le monde et l'intérêt pour les informations proches de nous était assez limité. Les premiers outils de recherche misaient donc peu sur ce phénomène. Plus tard, avec l'essor du nombre de pages web indexées couplé à l'évolution des sociétés, les concepteurs ont ressenti le besoin de géolocaliser des informations, puis les résultats de recherche.

Google a été le premier à vraiment se pencher sur la question de la recherche locale, mettant en avant des « lois » bien connues dans le métier de la presse. En effet, en proposant des résultats localisés, un moteur de recherche répond à la loi de proximité géographique (plus une information est proche de nous, plus elle est susceptible de nous intéresser). Parallèlement à cela, les efforts des moteurs comme Google pour favoriser la fraîcheur des contenus ont permis d'ajouter à ce phénomène la loi de proximité temporelle (plus une information est récente, plus elle est pertinente pour le lecteur).

Tout l'intérêt du SEO local repose sur ces deux aspects : il faut répondre avec de l'information pertinente et récente, tout en étant le plus proche possible de nos lecteurs et clients potentiels.

Algorithmes spécifiques

Google a très vite compris l'intérêt de la recherche locale mais, sur le plan technique, tout ne s'est pas fait en un jour. Google Local est né le 17 mars 2004 en version bêta (source : http://goo.gl/rl93Bn) et il fallait passer par une URL spécifique http://local.google.com. Dès le 8 février 2005, Google Maps naît à son tour et modifie le paysage du Web avec son système de cartes (source : https://goo.gl/2lnfzY). C'est le début de la recherche locale sur Google…

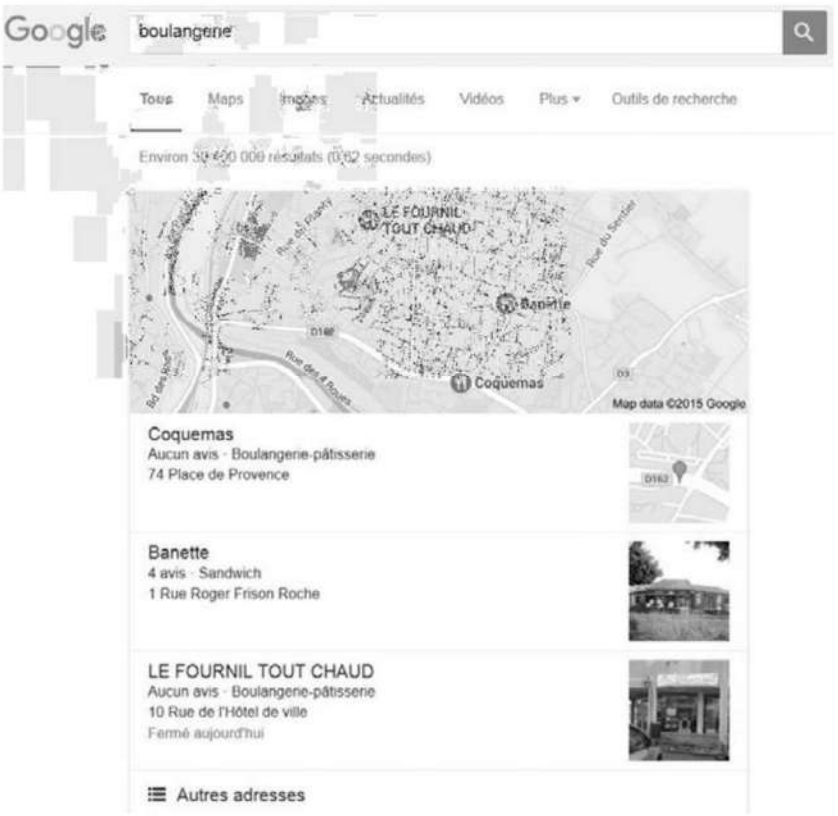
Parallèlement à ces outils, le géant américain lance le 15 mars 2005 Google Local Business Center pour que les internautes puissent inscrire leur adresse locale dans Maps (source : http://goo.gl/B4EkCl). Cet outil fort intéressant pour les entreprises changera ensuite plusieurs fois de nom, devenant Google Places (appelé Google Adresses en France) en avril 2010 (source : http://goo.gl/ixrzzC), puis Google+ Local en mai 2012 (source : http://goo.gl/foObtv). La dernière étape a été marquée par l'officialisation de Google My Business le 11 juin 2014 (source : http://goo.gl/UY5fru), accessible à l'adresse https://www.google.com/business.

Les SERP du moteur de recherche ont beaucoup évolué au fil des ans, affichant d'abord une liste de liens localisés, puis une onebox dédiée avec plusieurs affaires locales mises en avant, puis plus récemment

seulement trois résultats locaux prenant plus d'espace dans l'écran. Google ajustant sans cesse cet affichage, il serait trop long de dresser un historique précis de ces modifications.

Figure 1-16

Trois résultats locaux affichés dans les SERP en 2016



La recherche locale était surtout issue du travail d'indexation des entreprises dans Google Maps à l'origine, mais ceci a bien évolué avec deux algorithmes marquants du moteur de recherche : Venice et « Pigeon ».

Google Venice a été annoncé officiellement le 27 février 2012 (source : http://goo.gl/zGwo5f). Cette mise à jour a eu pour objectif de favoriser les résultats locaux dans les SERP. Par conséquent, les pages web et sites locaux géolocalisés ont obtenu une sorte de bonus de positionnement pour être mieux classés dans les résultats de recherche de Google. C'est la première fois que le moteur a mis en avant volontairement les résultats locaux, marquant la fin de la recherche « neutre » d'antan.

Google « Pigeon » (nom attribué par le site SearchEngineLand) est né le 24 juillet 2014 (source : http://goo.gl/qz0FRO) aux États-Unis, avant d'être déployé en France début juin 2015. Cet algorithme a pour but de favoriser les requêtes contenant des recherches locales, comme « hôtel Paris », « boulangerie Nantes », etc. Dans les faits, l'algorithme va plus loin puisqu'il permet à Google de coupler les critères locaux aux facteurs classiques de ranking. Par conséquent, la recherche locale est vraiment couplée aux critères habituels du moteur, pour ne faire plus qu'un en quelque sorte.

Figure 1-17

Officialisation du déploiement de Google « Pigeon » en France

Pour conclure sur les algorithmes de Google relatifs au SEO local, nous pouvons dire que les résultats géolocalisés sont mis en avant par le moteur quand cela semble pertinent. Pour ce faire, la firme utilise plusieurs systèmes en parallèle : une analyse fine des requêtes, une géolocalisation des internautes (ou plutôt du DSLAM relié au poste sur lequel la recherche est effectuée) et des pages web (grâce au crawl notamment), une revalorisation des résultats locaux selon la source de recherche.

Et la recherche locale sur Bing ?

Bing détient aussi son propre système de recherche locale, au même titre que d'autres moteurs de recherche d'ailleurs. Sur Bing, il faut bien différencier ce qui se fait aux États-Unis, via Bing Maps et Bing Places for Business (source : https://www.bingplaces.com), de ce qui se fait en France par exemple.

Bing Places for Business est l'équivalent de Google My Business et permet aux internautes d'enregistrer leurs adresses locales pour apparaître dans Bing Maps et être plus présents dans les SERP sur des requêtes spécifiques. En France, ce système n'est pas encore déployé complètement ; il faut donc utiliser les Pages Jaunes pour apparaître dans les résultats locaux du moteur de Microsoft.

Interfaces et outils de géolocalisation

Nous avons déjà évoqué rapidement les outils et services locaux des moteurs de recherche ; nous allons voir rapidement comment les utiliser à bon escient afin de faire apparaître les entreprises locales dans les services de cartographie de Google et Bing.

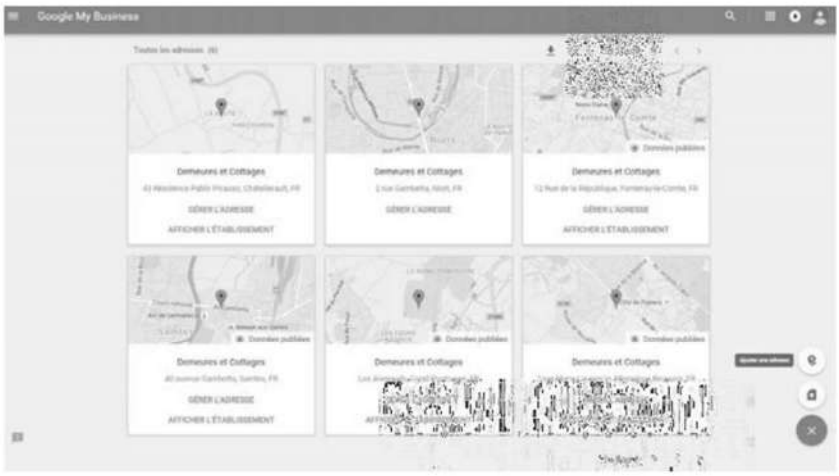
Commençons avec Google My Business, un outil très complet pour intégrer une ou plusieurs adresses pour une même entreprise, avec de nombreuses options disponibles :

- titre et description de l'entreprise ;
- coordonnées (adresse, numéro de téléphone et URL du site web associé) ;
- horaires d'ouverture généraux ;
- horaires d'ouverture et de fermeture exceptionnels ;
- catégorie d'entreprise ;
- photos de l'activité locale ;
- visite virtuelle (option non obligatoire à faire réaliser par un photographe agréé Google).

L'outil est accessible à partir de plusieurs URL sur Google, dont la plus connue est https://business.google.com. En général, les utilisateurs ne possèdent qu'une seule adresse, mais il est possible d'avoir plusieurs bureaux ou locaux pour une même enseigne, comme dans la capture suivante.

Figure 1-18

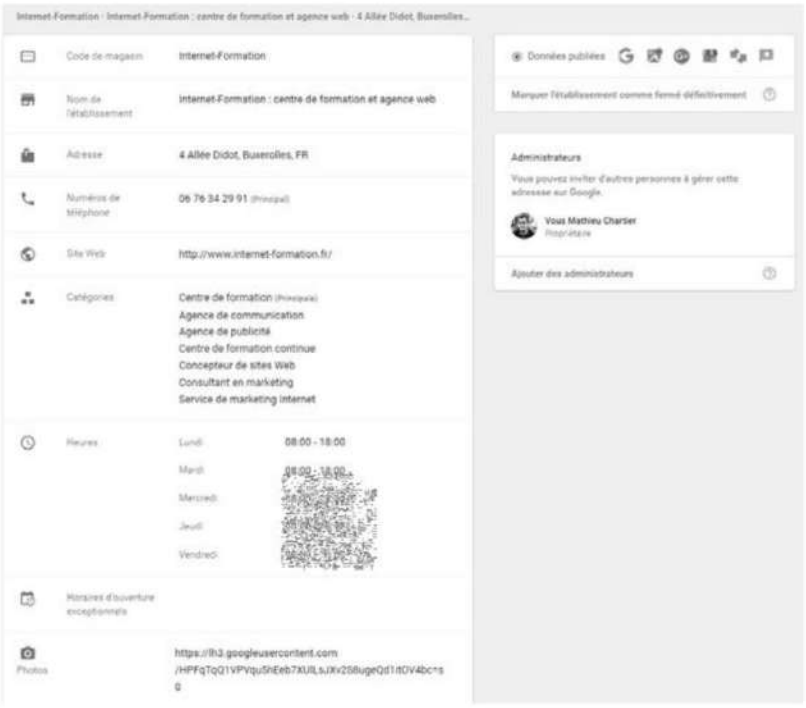
Enregistrement de plusieurs adresses d'une même enseigne



L'inscription est gratuite mais il faut posséder un compte Google pour profiter du service en ligne. Une fois l'inscription réalisée, vous pouvez ajouter une adresse ou plusieurs en cliquant sur le bouton en forme de +, ou en passant par le menu déroulant en haut à gauche de l'écran. Google vous enverra un courrier postal à l'adresse indiquée dans l'outil avec un code de validation pour finaliser l'inscription définitivement.

Figure 1-19

Exemple de fiche Google My Business remplie à 100 %



L'outil permet de gérer un workflow, c'est-à-dire d'autoriser plusieurs administrateurs à gérer les fiches Google My Business quand cela semble judicieux. Par défaut, seul le compte de départ est administrateur. Il convient de remplir un maximum de champs dans le formulaire d'ajout d'une activité locale pour améliorer sa visibilité, mais aussi pour répondre idéalement à toutes les questions que les internautes et mobinautes peuvent se poser lorsqu'ils tapent des requêtes locales ou recherchent dans Google Maps.

Chaque outil local de Google est indépendant, mais l'écosystème est bien conçu car tout est bien huilé et relié. En effet, Google Maps récupère les adresses de Google My Business et ce dernier s'associe aux avis déposés via Google+, tout cela étant parfois affiché dans les SERP du moteur de recherche classique. Il faut donc être présent sur l'ensemble de ces outils pour bénéficier pleinement des atouts du SEO local, mais n'ayez crainte, tout part justement d'une bonne fiche Google My Business.

Si vous devez inscrire plusieurs adresses pour une même marque ou enseigne, cela est possible en injectant manuellement une fiche supplémentaire (via le bouton +) ou directement en intégrant un fichier Excel. Plusieurs champs sont à remplir pour accélérer l'intégration, mais il convient de se référer à la documentation pour ne pas faire d'erreur (source : https://goo.gl/ScBjzU).

Figure 1-20

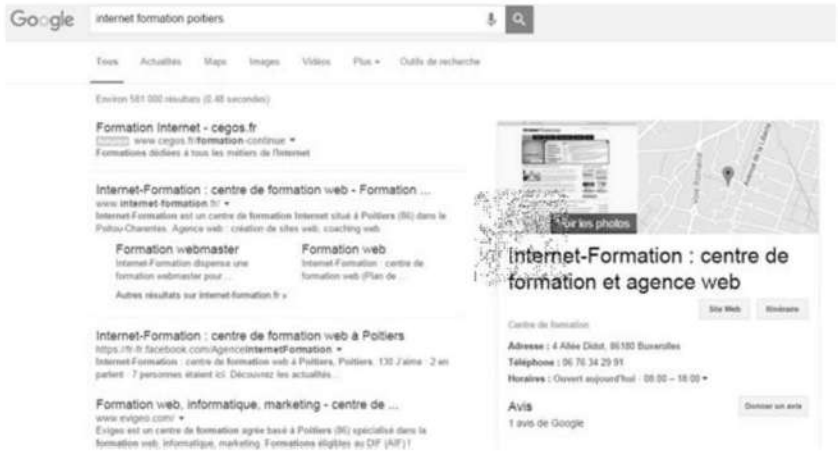
Exemple de fichier Excel pour intégrer plusieurs adresses

	Store code	Business	Address Line 1	Address Line 2	City	District State	Country	Postal Code	Primary phone	Additional phones	Website	Primary category
2	demeures0	Demeures et	2 rue Marie Laurencin		Migné-au-		FR	86550	05 49 55 35 99		http://www.demeures-	Constructeur
3	demeures0	Demeures et	43 Résidence Pablo		Châtellerault		FR	86100	05 49 02 57 47		http://www.demeures-	Constructeur
4	demeures0	Demeures et	2 rue Gambetta		Niort		FR	79000	05 49 55 35 99		http://www.demeures-	Constructeur
5	demeures0	Demeures et	Les Avenauds		Gond-		FR	16100	05 45 84 63 64		http://www.demeures-	Constructeur
6	demeures0	Demeures et	40 avenue Gambetta		Saintes		FR	17100	05 46 74 62 60		http://www.demeures-	Constructeur
7	demeures0	Demeures et	12 rue la République		Fontenay-le-		FR	85200	02 51 52 83 13		http://www.demeures-	Constructeur

Retenez que le plus important est de bien remplir les fiches Google My Business, car les informations peuvent ensuite être indexées et ressortir dans les SERP, dans Google Maps ou dans Google+. Ce sont de nouveaux biais d'entrée pour vos visiteurs…

Figure 1-21

Onebox locale avec données issues de Google My Business



Sur Bing, il est possible d'indexer nos adresses locales avec Bing Places for Business, mais uniquement dans certains pays. La France ne fait malheureusement pas encore partie de la liste et Bing ne communique pas sur une future intégration à ce jour. Bing Adresses est accessible à https://www.bingplaces.com.

Il faut ajouter une adresse et remplir un formulaire d'inscription, à l'instar de ce qui se fait sur Google My Business, bien que l'outil dispose d'un petit peu moins d'options.

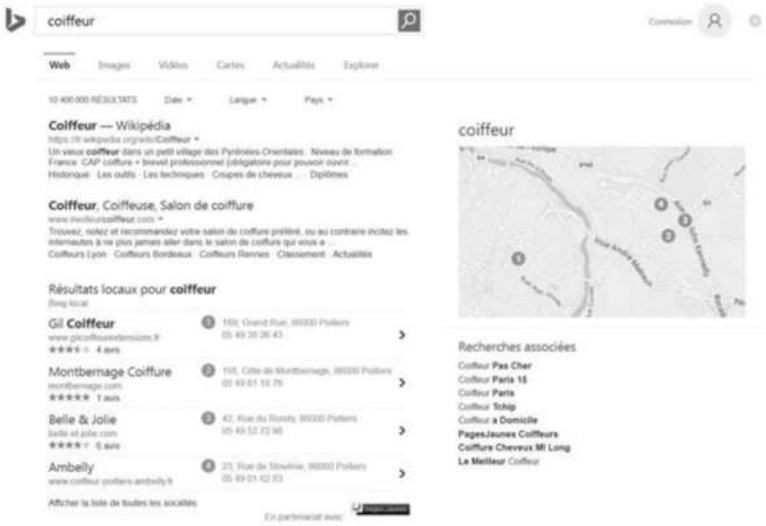
Figure 1-22

Ajout d'une nouvelle adresse dans Bing Places for Business

En France, Bing administre sa recherche locale et ses cartes en partenariat avec Pages Jaunes dans les SERP du moteur de recherche ; c'est pourquoi Bing Places n'est pas l'outil à privilégier, mais simplement une inscription dans les Pages Jaunes. Ainsi, vos adresses pourront apparaître dans les résultats de recherche sur des requêtes spécifiques au SEO local de Bing.

Figure 1-23

Exemple de recherche localisée sur Bing



La recherche géolocalisée de Bing, tout du moins en France, est bien moins évoluée que celle de Google. Certes, les résultats locaux ainsi que certaines requêtes affichent principalement des sources proches géographiquement, mais l'algorithmie est moins pertinente en règle générale.

Contourner la géolocalisation forcée

Google autorisait la modification de localisation dans les options de recherche de son moteur, mais depuis le 1er décembre 2015, il n'est plus possible de modifier le pays et la ville sources de la recherche (source : http://goo.gl/qqys7t). De ce fait, toutes les recherches effectuées actuellement sur Google sont forcément géolocalisées en fonction de l'endroit où vous recherchez.

Le SEO local prend alors encore plus de poids et il convient de faire bien attention à ne pas trop caler de mots-clés locaux dans les pages si vous ne souhaitez pas être favorisé dans des lieux spécifiques. Dans le même temps, les optimisations avec des mots-clés locaux chers à Google Venice et « Pigeon » vont être encore plus importantes pour mieux ressortir localement.

Si d'aventure vous souhaitez effectuer un suivi de positionnement neutre, et donc non géolocalisé, il est possible de placer le paramètre `"&near=NOM_V...L"` à la fin des URL de Google. Cela ne fonctionne pas systématiquement mais permet généralement d'effectuer des requêtes dans d'autres lieux que le vôtre. Toutefois, la meilleure solution est de passer derrière un proxy ou un VPN avec des adresses IP localisées loin de votre source géographique, bien que cela puisse avoir un certain coût…

Dans le même esprit, il est possible de changer de pays en tapant une URL comme http://www.google.com/webhp?cr=countryUS. Il faut juste changer éventuellement l'extension `.com` de Google et le code langue `US` avec ce que vous préférez pour contourner la géolocalisation par pays.

Enfin, sachez que deux autres astuces aident à passer outre la localisation par IP de Google. Pour ce faire, il est possible d'utiliser l'URL https://encrypted.google.com ou, encore mieux, la version *No Country Redirect* de Google.com (qui bloque les redirections vers google.fr par exemple), via l'URL https://www.google.com/ncr. Dans ces deux cas, c'est uniquement le moteur général qui est accessible et c'est donc plus neutre qu'un changement de zone géographique.

Positionnement sur des requêtes locales

Le positionnement des pages web sur des requêtes locales connaît quelques spécificités. Cela sort un peu du contexte de ce chapitre sur l'indexation mais propose une transition parfaite en vue de la seconde partie de cet ouvrage ; c'est pourquoi nous allons terminer ce chapitre en insistant sur quelques facteurs associés au ranking local dans les SERP.

Google n'a jamais donné d'indications précises au sujet des points à optimiser pour améliorer le positionnement local, mais les algorithmes Venice et Pigeon ont donné des prémices de réponses, avant que des études assez approfondies viennent compléter tant bien que mal les impressions générales ressenties par les experts du métier. L'analyse la plus complète est certainement celle fournie par Moz.com fin septembre 2015 (source : https://goo.gl/5XdtWL). Il s'agit d'un grand sondage réalisé sur les critères de référencement local auprès d'au moins quarante experts SEO dans le monde. Cela fait ressortir les grandes tendances ainsi que les facteurs qui semblent pris en compte par Google.

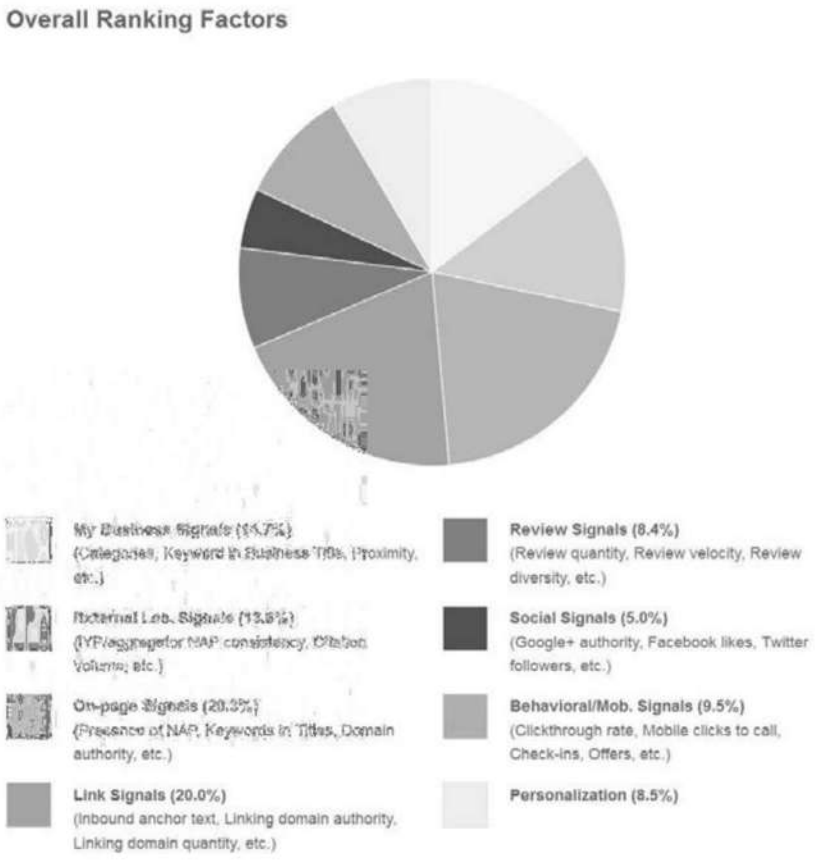
L'analyse des réponses a démontré que les critères classiques de ranking jouent vraiment un rôle dans le positionnement local, comme le présuppose l'algorithme Pigeon, mais que d'autres facteurs plus

spécifiques sont pris en compte. Sur l'ensemble des questions posées, l'étude de Moz a fait ressortir au moins cinq grandes familles de critères, dont l'ordre établi par les experts est celui-ci :

- signaux *on-page* : critères classiques de positionnement ;
- *netlinking* : obtention de liens entrants avec des ancres de liens optimisées et un profil de liens de qualité ;
- signaux issus de Google My Business : mots-clés du titre, de la description, des catégories d'intégration (...), proximité entre la recherche et l'entreprise...
- facteurs externes : nombre d'avis des utilisateurs sur l'entreprise, qualité de ces avis…
- critères comportementaux : taux de clics, durée des visites, *check-in*, nombre d'appels via les mobiles…

Figure 1-24

Résultats de l'étude de moz.com sur le SEO local



Ces familles de critères ne se fondent que sur l'expérience et les retours des spécialistes du métier, mais Google n'a confirmé que certains critères le 1er décembre 2015 (source : http://goo.gl/m2aYcr). En effet, Google considère que la recherche locale résulte de quatre familles de facteurs :

- pertinence de la fiche Google My Business par rapport à la requête des utilisateurs ;
- importance du trafic vers la fiche locale de l'entreprise ;

- distance entre l'entreprise et la source géographique de la recherche ;
- historique des recherches : nombre de fois où la fiche Google My Business a été utile historiquement sur la base de sa pertinence, de son importance et de sa distance.

Quand Google change sa déclaration...

Le dernier point évoqué par Rahul J., le porte-parole de Google qui a annoncé les critères de la recherche locale, a été modifié par Google au dernier moment. En effet, le critère de l'historique des recherches était différent à l'origine ; il impliquait le nombre de clics historiquement enregistrés sur la fiche Google My Business sur des mots-clés donnés (source : http://goo.gl/xKOqrx).

Nous pouvons être surpris par ce revirement de situation qui laisse planer un doute sur la réalité des facteurs pris en compte. Soit Rahul J. s'est trompé et Google a donc décidé de rectifier le tir, soit il avait raison mais la firme ne voulait pas que le critère soit connu dans le monde. Les deux hypothèses se tiennent et dans le second cas, nous pourrions imaginer que Google voudrait éviter du spam avec des clics générés automatiquement par des robots sur des requêtes locales.

Google n'a pas évoqué les critères classiques du positionnement dans son intervention sur le SEO local, mais il a tenu à rappeler les points essentiels à optimiser :

- sélectionner la bonne catégorie pour les activités locales ;
- partager la page Google My Business afin d'améliorer sa visibilité et son historique ;
- obtenir des avis et de bonnes notes pour améliorer la valeur de la fiche d'entreprise ;
- partager toutes les informations nouvelles et faire vivre la page au maximum (photos, horaires d'ouverture et de fermeture exceptionnels...).

Vous savez désormais ce que Google favorise en termes de SEO local, nous allons donc entrer dans le vif du sujet en évoquant tous les facteurs génériques de positionnement pour les moteurs.

2

Optimiser le positionnement par la technique

Nous avons vu dans le précédent chapitre qu'il était indispensable de penser à l'indexation des pages avant même de réfléchir à les positionner dans les résultats des moteurs de recherche.

Ceci étant dit, il nous faut maintenant savoir ce qu'il est possible de faire pour optimiser au mieux le classement des pages web afin d'augmenter considérablement la visibilité des sites mais aussi le nombre de visites.

Il existe de nombreux livres complets sur le sujet qui vous permettront de maîtriser pleinement chacun des critères pris en compte par les moteurs en termes de positionnement web. Nous ne traiterons ici que de cas spécifiques que la technique permet d'optimiser mieux que la théorie, aussi qualitative soit-elle.

Rappels des fondamentaux

Méthodologie du positionnement

Savoir se référencer est une chose, mais réussir à positionner les pages dans les SERP en est une autre. Il n'est pas toujours aisé de maîtriser les deux étapes ou d'obtenir d'aussi bonnes conclusions dans les deux cas. En effet, l'indexation est parfois difficile car les robots peuvent rapidement se montrer susceptibles. Mais lorsqu'il s'agit de classer les pages dans les SERP, ce phénomène est démultiplié et nous ramène à quelques vérités :

- les algorithmes de pertinence sont de plus en plus pointus, précis et efficaces ;
- la concurrence dans chaque domaine est grandissante et les places libres de plus en plus rares ;

- les facteurs de positionnement sont nombreux mais pas toujours applicables selon les sites web que nous contrôlons ;
- les vérités d'aujourd'hui ne sont pas toujours celles de demain.

Partant de ces postulats, nous savons déjà que la méthode à suivre pour positionner les pages est incertaine et que rien ne peut garantir de bons résultats, même si nous faisons en sorte de respecter à la lettre chaque facteur pris en compte par les moteurs.

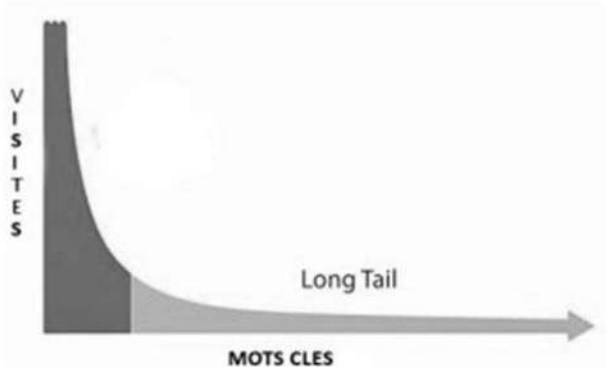
Il est primordial d'intégrer cela car obtenir un mauvais classement n'est malheureusement pas toujours totalement de notre ressort. Nous devons justement savoir réagir promptement et efficacement pour rétablir l'ordre et gagner des positions sur certaines requêtes.

La méthodologie du positionnement présente des incertitudes mais une chose est sûre, elle doit suivre plusieurs étapes pour fonctionner :

- trouver les bons mots-clés est essentiel car ce sont sur ces termes et ces expressions que les moteurs vont s'appuyer pour positionner le site dans les SERP ;
- travailler la longue traîne, à savoir des expressions de plusieurs mots (au sens plus précis que les mots-clés majeurs), afin de pouvoir ressortir sur un nombre de requêtes bien plus larges et souvent bien mieux ciblées que les expressions généralistes ;

Figure 2-1

Concept de la longue traîne (Chris Anderson, rédacteur en chef du magazine Wired)



- optimiser les critères internes aux pages pour que les codes sources soient dans des conditions idéales pour convenir aux robots et aux algorithmes de pertinence ;
- profiter des facteurs de positionnement externes aux pages tels que le PageRank ou l'usage des réseaux sociaux pour conforter voire booster encore davantage le classement des pages.

Chaque moteur de recherche présente des interprétations variables des codes, des contenus et des critères externes mais dans leur ensemble, les principaux outils de recherche du marché s'appuient sur des facteurs similaires pour positionner les pages. Nous n'aurons donc pas à tout changer en fonction des moteurs et des marchés que nous ciblons.

Quelques différences entre les moteurs de recherche

Concernant les différences entre les moteurs de recherche, nous pouvons noter que Bing insiste davantage sur le comportement des internautes (nombre de visites, taux de rebond, nombre de pages vues…) que Google dans son classement final, mais il est également moins insistant sur le suivi des auteurs de contenu. Ces petites différences n'ont jamais impliqué les fondamentaux du positionnement…

Les optimisations internes

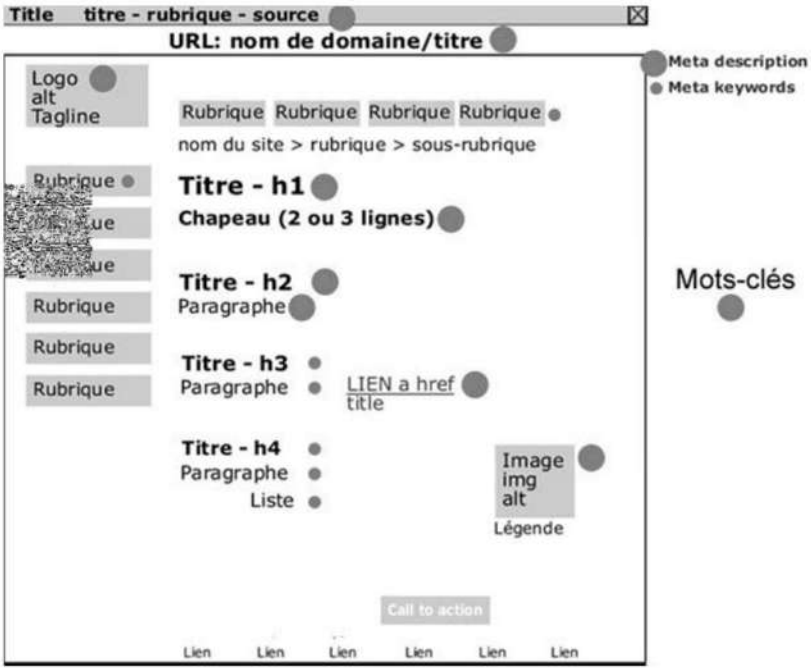
Nous avons rappelé qu'il était primordial de trouver de bons mots-clés à inscrire dans les pages web car ce sont eux qui forment le socle du positionnement pour les moteurs de recherche. En effet, les robots ne voient que des codes sources et extraient les contenus afin de les traiter dans un second temps. Ceci leur permet de qualifier les mots-clés contenus dans chaque page afin de noter chaque document à sa juste valeur en fonction de requêtes précises.

Une fois que nous possédons nos listes de mots-clés, il ne reste qu'à les placer dans des zones « chaudes » pour les valoriser, les mettre en exergue et donner aux robots de quoi manger. Plus nous comblons leur appétit et leur soif de pertinence, plus les pages web risquent de remonter sur des expressions précises. Rappelons donc une liste de facteurs pris en compte pour valoriser les termes clés…

Voici une illustration d'Isabelle Canivet, spécialiste de la rédaction web et auteure du livre *Bien rédiger pour le Web*, paru aux éditions Eyrolles. Elle pointe l'ensemble des « zones chaudes » des pages web, c'est-à-dire les blocs dans lesquels les mots-clés doivent se trouver pour avoir un impact sur le classement final.

Figure 2-2

Optimisation interne des pages selon Isabelle Canivet



La balise <title>

Le principal critère de positionnement reste le titre des documents qualifié par les balises `<title>` placées dans la section `<head>` des pages HTML. Son rôle n'est pas toujours primaire dans le classement final mais il faut avouer que les titres ont un impact dans une large majorité de cas, quel que soit le moteur ciblé.

Les titres doivent être uniques et assez courts, utiliser des termes relatifs au contenu des pages optimisées et ne présenter quasiment aucun *stop words* (articles, conjonctions…). Nous savons que Google présente 70 caractères dans les SERP contre 65 pour Bing. Nous devons donc nous tenir à ne pas dépasser outre mesure cette longueur maximale, ce qui représente environ sept à huit mots-clés.

La plupart des CMS, Frameworks et autres plates-formes web génèrent les titres à partir des intitulés de menus ou des titres des articles. Cela permet de gagner du temps mais c'est souvent loin d'être la meilleure option pour optimiser le positionnement. Il est donc préférable d'opter pour des méthodes qui permettent de personnaliser entièrement les titres pour chaque page des sites web (extensions, codes personnels, modifications des thèmes natifs…).

Affichage et gestion des balises <title> par les moteurs

Les titres des documents rédigés entre les balises `<title>…</title>` sont très souvent mis en avant dans les moteurs de recherche pour présenter les pages. S'ils sont manquants ou jugés comme peu pertinents (ou *spammy*), les moteurs génèrent parfois leurs propres titres de remplacement…

Les métadonnées

Les métadonnées sont des informations accompagnant un fichier. Elles permettent d'apporter des précisions sur les documents. Dans les pages web, elles sont indiquées dans les balises `<meta />` dont les variantes sont nombreuses.

Aucune n'a d'intérêt direct pour le positionnement, bien que ce fût le cas par le passé. De nos jours, il ne persiste que quelques balises de métadonnées intéressantes.

- La balise meta `description` permet d'ajouter un court texte qui résume le contenu des pages dans les SERP. À l'instar des titres mis en exergue dans les pages de résultats, les descriptions sont présentées aux internautes pour qualifier les contenus et inciter aux clics. Nous devons donc les travailler afin d'optimiser le taux de clics vers les pages cibles.
- Les balises meta `keywords` sont destinées à ajouter des listes de mots et expressions clés afin de préciser aux moteurs les termes qui qualifient le mieux les contenus internes. Il est admis qu'ils n'ont plus d'influence sur le positionnement. Ils pourraient même donner des indications aux moteurs concernant les mots-clés sur lesquels nous souhaitons être classé, ce qui risque de se retourner contre nous en définitive. Libre à vous de les remplir ou non mais quoi qu'il en soit, ne passez pas des heures pour cela tant ce facteur a perdu de son importance.
- Enfin, les métadonnées `robots` permettent de bloquer l'indexation au même titre que les fichiers `robots.txt`. Elles doivent être placées dans les pages à déréférencer avec un attribut `content` dont la valeur est `noindex, nofollow` si votre souhait est de bloquer l'accès aux robots. Son utilité est à nuancer si vous utilisez déjà un fichier `robots.txt`…

Pour rappel, les balises de métadonnées se présentent ainsi :

```
<meta name="description" content="description personnalisée" />
```

Il arrive fréquemment que les spécialistes considèrent que remplir les métadonnées est inutile car elles n'affectent pas le classement des pages web. En réalité, ces propos peuvent être nuancés car il faut bien distinguer deux états de fait :

- les critères de positionnement indiquent des zones dans lesquelles les mots-clés sont valorisés et mieux « notés » par les moteurs ;
- les zones « froides », par opposition aux zones optimisées, ont aussi un rôle à jouer et permettent d'inclure également des mots-clés qui peuvent influencer le positionnement final.

Nous devons considérer les métadonnées comme des zones froides qui ont un rôle à la fois visuel et incitatif dans les SERP, mais aussi car elles constituent des contenus à part entière. Il est peu probable que les moteurs captent tous les textes et ignorent les métadonnées. Étant donné que nous savons qu'elles sont réutilisées ensuite dans les résultats de recherche, nous devons donc au moins les travailler avec la même attention que les contenus textuels.

Description et titre, même combat !

Il arrive que les descriptions soient entièrement personnalisées par les moteurs de recherche s'ils estiment qu'elles sont imparfaites ou qu'elles sont exagérément truffées de mots-clés *(keyword stuffing)*.

Les contenus textuels

Les titres de documents et les métadonnées font partie des facteurs historiques des optimisations internes, mais les contenus sont à ce jour ce qui constitue le point fort des sites qui réussissent à se distinguer sur la Toile et dans les SERP. Nous devons donc les optimiser avec intelligence pour obtenir de bons résultats.

Titres internes

Les titres internes sont générés à l'aide des balises `<h1>` à `<h6>` en HTML (les titres des balises `<h1>` étant plus importants et grands que ceux des balises `<h6>`). Les termes insérés entre ces balises ont plus de poids pour les moteurs, mais il convient de ne pas faire de bourrage de mots-clés ni de créer des titres interminables.

S'il n'existe aucune longueur conseillée, sachez qu'il faut rester mesuré et présenter des titres internes en adéquation avec les contenus qui suivent mais aussi avec le `<title>` des pages rédigées. Les moteurs de recherche savent interpréter les contenus et détecter s'il s'agit de spam ou de contenus de mauvaise qualité. Nous détaillerons ce point lorsque nous évoquerons Google Panda (voir Chapitre 3, section « Google Panda »).

Généralement, il est conseillé de n'avoir qu'un seul titre `<h1>`, un ou plusieurs `<h2>` et `<h3>`, etc. Le titre le plus important de la page est souvent le logo, c'est pourquoi on lui assigne souvent une balise `<h1>`. À noter qu'il est toléré dans ce cas d'avoir un second titre de premier niveau pour les intitulés d'articles ou

de pages. Au-delà, nous devons descendre d'un cran pour chaque titre de plus bas niveau, en respectant une certaine hiérarchie dans le code.

Gestion des titres dans les CMS

Les utilisateurs de backoffice ont souvent plus de facilité à gérer la hiérarchisation interne des pages car la grande majorité des thèmes graphiques imposent un `<h2>` dans les articles ou les pages, ce qui correspond aux titres rédigés par les auteurs dans les pages de création de contenus.

Rédiger et enrichir les codes

À l'instar des titres internes, il existe des méthodes pour mettre en avant certains contenus plus que d'autres. Cela passe par l'usage de balises HTML spécifiques qui mettent en exergue des termes clés afin que les moteurs sachent bien ce qui compte le plus à nos yeux.

Le couple principal est `<strong>…</strong>`, qui permet d'insister sur des mots précis. Sa représentation initiale est une mise en gras mais cela peut varier si le code CSS est modifié (possible pour cet élément, contrairement à `<b>` qui ne fait que mettre en gras). Il est primordial d'encadrer des expressions et des mots forts avec ce balisage pour les valoriser et optimiser certaines futures requêtes.

Pour certains moteurs de recherche, il arrive encore que les balises `<em>…</em>` (emphase en HTML) puissent jouer un petit rôle au même titre que `<strong>`, mais cela semble de moins en moins vrai. Ici, la représentation native est une mise en italique et peut donc s'avérer pratique dans certains cas.

Illustrations et multimédia

Les balises multimédia sont de plus en plus nombreuses depuis l'arrivée de l'HTML 5, mais toutes ne jouent pas encore de rôle majeur dans le positionnement des pages web. Nous savons toutefois que plusieurs éléments ont un impact sur la valorisation des contenus tant qu'ils ne sont pas suroptimisés.

- L'attribut `alt` des images permet d'indiquer un texte de remplacement qui sera accessible aux personnes malvoyantes mais aussi en cas de non-chargement des illustrations. Les textes et légendes contenus au sein des attributs `alt` ont plus de valeur pour les moteurs de recherche, nous devons donc les travailler avec précision.
- Les contenus représentés par les balises multimédia `<iframe>` et `<embed>` peuvent être accompagnés de balises `<noframe>` et `<noembed>` pour intégrer des contenus additionnels cachés ou de remplacement afin de qualifier les vidéos ou sons mis en place dans les pages web. Les autres balises telles que `<object>`, `<audio>` et `<video>` ne présentent pas ce type d'élément, il suffit d'intégrer les textes de remplacement à l'intérieur (entre les balises) pour obtenir le même résultat.

Quels que soient les contenus multimédia que vous mettez en avant, pensez toujours à ajouter ces textes de remplacement car leur rôle ne sera jamais négligeable pour le positionnement final des pages web. La majorité des outils de création de contenus comme WordPress ou Drupal mettent à disposition des champs pour spécifier ces contenus, seuls leurs intitulés varient d'un service à un autre…

Hypertextualité et ancres de liens

Les liens jouent aussi un grand rôle dans le classement des pages web. Certes, leur impact tient davantage des facteurs externes que des critères internes, mais il est indispensable de bien travailler les textes cliquables (ancres de liens) afin que les liens aient plus de poids pour la page visitée mais aussi pour la page ciblée par l'URL.

Au même titre que les balises `<strong>`, par exemple, les textes insérés entre les balises d'ancres `<a>...</a>` ont plus de poids pour les moteurs de recherche. Il est important de travailler la qualité des ancres de liens et de les faire varier pour des pages données afin d'éviter d'éventuelles sanctions. Nous reviendrons en détail sur ce point lorsque nous évoquerons Google Penguin (voir chapitre 3, section « Google Penguin »).

Retenez que les liens doivent avoir des textes cliquables optimisés et variés pour valoriser les pages web et améliorer le classement sur des requêtes données (qui correspondent ici aux ancres travaillées).

FreshRank, Freshness et mises à jour des contenus

Il est important de proposer des contenus mis à jour fréquemment pour valoriser les pages voire les sites web tout entiers. En effet, les moteurs de recherche considèrent les pages mises à jour comme plus pertinentes ; nous devons donc créer des zones mises à jour assez régulièrement pour améliorer le classement général du site.

Google a mis en place dès 2007 un algorithme intitulé *Query Deserved Freshness* (QDF), officialisé par l'ingénieur logiciel Amit Singhal le 3 juin 2007 (source : http://goo.gl/OST6GW). Ce premier algorithme du genre a permis au moteur de recherche de valoriser les pages web récentes ou d'actualité lorsque les requêtes imposaient de la fraîcheur de contenu. 17 % des requêtes étaient annoncées comme affectées dès 2007.

Si vous doutez encore de la pertinence de ce facteur, sachez que Google a également déposé un brevet le 18 mars 2008 intitulé *Information Retrieval Based on Historical Data* pour présenter la notion de FreshRank (source : http://goo.gl/bcHKs). L'objectif du brevet est de décrire en détail les facteurs liés à l'ancienneté et la mise à jour des pages web.

Dans la lignée du FreshRank, Google a déployé Freshness le 3 novembre 2011 (source : https://goo.gl/a9u3c4). 35 % des recherches ont été affectées par cette mise à jour favorisant les pages web aux contenus récents dans les SERP lorsque les recherches le nécessitent. Cela vise surtout deux types de requêtes :

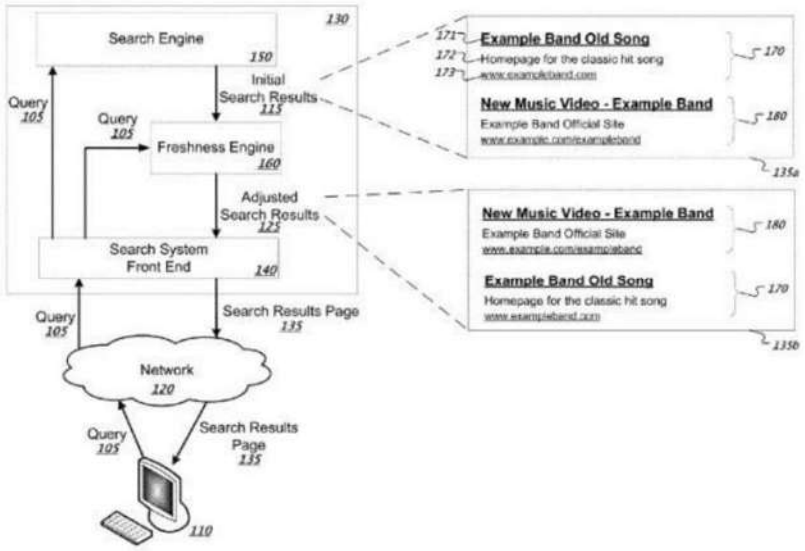
- les requêtes changeantes comme celles sur des avis, des notes, des commentaires...
- les requêtes événementielles, soit pour des dates récurrentes (Noël, soldes, élections présidentielles, 14 juillet...), soit pour des actualités chaudes ou des tendances (informations et modes du moment). Les requêtes contenant ces types de mots-clés sont comprises par le moteur de recherche pour restituer les pages les plus récentes et pertinentes sur le sujet.

Enfin, Google a obtenu un nouveau brevet du même type appelé *Freshness based ranking* le 17 novembre 2015 (source : http://goo.gl/OhGMBh). Ce dernier est dans le même esprit que les précédents et offre au moteur de recherche la possibilité de favoriser le positionnement des pages web en fonction

de la fraîcheur et des mises à jour de leurs contenus. Ce nouveau système ne se limite pas aux résultats naturels mais permet aussi à Google de favoriser le classement des ressources récentes sur Google Actualités, Images, Vidéos et même dans le moteur de recherche des blogs. Ainsi, de nombreuses requêtes et recherches des utilisateurs sont affectées en fonction de la proximité temporelle des contenus web.

Figure 2-3

Nouvel algorithme Freshness based ranking obtenu par Google



Sur Google, nous savons grâce aux algorithmes Freshness et FreshRank que les pages les plus fréquemment mises à jour gagnent des places dans les SERP pour une grande quantité des recherches. Sachez également que les pages anciennes, souvent commentées et visitées, vont aussi être mieux classées grâce à leur pertinence importante et à l'appui du comportement des internautes.

Gestion de l'écrit et de l'orthographe

Au sein des pages, nous devons toujours placer les contenus les plus importants le plus haut possible dans le code car les moteurs dévaluent peu à peu leur poids en fonction de leur placement dans les codes sources. Nous devons donc gérer les écrits et les zones à rédiger afin de ne pas tomber dans des pages peu valorisées.

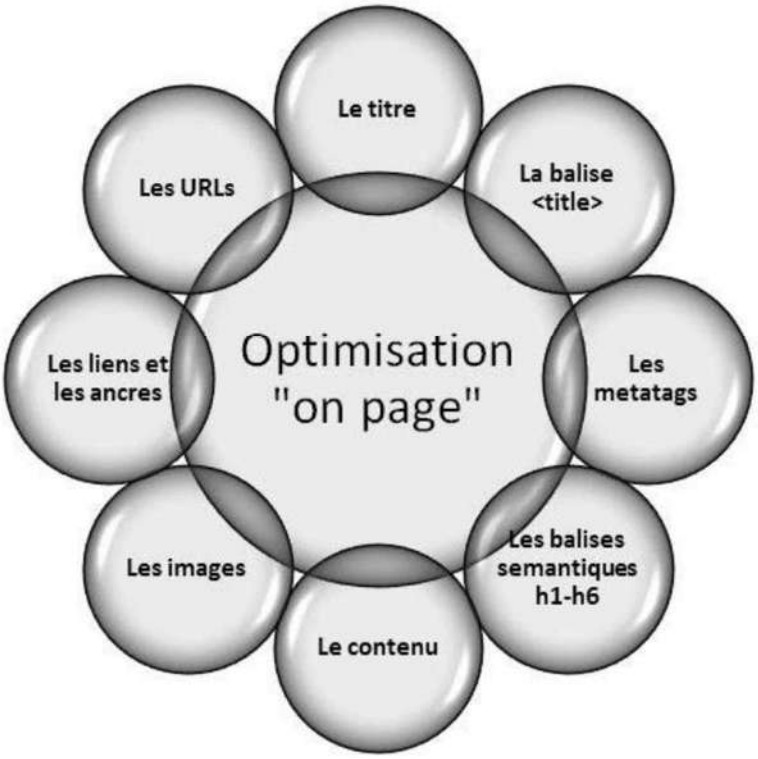
De plus, il est de plus en plus important de rédiger proprement les contenus car la recherche sémantique développée par les principaux outils de recherche tels que Google, Bing et Yandex risque petit à petit de modifier leur vision de l'écrit. En effet, il est fort probable que les moteurs arrivent à terme à lire, comprendre et noter la qualité rédactionnelle au sein des pages web pour mieux classer les documents dans les SERP.

Nous savons, par exemple, que la qualité de l'orthographe joue un rôle sur Bing depuis l'annonce officielle présentée fin février 2014 (source : http://goo.gl/OnzLTe). Nul doute que d'autres moteurs dont Google appliqueront ce type de pratiques à l'avenir. Cela peut d'ailleurs être imaginé avec le déploiement d'algorithmes comme Google Hummingbird dont l'objectif est de mieux comprendre les recherches conversationnelles…

Résumons l'ensemble des facteurs d'optimisation interne par un schéma explicatif. Si vous voulez en savoir davantage à ce sujet, nous ne pouvons que vous conseiller de vous tourner vers d'autres lectures comme l'excellent ouvrage *Réussir son référencement web* d'Olivier Andrieu, publié aux éditions Eyrolles.

Figure 2-4

Résumé des optimisations on page en SEO



Compatibilité mobile, App Indexing et AMP HTML

Les supports mobiles prennent de plus en plus d'importance dans la vie quotidienne, et les moteurs de recherche l'ont bien compris, notamment Google. La firme s'est donc penchée sur des facteurs clés relatifs à la mobilité afin de valoriser les sites web et applications mobiles qui font des efforts pour aider le moteur mais surtout les mobinautes.

Plusieurs nouveautés ont connu une forte valorisation en 2015 de la part de Google (certains systèmes existaient déjà) avec un boost SEO accordé en contrepartie de leur mise en place.

- Le 21 avril 2015 a été marqué par ce que les webmasters ont appelé le « mobilegeddon », à savoir une modification profonde de l'algorithme de pertinence de Google Mobile afin de mieux valoriser les sites web compatibles mobiles. Cela signifie que les pages web créées sur une architecture adaptative *(responsive web design)* ou les sites web mobiles à part entière peuvent obtenir un meilleur positionnement dans les SERP mobiles.

- L'App Indexing est une méthode d'indexation que nous avons déjà évoquée dans le chapitre précédent. Pour favoriser sa mise en place, Google a annoncé accorder un boost de ranking aux liens profonds d'applications mobiles affichés dans les SERP lorsqu'ils proviennent de l'App Indexing. L'objectif est clairement de pousser les développeurs d'applications à utiliser l'API de Google et surtout de les forcer à indexer un maximum de liens profonds.
- AMP HTML est une réécriture de l'HTML destinée à accélérer considérablement l'affichage des pages web sur les supports mobiles. Il s'agit de versions statiques (une sorte de cache) des pages web qui économise beaucoup de temps de chargement (et téléchargement) des ressources web souvent gourmandes en bande passante (images, vidéos, PDF…).

Utiliser AMP HTML a-t-il un avantage en SEO ?

AMP est un projet open source soutenu par de nombreuses sociétés comme Google, Twitter, Pinterest ou encore Parse.ly. La réécriture HTML faite par AMP va évoluer dans le temps et s'améliorer pour accélérer de plus en plus de processus dans les pages web mobiles.

Google n'a pas confirmé de valorisation de positionnement pour les utilisateurs de l'AMP HTML, mais cela améliore considérablement le PageSpeed et le confort de navigation sur mobile ; ce n'est donc pas anodin. Qui plus est, la firme a confirmé que les pages en AMP repérées par Googlebot-mobile pourraient remplacer les pages classiques dans les SERP mobiles dès 2016 (source : http://goo.gl/ytsUqw).

Nous reviendrons en détail sur ces différents éléments associés aux supports nomades dans ce chapitre afin que vous puissiez tirer profit au maximum de ces critères d'avenir...

Les optimisations off page

Netlinking

Le *netlinking* correspond à toutes les techniques qui permettent d'obtenir des liens entrants vers les pages web d'un site. Il s'agit d'un des facteurs les plus importants pour le positionnement depuis l'arrivée de Google en 1998 avec son indétrônable PageRank.

PageRank, TrustRank, BrowseRank…

Les facteurs de netlinking sont nombreux et chaque moteur compose ses propres algorithmes pour mesurer la qualité et le nombre de liens entrants *(backlinks)* obtenus par les pages web.

Google s'est fait connaître avec le PageRank en 1998 qui permet de qualifier les pages web en fonction du nombre de liens obtenus. Plus une page obtient de liens pointant vers elle, plus sa note sur dix est élevée. De nos jours, ce critère a été couplé à la notion de TrustRank qui détermine la qualité des liens. Nous reviendrons en détail sur ces facteurs importants dans ce chapitre.

Chez Bing, Le couple PageRank/TrustRank a laissé sa place au BrowseRank qui réalise le même type de calcul et d'analyse du profil des liens obtenus par les pages pour les faire remonter dans les SERP. Bing dispose aussi d'un algorithme appelé StaticRank, souvent oublié par les spécialistes, mais qui permet de qualifier les contenus des pages web (son rôle est indépendant du netlinking).

Chaque moteur développe ses propres méthodes pour maîtriser la qualité du netlinking des sites web afin de classer plus ou moins bien les pages dans les résultats de recherche. Ces critères restent à ce jour essentiels pour réussir son positionnement web.

Aujourd'hui, l'abondance de liens est la cible de filtres et d'algorithmes suite aux abus commis par certains référenceurs peu délicats à ce sujet. La mise en place de systèmes de vérification de la qualité de netlinking impose que des règles soient respectées pour ne pas voir les pages chuter drastiquement dans les SERP. Nous reviendrons sur ce sujet plus loin.

Diverses sources de netlinking

L'obtention de liens peut se faire de multiples manières mais l'important est d'avoir toujours un profil qui semble « naturel » et qui permet aux moteurs de recherche de ne pas déceler un trop-plein d'optimisation du netlinking.

Voici une liste de sources qui peut vous permettre d'obtenir des liens assez facilement, mais gardez bien en tête que l'usage abusif de certaines sources risquent de se solder par de lourdes sanctions :

- les annuaires de recherche ;
- les communiqués de presse ou « CP » (attention aux faux communiqués qui sont chassés par les moteurs) ;
- les articles invités (ou *guest blogging*) qui permettent en général d'obtenir de bons liens sur des sites qualifiés ;
- les flux de syndication (RSS ou Atom) ;
- les services d'agrégation de contenus ;
- les réseaux sociaux ;
- les commentaires et avis de consommateurs autorisés dans certains sites web (e-commerce, forums, blogs...) ;
- les fichiers PDF contenant des liens référencés dans les SERP.

Vitesse des pages web et serveurs

La vitesse de chargement des pages et des serveurs est un critère assez récent qui a été mis en place par Google (et Yahoo! de son côté). L'objectif est de qualifier les pages web en fonction de divers critères destinés à accélérer leur chargement. Contrairement aux idées souvent avancées par les spécialistes, les moteurs ne notent pas uniquement les sites par rapport à la vitesse pure de chargement des pages, mais bien selon une liste de critères précis à optimiser.

L'essentiel de ce chapitre sera consacré à l'optimisation de ces nombreux facteurs techniques relatifs aux serveurs Apache et IIS (Microsoft) afin que vous puissiez maîtriser les tenants et aboutissants de ces optimisations (indiquées dans le PageSpeed de Google notamment). Il s'agit très certainement de l'un des facteurs les plus difficiles à optimiser mais son rôle n'est pas à négliger dans le positionnement final.

En revanche, rien ne dit que d'autres moteurs que Google utilisent encore à ce jour de telles analyses, bien que nous sachions que Yahoo! a utilisé YSlow et que Bing s'intéresse de près à la qualité des codes sources et à la vitesse des pages et des serveurs. Le mieux est de considérer que cette liste de critères relatifs à la notion de PageSpeed sera à travailler avec précision à l'avenir.

AuthorShip et AuthorRank

Nous allons consacrer une grande partie de ce chapitre à traiter des phénomènes récents que constituent l'AuthorShip et l'AuthorRank.

Initialement, ces facteurs de Google sont destinés à composer un arbre relationnel concernant l'ensemble des contenus publiés sur le Web par un même internaute afin de le noter selon son impact sur la Toile. Désormais, ces facteurs ont changé de visage sur Google et ont été étendus à Bing voire Yahoo!. De récentes informations ont permis de mieux comprendre comment les auteurs sont notés par ces moteurs de recherche, nous expliquerons en détail ce qu'il en ressort.

Sécurité et sites web en HTTPS

Google a annoncé le 6 août 2014 la prise en compte du protocole HTTPS comme nouveau critère de positionnement (source : http://goo.gl/xZYftr). Depuis plusieurs années, de nombreux moteurs de recherche basculent leurs outils en HTTPS afin de proposer une navigation plus sécurisée aux utilisateurs. Ce nouveau facteur s'est donc logiquement imposé aux yeux de Google.

Un facteur évolutif ?

Les différents porte-parole de Google qui ont évoqué la sécurisation des sites web ont indiqué qu'il s'agit à ce jour d'un critère de faible poids mais que cela pourrait évoluer rapidement à l'avenir. Durant la conférence « Search marketing expo east » aux États-Unis, Gary Illyes de Google Zurich a même indiqué que seulement 1 % des requêtes saisies par les utilisateurs profitent actuellement de ce facteur, avant de préciser que les ingénieurs réfléchissent à le faire évoluer à l'avenir (source : http://goo.gl/tnlFXe).

Désormais, la firme a décidé de mettre un peu plus en avant les pages web qui feraient l'effort de proposer le protocole sécurisé HTTPS. À ce jour, il semblerait que l'impact soit encore très faible comme l'a reconnu John Mueller, analyste des tendances de Google, le 11 août 2014 (source : http://goo.gl/uNP5UZ). Ce facteur impacte bien chaque page de façon indépendante et risque de prendre plus de poids dans l'algorithme dans les mois à venir, forçant ainsi nombre de webmasters à faire la bascule vers HTTPS.

Le 14 septembre 2015, Gary Illyes, Trends Analyst à Google Dublin, a précisé dans un tchat vidéo avec Bruce Clay (SEO américain de renom) que le critère de l'HTTPS pouvait faire la différence entre deux URL lorsqu'elles ont un poids similaire ou presque (source : http://goo.gl/kzwYH6). Par conséquent, le boost SEO accordé aux pages web en HTTPS relève davantage d'un outil pour départager dans le classement des SERP que d'un critère à part entière en 2015.

Figure 2-5

L'HTTPS comme critère de ranking depuis le 11 août 2014

We want to go even further. At Google I/O a few months ago, we called for "HTTPS everywhere" on the web.

We've also seen more and more webmasters adopting HTTPS (also known as HTTP over TLS, or Transport Layer Security), on their website, which is encouraging.

For these reasons, over the past few months we've been running tests taking into account whether sites use secure, encrypted connections as a signal in our search ranking algorithms. We've seen positive results, so we're starting to use HTTPS as a ranking signal. For now it's only a very lightweight signal — affecting fewer than 1% of global queries, and carrying less weight than other signals such as high-quality content — while we give webmasters time to switch to HTTPS. But over time, we may decide to strengthen it, because we'd like to encourage all website owners to switch from HTTP to HTTPS to keep everyone safe on the web.

Rien ne dit que son évolution ne va pas permettre d'aller plus loin à terme, mais cela ne semble pas être le chemin que le critère prend, d'autant plus que Gary Illyes a déclaré ceci : « Je ne peux pas attendre que tout le monde migre vers HTTPS. Certaines personnes n'ont pas les ressources pour cela, d'autres ne veulent pas le faire pour d'autres raisons. C'est important de manière générale, mais si vous ne le faites pas, ce n'est pas grave. »

Dans les faits, le passage d'un site web de HTTP à HTTPS peut s'avérer quelque peu fastidieux tout comme le choix des certificats SSL/TLS (fichiers de validation nécessaires pour le transfert des données chiffrées) auprès des autorités de certification agréées telles que Verisign, Thawte, GlobalSign, GeoTrust ou encore Comodo et TBS Internet… Renseignez-vous bien auprès de votre hébergeur pour effectuer cette mutation sans accroc, mais sachez aussi que certains certificats peuvent coûter de plusieurs dizaines à plusieurs centaines d'euros.

Toutefois, ce frein financier pourrait disparaître car une majorité de certificats SSL/TLS vont devenir gratuits grâce au projet « Let's Encrypt » propulsé par l'EFF *(Electronic frontier foundation).* Il s'agit d'une nouvelle autorité de certification financée par des marques comme Cisco, Mozilla, Akamai ou encore IdenTrust. Plusieurs hébergeurs se sont déjà joints au projet pour proposer des certificats gratuits à leurs utilisateurs, cela devrait donc se propager en 2016 et par la suite…

Des webmasters hésitants à migrer vers HTTPS ?

Fin 2015, encore assez peu de webmasters osent franchir le pas d'une migration vers HTTPS pour des raisons de coûts, de lourdeur technique (la migration n'est pas toujours simple à mettre en œuvre, notamment sur des sites importants), de poids du critère (plutôt faible à ce jour), voire pour d'autres causes comme la lenteur annoncée des pages sécurisées (dans les faits, HTTPS ne ralentit pas vraiment le chargement des pages, contrairement aux idées préconçues).

Le 28 août 2015, John Mueller est venu ajouter un doute supplémentaire pour les webmasters réticents. En effet, il a annoncé dans une vidéo que le boost SEO relatif au passage vers HTTPS ne serait accordé qu'aux pages web ne contenant des connexions qu'avec des ressources en HTTPS (source : http://goo.gl/yXDqQU). Le porte-parole de Google a précisé que les pages qui ne proposeraient pas 100 % de ressources sécurisées ne seraient pas considérées comme des pages HTTPS et, donc, ne bénéficieraient pas du bonus de positionnement normalement accordé pour ce protocole. Étant donné que de nombreuses ressources externes comme des images, des vidéos, des scripts… sont encore diffusées via HTTP, autant dire que cela ne donne pas vraiment envie aux webmasters d'effectuer une telle migration au risque de ne même pas obtenir un boost SEO. Toutefois, d'autres annonces n'ont pas confirmé les dires du porte-parole de Google en 2016.

La migration d'un site de HTTP vers HTTPS doit se faire en plusieurs étapes clés afin d'éviter tout risque.

- Corriger toutes les URL internes du site pour les passer en HTTPS. Idéalement, il convient d'ajouter plutôt des URL relatives (seulement le chemin vers la ressource web) ou des URL relatives de protocole (de la forme "//www.site.extension") afin de laisser le navigateur s'adapter automatiquement au bon protocole.
- Effectuer des redirections permanentes (301) des anciennes URL vers les nouvelles pages web en HTTPS.
- Vérifier que les fichiers `.htaccess` (ou `web.config`), `robots.txt` et les autres techniques de désindexation ne viennent pas interférer avec la migration vers HTTPS. Il arrive en effet que d'anciennes directives bloquent la bonne indexation des pages en HTTPS par mégarde. Il est recommandé de bien vérifier que rien ne puisse empêcher le crawl des fichiers après migration.
- Vérifier la validité des certificats SSL pour ne pas tomber dans l'expiration et des messages d'erreurs intempestifs pour les visiteurs. Il faut penser à renouveler les certificats fréquemment.

Une autre méthode de redirection conseillée vise à utiliser si possible HTTP Strict Transport Security (HSTS). Cette technique redirige automatiquement et de manière sécurisée les ressources en HTTP vers leurs équivalents en HTTPS, tout cela côté serveur. De ce fait, même un robot d'indexation est renvoyé vers la version sécurisée du site web. Voici une méthode en PHP pour appliquer HSTS :

```
<?php
// Vérification en PHP de la présence d'HTTPS ou non
if (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] != 'off') {
        header('Strict-Transport-Security: max-age=31536000');
} else {
        header("Status: 301 Moved Permanently", false, 301);
        header('Location: https://'.$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI']);
        exit();
}
?>
```

```
# Variante PHP avec l'envoi direct d'un en-tête HSTS (déconseillé)
header("Strict-Transport-Security:max-age=31536000");
?>
```

Et voici une méthode équivalente en VB Script pour les technologies Microsoft (ASP) :

```
<%
' Vérification de la présence d'HTTPS en ASP (VBS)
If Request.Url.Scheme = "https" Then
        Response.AddHeader "Strict-Transport-Security","max-age=500"
ElseIf Request.Url.Scheme = "http" Then
        Response.Status="301 Moved Permanently"
        Response.AddHeader "Location", "https://" + Request.Url.Host
                                                  + Request.Url.PathAndQuery
End If
%>
```

Compatibilité de l'HSTS

HSTS n'est pas compatible avec l'ensemble des navigateurs. C'est pourquoi cette solution est souvent une technique de secours. Actuellement, Internet Explorer 11 et Microsoft Edge prennent en compte HSTS, à l'instar de Google Chrome et Chromium, Opera (depuis la version 12), Firefox (depuis la version 4) et Safari (depuis la version 7.0 présente sur Mac OS Mavericks 10.9). Les anciens navigateurs ne permettent donc pas de profiter pleinement de cette fonctionnalité, Safari et Internet Explorer notamment.

Cela va sans dire, mais il faut également vérifier les liens des extensions et modules présents dans les sites web, ainsi que les URL pointant vers les autres ressources (CSS, JavaScript…) afin que l'ensemble du site soit migré entièrement et sans problème.

Les problèmes causés par le passage à HTTPS

De nombreux soucis ont été décelés lors du passage des sites en HTTPS. Il faut savoir que seules les données sont chiffrées avec SSL/TLS lors des transferts d'informations, ce qui signifie que le reste de la sécurité des sites incombe encore aux webmasters, notamment au sein des bases de données.

Ce nouveau facteur de positionnement a causé quelques peines, même au sein de Google, notamment auprès des membres de l'équipe de Google Adsense (la régie publicitaire de la firme) car le passage de HTTP à HTTPS peut engendrer des baisses de gain pour les clients (du moins dans un premier temps) en plus de quelques problèmes de compatibilité avec les programmes de la régie.

Cette modification implique que tous les liens internes des pages doivent être vérifiés et modifiés pour fonctionner avec le nouveau protocole. Il faut donc être vigilant lors du portage du site vers HTTPS en faisant les bonnes redirections, en nettoyant les liens hypertextes et en vérifiant que des fichiers `robots.txt` ne viennent pas fausser le portage à cause d'une règle bloquante…

Fin 2015, plusieurs sources officielles de Google ont indiqué que les pages en HTTPS allaient être indexées en priorité sur les pages HTTP classiques quand les deux versions existent (source : http://goo.gl/CqoMRV). Le but est de valoriser davantage les versions sécurisées aux utilisateurs dans les résultats de recherche.

Cela n'a pas demandé beaucoup de temps puisque la société Moz a publié une étude qui démontre que 25 % des résultats de la première page sont des URL en HTTPS (source : http://goo.gl/Y0kQ2a). De fait, nous pouvons imaginer que cela est dû à ce phénomène voire aussi au petit bonus accordé par Google pour les pages sécurisées.

Autres critères externes

La liste des critères externes ne se limite pas à l'optimisation du netlinking, du PageSpeed ou encore de l'AuthorRank. Il existe aussi quelques critères sur lesquels nous ne pouvons pas ou peu intervenir. En effet, de nombreux facteurs sont indépendants de notre travail, bien qu'ils en soient souvent la conséquence, mais ont pourtant leur rôle dans le classement final…

Ancienneté du nom de domaine et des liens

Les moteurs de recherche, Google en tête, prennent en compte l'ancienneté des pages, du nom de domaine mais aussi des backlinks afin de mesurer la pertinence des pages.

Par conséquent, nous pouvons considérer que plusieurs facteurs relatifs à l'historique des pages ont un impact sur le classement final :

- plus un nom de domaine est ancien, plus il prend de valeur aux yeux des moteurs de recherche ;
- plus une page est ancienne, plus elle est considérée comme pertinente ;
- plus les backlinks sont anciens, plus ils ont de valeur pour les pages ciblées.

Ces facteurs sont indépendants de nos efforts mais dévoilent tout de même quelques informations. Il est par exemple préférable de réserver et faire vivre un domaine le plus tôt possible. Il est aussi conseillé d'obtenir rapidement quelques backlinks de qualité pour qu'ils soient de plus en plus valorisés avec le temps. Certes, nos contenus vont vieillir naturellement mais nous pouvons anticiper ce phénomène pour optimiser la gestion des âges à notre manière.

Statistiques comportementales

Les moteurs de recherche comme Bing considèrent que les facteurs comportementaux sont à prendre en compte pour valoriser les pages web. Il est vrai que les internautes restent les meilleurs acteurs pour préciser la qualité des pages et leur intérêt sur le Web.

Dans le BrowseRank et le StaticRank de Bing, les statistiques comportementales influent sur le classement définitif des pages. Ce phénomène a aussi été ajouté chez Google qui mentionne souvent le rôle du comportement des usagers au sein des pages. Nous savons que les facteurs suivants peuvent influer sur le positionnement :

- le nombre de visiteurs uniques reçus par les pages ;
- le nombre moyen de visites par page ;
- le nombre de pages vues par session ;
- le taux de rebond.

Dans les faits, il semblerait que les statistiques comportementales n'aient pas une influence majeure dans le classement des SERP mais cela reste à prouver et nous ne pouvons pas savoir si cela changera à terme.

Notre rôle est de proposer des contenus à forte plus-value afin que les internautes y trouvent leur compte et réagissent bien au point que les moteurs de recherche valorisent nos pages web. L'influence de notre travail est donc directement mise en cause dans l'obtention d'un bon classement de pages web.

Nombre de pages des sites web

Un dernier critère compte énormément pour Google et Bing, il s'agit du nombre de pages qui composent un même site. Plus un site contient de pages, plus il est considéré comme important et pertinent. Bien entendu, la qualité de ces pages n'est pas à négliger, les moteurs ne cherchent pas à faire du nombre mais bien à noter favorablement les sites web qui proposent de nombreux contenus.

Ce facteur n'est pas le plus important dans la hiérarchie des notes attribuées aux pages web car tous les sites web ne peuvent pas bénéficier des mêmes avantages à ce sujet. En effet, si ce critère était primordial, il ne ferait aucun doute que les sites statiques ou de présentation seraient toujours placés tout en bas des SERP.

De facto, nous remarquons que ce n'est pas le cas mais que des sites comme des boutiques en ligne ou des blogs sont avantagés car ils proposent sans cesse de nouvelles pages et de nouveaux contenus pour les internautes. Nous devons donc retenir qu'il peut être intéressant de proposer ce type de site web parallèlement à un site vitrine si notre positionnement ne décolle pas, même s'il ne s'agit pas de l'unique solution pour sauver la mise…

Gérer les fichiers .htaccess et les serveurs Apache

Les fichiers `.htaccess` représentent des listes d'options de configuration relatives aux serveurs Apache (les plus courants). Ce sont les premiers fichiers lus lors d'une visite d'un site web, que ce soit par un internaute ou un robot, avant même le fichier `robots.txt`.

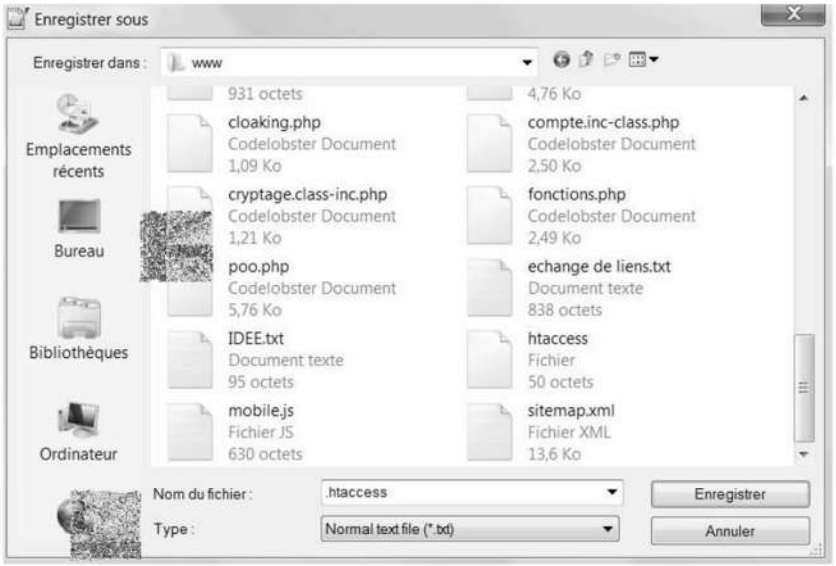
Ces fichiers sont très importants dans la gestion des sites web tant ils ont la capacité de modifier le comportement général des serveurs Apache. Le seul inconvénient est que certaines instructions ne fonctionnent que si nous sommes en possession d'un serveur dédié (et non d'un hébergement mutualisé).

Nous allons étudier plusieurs techniques importantes relatives aux fichiers `.htaccess`, fonctionnelles uniquement sur les serveurs Apache. Nous tenterons par la suite d'établir quelques parallèles avec les serveurs IIS de Microsoft afin que tout le monde puisse optimiser au mieux son référencement.

Les fichiers `.htaccess` doivent être placés à la racine des dossiers concernés par les directives. Par exemple, si vous voulez forcer l'encodage en UTF-8 dans un répertoire et mettre de l'ISO-8859-1 dans un autre, vous aurez un fichier `.htaccess` par dossier. En d'autres termes, un site peut contenir un nombre important de fichiers `.htaccess` en fonction de son architecture initiale. Il convient donc de bien réfléchir à cet aspect dès le début, notamment si vous devez à terme réécrire des URL.

Figure 2-6

Création d'un fichier .htaccess avec Notepad++



Sur Windows, il est impossible de créer directement des fichiers de la forme « .quelque-chose » car le système d'exploitation a absolument besoin d'un nom de fichier avant le point et l'extension. Il existe cependant quelques astuces pour créer ces fichiers sans être trop embêté par l'ami Windows.

- Ouvrez un éditeur de texte, quel qu'il soit (Notepad++, WordPad, Word, Bloc-notes…). Sélectionnez Fichier>Enregistrer sous…, puis saisissez `.htaccess` en lieu et place du nom de fichier. Confirmez l'enregistrement et le tour est joué.
- Créez un fichier texte intitulé `htaccess.txt`, transférez-le sur votre serveur distant avec un client FTP (FileZilla, Cyberduck, GoFTP, FireFTP…), puis renommez le fichier `.htaccess`. Il vous suffit de le télécharger pour l'obtenir sur Windows avec le nom correct.
- La dernière technique consiste à passer en ligne de commande avec Windows à l'aide de l'éditeur DOS. Exécutez `cmd.exe`, puis renommez le fichier texte créé en `.htaccess` grâce à l'instruction `rename htaccess.txt .htaccess`.

Les serveurs Apache fonctionnent avec un système de modèle à implémenter, les fichiers `.htaccess` sont plutôt des fichiers généraux de configuration et il est préférable de bien savoir si les modules intéressants sont activés et installés sur le serveur, au risque d'écrire des lignes de code dans le vide.

Figure 2-7

Création d'un fichier .htaccess sur Windows en ligne de commande

Renommer les fichiers .htaccess

Sachez que le fichier ne doit pas obligatoirement s'intituler `.htaccess`. Si vous êtes passionné(e) de manipulation en tous genres, retenez qu'il est possible de modifier le nom des fichiers en utilisant la directive `AccessfileName` des serveurs Apache avec une simple ligne de code : `AccessFileName .fichierconfig`

Gardez en mémoire que les directives sont appliquées dans le sens de lecture du serveur. En d'autres termes, un fichier `.htaccess` placé à la racine s'applique de manière récursive sur les sous-répertoires, sauf en cas d'écrasement des informations. Si nous reprenons l'exemple des jeux de caractères : si le fichier placé à la racine configure l'UTF-8 mais qu'un autre dossier configure l'ISO-8859-1, alors la directive sera remplacée uniquement dans ce répertoire.

Enfin, sachez que les fichiers `.htaccess` répondent à des modules, à des directives mais aussi à des options (si elles sont activées à l'aide de `AllowOverride Options`, ce qui est souvent le cas par défaut). Nous verrons donc parfois des codes complexes pour les plus débutants d'entre vous, ne soyez pas surpris(e) ni frustré(e) de ne pas tout comprendre. L'objectif de ces codes est de s'appliquer pour la plupart par un simple copier-coller…

PageSpeed et vitesse de chargement des pages

Le PageSpeed a été créé en 2009 et correspond à une des dernières inventions phares de Google en matière de référencement, avant l'arrivée de l'AuthorRank et autres Google Panda, Penguin et RankBrain dans la hiérarchie des grands changements. Il s'agit d'une note calculée sur 100 qui comporte de nombreux facteurs d'optimisation des pages afin d'accélérer leur chargement et leur vitesse d'accès sur le serveur.

Matt Cutts, responsable de l'équipe Google Webspam, a annoncé début 2010 l'impact du PageSpeed en matière de positionnement web (source : http://goo.gl/U2jCol). Il s'agit d'un critère intéressant et surtout du meilleur moyen pour forcer les webmasters à agir. Cette déclaration a été officialisée plusieurs fois depuis, donc nous pouvons légitimement penser que c'est encore le cas actuellement, bien que certains détracteurs annoncent le contraire.

Le 1er décembre 2015, plusieurs questions ont été posées à ce sujet à des porte-paroles de Google comme John Mueller et Zineb Ait Mahajji. Leurs réponses ont apporté de bonnes précisions sur l'impact de la vitesse dans les algorithmes.

Zineb Ait Mahajji a répondu que les optimisations de la vitesse de chargement correspondaient à un critère de ranking du moteur (source : https://goo.gl/2CFnzv), tandis que John Mueller a apporté encore davantage de précisions en indiquant que Google prenait en compte deux facteurs de vitesse :

- vitesse pure du serveur : temps de téléchargement d'une page ou ressource web pour Googlebot. Dans ce cas, la distance et la connexion entre Googlebot (datacenters) et le serveur affecte ce facteur de vitesse ;
- PageSpeed traditionnel : temps de chargement d'une page dans le navigateur. Le temps de connexion joue un rôle infime ici, au détriment d'optimisations générales de la vitesse. Ce critère a un rôle (faible) dans le classement des pages web.

En réalité, c'est souvent l'amalgame entre la vitesse de chargement des pages au sens strict et les optimisations du PageSpeed qui sème le doute sur le rôle de ce critère en termes de positionnement. En effet, la vitesse de chargement est importante pour l'expérience utilisateur et pour accélérer la lecture des robots des moteurs de recherche, mais ne peut pas être un facteur à part entière de positionnement. Trop de paramètres rentrent en ligne de compte (heures de connexion, temps de latence du serveur, ralentissement de la connexion…) et il serait quasi impossible de calculer une « note » équitable et pérenne sur ce principe.

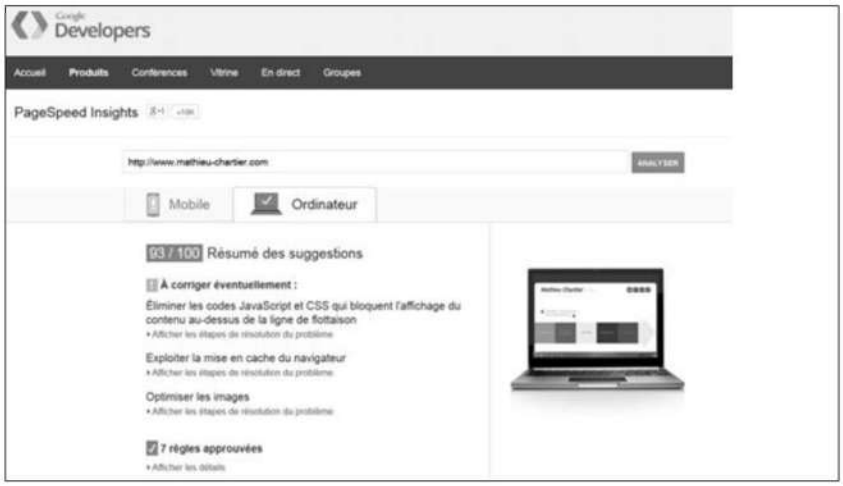
Figure 2-8

Indications de John Mueller sur le critère de la vitesse en SEO

En revanche, la note avancée par le PageSpeed se base sur une batterie de critères et celle-ci peut effectivement affecter le classement avec plus d'équité, à la fois parce que les critères optimisés ont un impact sur les sites, mais aussi parce que cette note est bien plus équilibrée et mesurable par les moteurs de recherche.

Figure 2-9

Analyse du site www.mathieu-chartier.com avec l'outil PageSpeed Insights



Les indications de John Mueller démontrent que Google analyse la vitesse en deux points précis : le crawl via une meilleure vitesse de téléchargement des ressources web, les optimisations des pages favorisant une meilleure vitesse de chargement. Ces deux facteurs sont appuyés en partie par un brevet intitulé *Using resources load times in ranking search result* publié en novembre 2010 (source : http://goo.gl/B3Y1px), puis les optimisations du PageSpeed ont certainement permis au moteur d'aller plus loin dans les détails.

Pour calculer la note d'un site web, vous pouvez utiliser l'outil PageSpeed Insights (source : http://goo.gl/rkUZUt). Tous les critères qui méritent des améliorations sont affichés avec une aide en ligne afin de se faciliter la tâche, bien que les facteurs les plus techniques ne soient pas expliqués en détail…

Dans les faits, il est rarement possible d'atteindre 100 % d'optimisation dans PageSpeed Insights, sauf s'il s'agit d'un petit site de présentation de quelques pages hébergées sur un serveur dédié. Rien que le fait de passer par des hébergements mutualisés peut faire baisser la note, car les serveurs sont ralentis dans ce cas et l'outil d'analyse le ressent avant d'attribuer la note finale.

Liste des facteurs d'optimisation

Google présente dans son aide aux webmasters une quinzaine de groupes d'options à optimiser (source : http://goo.gl/4mig3p). 31 critères étaient mis en avant mais l'outil évolue fréquemment. Des facteurs ont été regroupés et d'autres ont été ajoutés, notamment dans l'optimisation des serveurs et au sujet des sites mobiles. Voici désormais le découpage des règles à optimiser.

Règles relatives à la vitesse

- **Éviter les redirections vers la page de destination :** ce facteur se focalise sur les redirections générées en cascade pour diriger les internautes vers une version mobile d'un site notamment. Dès qu'une page s'ouvre, un chargement et des requêtes HTTP s'effectuent, il faut donc éviter de charger trop de pages intermédiaires pour rediriger vers un site mobile. Nous verrons comment éviter ce type de problème par la suite…
- **Autoriser la compression :** l'objectif est de compresser au format GZIP les données envoyées par le serveur directement lors du chargement des pages afin de réduire le poids des informations et donc d'accélérer le processus général. Il faut se référer au module `mod_deflate` des serveurs Apache, mais il existe également des alternatives pour IIS de Microsoft.
- **Améliorer le temps de réponse du serveur :** Google préconise un temps de réponse inférieur à 200 ms. Dans les faits, les possesseurs d'hébergements mutualisés sont souvent touchés par ce facteur malgré de bonnes optimisations du PageSpeed. Il s'agit d'un des derniers critères à avoir été mis en place et il a clairement permis d'abaisser les notes d'une multitude de sites web. Pour réduire le temps de réponse du serveur, il faut limiter le nombre de requêtes SQL autant que possible, éviter de multiplier l'usage de bibliothèques (jQuery, Prototype…), frameworks, CMS (...) pour limiter les temps de chargement lourds (certes, ces outils sont pratiques mais souvent très consommateurs d'énergie…).
- **Exploiter la mise en cache du navigateur :** Google souhaite qu'un maximum de ressources statiques soient mises en cache afin de réduire considérablement les temps de chargement des pages web. Ce facteur est très important et peut vraiment avoir un impact sur la vitesse d'affichage des pages. Nous reviendrons donc dessus en détail car plusieurs critères sont à optimiser.

- **Réduire la taille des ressources :** l'objectif est de réduire au maximum le poids des fichiers CSS, HTML (ou PHP, ASP...) et JavaScript. Dans les faits, cela consiste à supprimer tous les espaces dans les codes sources, tous les sauts de ligne ainsi que tous les points-virgules parfois inutiles en CSS. De nombreux outils peuvent nous aider pour faire des compressions de bonne facture, nous détaillerons cela plus loin.
- **Optimiser les images :** Google souhaite à tout prix que deux critères soient respectés en ce qui concerne les images présentes dans les sites web. D'une part, les images doivent être découpées à la taille exacte de celle prévue dans les pages web et d'autre part, il faut les compresser au maximum à l'aide de logiciels puissants sans pour autant perdre en qualité de rendu. Pour commencer, si vous enregistrez vos images avec la fonction Enregistrer pour le Web... de votre logiciel de traitement d'images, c'est un bon début. Mais souvent cela ne sera pas suffisant. Nous vous présenterons par la suite quelques logiciels utiles pour cet aspect.
- **Optimiser la diffusion des ressources CSS :** ce critère a pour but d'éviter la multiplication de fichiers CSS, mais aussi d'optimiser leur chargement. Cela passe par le regroupement des feuilles de styles en un nombre minimaliste de fichiers (un seul dans l'idéal), par la suppression des fonctions CSS `@import` qui ralentissent le chargement des pages web et/ou par le déplacement du lien `<link rel="stylesheet" href="style.css" />` en bas de code source (après la fermeture de `</html>`, bien que cela soit impropre pour le W3C).
- **Afficher en priorité le contenu visible :** ce facteur est en quelque sorte un condensé des autres facteurs puisque son rôle est d'augmenter le temps de chargement des informations situées au-dessus de la ligne de flottaison. Pour ce faire, il faut réduire la taille des ressources et des images, charger les scripts utiles au début et les autres plus tard. En d'autres termes, il faut placer les contenus importants le plus haut possible et optimiser les autres facteurs pour répondre favorablement à celui-ci...
- **Supprimer les scripts JavaScript qui bloquent l'affichage des contenus :** ce facteur se rapproche de celui sur la diffusion des ressources CSS puisque l'objectif est de placer en fin de code source les ressources JavaScript. L'idéal est de charger ces codes de manière asynchrone, de limiter le nombre de fichiers .js, d'éviter d'utiliser des scripts hébergés sur des serveurs distants (Google est expert pour nous proposer ses scripts pourtant...), et de les placer avant la fermeture du code HTML. Attention, il arrive que certains scripts nécessaires méritent de rester dans la section `<head>`, c'est notamment le cas des *sliders* (ou *slideshows*, carrousels...) car l'affichage des images provoquera un drôle d'effet avant que le script ne soit exécuté sinon...
- **Utiliser des scripts asynchrones :** l'usage de scripts JavaScript (voire jQuery, Prototype...) est courant mais nous chargeons trop souvent ces scripts de manière synchrone, c'est-à-dire qu'ils s'appliquent au moment de la lecture en quelque sorte. L'idéal est d'utiliser des méthodes pour rendre les scripts asynchrones afin qu'ils se lancent une fois que les éléments essentiels des pages web sont chargés. Google liste dans sa documentation nombre de scripts souvent utilisés avec des informations pour les charger de manière asynchrone. Nous verrons comment indiquer aux scripts de se charger de manière asynchrone en HTML.

Règles relatives à l'ergonomie

- **Éviter les plug-ins :** Google fait clairement comprendre ici que l'usage de plug-ins Flash, Java ou Silverlight ne fait pas bon ménage avec la vitesse de chargement. La firme soutient depuis le départ le langage HTML 5 et son développement, il faut dire que les temps de chargement et les plantages sont

quasiment réduits à néant avec cette technologie, contrairement aux plug-ins évoqués. Cette mention nous indique indirectement qu'il est préférable d'éviter les langages ActionScript (Flash) voire Java sous certaines formes (applets Java).

- **Configurer la fenêtre d'affichage :** il s'agit d'un critère relatif à l'adaptation sur des supports mobiles, notamment en *responsive web design*. Google préconise l'utilisation de la balise meta `viewport` sous la forme `<meta content="width=device-width, initial-scale=1" name="viewport" />` en évitant les valeurs `maximum-scale` (zoom maximal), `minimum-scale` (zoom minimal) ou encore `user-scalable`, bien qu'elles soient pratiques dans certains cas. Qui plus est, sachez qu'il est possible d'utiliser la fonction CSS `@viewport`, souvent méconnue mais pas inutile…
- **Adapter la taille du contenu à la fenêtre d'affichage :** ce facteur est dans la lignée du précédent et met en avant une nouvelle fois le responsive web design. L'objectif est de réaliser des mises en page 100 % relatives avec des unités en %, em, ex ou deg. L'unité px est tolérée dans certains cas mais doit vraiment être utilisée avec prudence. Enfin, il convient de rendre les médias adaptatifs avec un code CSS pouvant ressembler à celui-ci :

```
img, object, embed, iframe, video, audio {
width:100%;
max-width:100%;
height:auto;
}
```

- **Dimensionner les éléments tactiles de manière appropriée :** les principes ergonomiques et Google conseillent d'opter pour des boutons de tailles suffisantes sur les supports mobiles afin que la navigation ne soit pas dégradée. Cela se résume à utiliser des icônes et des boutons dont la taille minimale serait 32 × 32 pixels voire 48 × 48 pixels avec des marges pour que les utilisateurs n'appuient pas malencontreusement sur les mauvais boutons.
- **Utiliser des tailles de police lisibles :** en responsive web design ainsi qu'en ergonomie web, il est recommandé d'utiliser des polices lisibles et surtout des tailles d'écriture suffisantes pour ne pas gêner la lecture. La taille par défaut des navigateurs est souvent 16 px, sachant qu'un pixel représente 0,75 point. Vous pouvez ainsi facilement faire des conversions entre les tailles en points, souvent utilisés en rédaction, et les tailles finales en pixels (par exemple, 16 px est équivalent à 12 pt avec le calcul élémentaire 16 × 0,75). En revanche, nous utiliserons plus souvent les em (cadratin) en design adaptatif. Il faut alors calculer les tailles différemment à l'aide d'une formule simple : taille initiale / taille du contexte = taille finale. Par exemple, si vous souhaitez passer de 12 px à une taille relative en em, il faut alors effectuer le calcul en fonction de la taille du contexte (16 px par défaut) en faisant 12 / 16 = 0,75 em.

Gérer son ergonomie est important !

Nous ne traiterons pas en détail tous ces facteurs mais essentiellement ceux qui posent le plus fréquemment des problèmes d'optimisation et qui permettent parfois assez facilement de gagner des points importants sur la note finale du PageSpeed. Porter attention à l'optimisation de ces facteurs a aussi un impact sur l'expérience utilisateur voire sur le taux de clics et de conversions (en cas de vente).

Google préconise souvent le responsive web design mais par expérience, nous pouvons remarquer que de nombreux sites adaptatifs reçoivent une note de PageSpeed moyenne (70/100 à 80/100) car cette technique ne permet pas de répondre à tous les facteurs, notamment ceux concernant la taille des images et leur poids. Il convient alors de créer des images pour chaque format de site web afin qu'elles soient mieux optimisées, mais cela prend souvent beaucoup de temps et en général, les CMS ou frameworks ne sont pas préparés pour ce genre de pratique…

Si l'outil de Google ne vous suffit pas, vous pouvez vous référer à d'autres logiciels en ligne ou extensions de navigateur afin de calculer la note du PageSpeed. Par exemple, vous pouvez installer l'outil GTMetrix ou encore les extensions PageSpeed et YSlow pour Firefox et Chrome notamment.

Nous avons vu que certains critères du PageSpeed nécessitent des explications plus approfondies pour être parfaitement appliqués. Nous allons donc détailler certains d'entre eux afin que vous puissiez effectuer un minimum d'optimisations plus ou moins simples dans vos sites web.

Retenez toutefois que les frameworks et CMS constituent souvent des freins à l'optimisation de la vitesse de chargement des pages. Si cela ne rend pas la tâche impossible, ces outils ne la facilitent pas et les notes sont souvent un tout petit peu moins qualitatives par défaut sur des sites ainsi conçus, bien que cela soit rattrapable et parfois mesuré.

Figure 2-10

Analyse d'une page web avec l'extension YSlow sur Mozilla Firefox

Éviter les redirections vers la page de destination

Il convient d'éviter au maximum les redirections qui renvoient vers un site mobile une fois le site original chargé. Souvent, nous créons un système de détection de la largeur des fenêtres en JavaScript avec des expressions régulières ou la fonction `matchMedia()` :

```
<script>
if (window.matchMedia("(max-width:640px)").matches) {
// Code effectué si la largeur est inférieure à 640 px
} else {
// Code effectué si la largeur est supérieure à 640 px
}
</script>
```

Parfois, il s'agit d'une détection en PHP des user-agent afin de rediriger les internautes vers la version de site adéquate, comme dans l'exemple suivant :

```
function isMobile(){
    $agent = $_SERVER['HTTP_USER_AGENT'];
    // Effectue un test pour savoir s'il s'agit d'un mobile ou non (exemples)
    return preg_match('/(iphone|android|symbian|palm|blackberry)/iU', $agent);
}
```

Cependant, si ces techniques sont parfois de bonne augure et pratiques, elles imposent généralement un chargement de tout ou partie de la première page avant la redirection, ce qui multiplie les requêtes inutiles.

Plusieurs méthodes permettent d'éviter les redirections mal effectuées :

- opter pour une mise en page et une mise en forme adaptatives à l'aide du responsive web design en HTML 5 et CSS 3 ;
- créer une application mobile native dans les langages de chaque support (Objective-C, Java...) ou via des systèmes de compilation de codes HTML, CSS et JavaScript (PhoneGap le permet, par exemple) afin d'avoir un site détaché d'une version applicative sur mobile (bien que cela ne rende pas le site compatible sur les navigateurs mobiles pour autant...) ;
- créer un site mobile sous la forme d'un sous-domaine tel que m.site-mobile.fr ou d'un nom de domaine propre comme site-mobile.mobi. Ce cas nécessite donc une redirection à effectuer proprement afin de limiter les requêtes du serveur.

La première étape consiste à ajouter une balise `<link rel="alternate" href="url-mobile" />` dans la section `<head>...<head>` de la page HTML destinée aux grands écrans. Vous pourrez ainsi indiquer à Googlebot l'existence d'une version mobile à prendre en compte en fonction des dimensions de l'écran. Voici comment procéder :

```
<link rel="alternate" media="only screen and (max-width:640px)" href="http://m.site.fr" />
```

La seconde étape consiste à réaliser la redirection vers le site mobile à partir d'un fichier `.htaccess` en lui ajoutant des conditions (reconnaissance des agents et donc des mobiles). Il s'agit d'une règle simple de réécriture, nous reviendrons plus en détail sur la réécriture d'URL plus loin dans ce chapitre. Voici un code fonctionnel qui évitera des pertes de chargement pour les serveurs et les robots d'indexation :

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} "ipod|iphone|ipad|android|palm|IEmobile|Windows Phone" [NC,OR]
RewriteRule (.*) http://m.votredomaine.com [R=301,L]
```

Le cas de Windows Phone

Il est parfois difficile de détecter les user-agent de Windows Phone car ils changent d'une version à l'autre et sont parfois reconnus comme des téléphones Android. Le site webapps-online.com a listé des possibilités et permet de mieux comprendre les agents compatibles avec ce système d'exploitation (source : http://goo.gl/o1ekwq).

Autoriser la compression

Le module `mod_deflate` des serveurs Apache permet d'ajouter un filtre de sortie qui autorise la compression au format Gzip des données avant de les envoyer aux clients. Son rôle est donc de réduire le temps de latence au chargement des informations. Pour ce faire, il suffit de copier-coller un code dans le fichier `.htaccess` situé à la racine du serveur (c'est suffisant en général), tel que celui-ci :

```
<IfModule mod_deflate.c>
# Autoriser la compression uniquement pour ces types MIME
# Possibilité  d'ajouter : SetOutputFilter DEFLATE
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE text/JavaScript
AddOutputFilterByType DEFLATE application/JavaScript
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/rss+xml
AddOutputFilterByType DEFLATE application/atom_xml
AddOutputFilterByType DEFLATE application/x-JavaScript
AddOutputFilterByType DEFLATE application/x-httpd-php
AddOutputFilterByType DEFLATE application/x-httpd-fastphp
AddOutputFilterByType DEFLATE application/x-httpd-eruby
AddOutputFilterByType DEFLATE image/svg+xml

# Degré de compression des données (de 1 à 9)
DeflateCompressionLevel 9

# Résolution des problèmes rencontrés avec d'anciens navigateurs
BrowserMatch ^Mozilla/4 gzip-only-text/html
BrowserMatch ^Mozilla/4\.0[678] no-gzip
BrowserMatch \bMSI[E] !no-gzip !gzip-only-text/html

# Application du module mod_deflate à partir des extensions de fichier
<IfModule mod_mime.c>
AddOutputFilter DEFLATE js css htm html xml
</IfModule>

# Évite que les proxies délivrent des contenus inadéquats
<IfModule mod_headers.c>
Header append Vary User-Agent env=!dont-vary
</IfModule>
</IfModule>
```

Pour vérifier que le module `mod_deflate` est fonctionnel et a été pris en compte sur votre site web, il suffit de contrôler les en-têtes HTTP et les lignes `content-encoding: gzip` et `accept-encoding: gzip, deflate`. Pour cela, vous pouvez utiliser l'extension Firebug sur Firefox. Il suffit de charger la page du site, d'activer Firebug et de cliquer sur *Réseau>HTML*. Un certain nombre d'informations apparaîtra dans la section En-têtes relative à votre page ou nom de domaine.

Figure 2-11

Vérification des en-têtes HTTP et de la compression Gzip avec le module mod_deflate

Exploiter la mise en cache du navigateur

La mise en cache des données est l'un des éléments fondamentaux du PageSpeed et il est conseillé de bien la mettre en œuvre pour obtenir de vrais gains de performance. Dans les principaux CMS, nous utilisons parfois des extensions, les plus connues étant certainement W3 Total Cache ou WP Super Cache sur WordPress. Mais lorsqu'il s'agit d'optimiser manuellement le cache, les choses se compliquent souvent…

Plusieurs facteurs peuvent entrer en ligne de compte dans la gestion du cache côté serveur :

- la gestion des ETags (pour *entity tags*), c'est-à-dire une ligne d'en-tête encodée et unique qui change lorsque l'information est mise à jour et demandée sur le serveur. Soit nous devons supprimer les ETags pour éviter des téléchargements inutiles d'informations (quand aucune mise à jour ne l'impose réellement), soit nous pouvons ajouter une sorte de filtre aux ETags afin de ne récupérer les données qu'en cas de besoin (on se base en général sur la taille des ressources pour comparer les informations mises à jour ou non) ;
- le module `mod_expires` des serveurs Apache permet d'ajouter des dates d'expiration du cache sur des extensions et fichiers donnés s'il est activé. Ainsi, nous pouvons préciser que des types de fichiers doivent être mis en cache pendant au moins une semaine, un mois voire un an si cela semble judicieux ;
- une alternative au module `mod_expires` consiste à utiliser le Cache Control du module `mod_headers`. Il n'est pas nécessaire d'appliquer les deux car ils sont redondants, mais si cela vous rassure et vous permet d'assurer une bonne mise en cache, n'hésitez pas à ajouter les deux blocs de code au sein du fichier `.htaccess`.

Voici un code complet contenant toutes les possibilités présentées :

```
# Suppression des ETags ou deux variantes :
# FileETag Size pour comparer les données par leur taille
# FileETag MTime Size pour comparer par la taille et la date de mise à jour
FileETag none
Header unset ETag
```

```
# Ajout des dates d'expiration du cache
<IfModule mod_expires.c>
ExpiresActive On
ExpiresDefault "access plus 31536000 seconds"
ExpiresByType image/jpg "access plus 1 months"
ExpiresByType image/png "access plus 1 months"
ExpiresByType image/jpeg "access plus 1 months"
ExpiresByType image/gif "access plus 1 months"
ExpiresByType text/ico "access plus 1 months"
ExpiresByType image/ico "access plus 1 months"
ExpiresByType image/icon "access plus 1 months"
ExpiresByType image/x-icon "access plus 1 months"
ExpiresByType text/css "access plus 30 days"
ExpiresByType text/JavaScript "access plus 30 days"
ExpiresByType text/html "access plus 15 days"
ExpiresByType application/xhtml+xml "access plus 2592000 seconds"
ExpiresByType application/JavaScript "access plus 2592000 seconds"
ExpiresByType application/x-JavaScript "access plus 2592000 seconds"
ExpiresByType application/x-shockwave-flash "access plus 2592000 seconds"
</IfModule>

# Gestion du Cache Control (si nécessaire)
<FilesMatch "\.(ico|pdf|flv|jpg|jpeg|png|gif|swf|mp3|mp4|mpeg|avi|asf)$">
Header set Cache-Control "max-age=2592000, public"
</FilesMatch>
<FilesMatch "\.(html|htm|xml|txt|xsl|svg|)$">
Header set Cache-Control "max-age=604800, must-revalidate"
</FilesMatch>

# Désactive le contrôle du cache pour les fichiers dynamiques
<FilesMatch "\.(pl|php|asp|aspx|py|cgi|spl|scgi|fcgi)$">
      Header unset Cache-Control
</FilesMatch>
</IfModule>
```

Gestion des durées d'expiration

Nous pouvons remarquer que les durées appliquées au module `mod_expires` peuvent être rédigées en `seconds`, `days`, `months`, etc. Il suffit d'adapter les valeurs en fonction des besoins et de la note du PageSpeed pour trouver la solution adéquate. Ici, les directives `1 months`, `30 days` et `2592000 seconds` sont donc équivalentes.

Dans certains cas, il est intéressant de créer des fichiers avec une partie du nom correspondant à un hachage automatique dès que le fichier est mis à jour. Par exemple, au lieu d'avoir le fichier `style.css`, nous aurions le fichier `style_A1H7SH.css` à la place qui se mettrait en cache, puis si une mise à jour avait lieu, le fichier s'appellerait alors `style_B1D8G5.css`, par exemple, et ainsi de suite. L'avantage est que le fichier sortira du cache uniquement lorsque son hachage sera différent, ce qui permet de donner des dates d'expiration plus longue sans aucun risque.

Réduire la taille des ressources

La réduction de la taille des ressources s'effectue simplement en supprimant tous les espaces vides dans les codes sources ainsi que tous les points-virgules (;) inutiles dans les feuilles de styles. Google veille avant tout à ce que les fichiers HTML, JavaScript et CSS soient compressés au maximum pour optimiser la vitesse de lecture et d'affichage des pages web.

Ce facteur est relativement simple à mettre en œuvre et permet de gagner quelques millisecondes non négligeables pour chaque chargement de page alors n'hésitez pas à le faire.

Il existe de nombreux outils et extensions pour faciliter les tâches de compression des ressources web, mais attention aux options et aux problèmes que cela peut parfois engendrer.

- Réduction des fichiers CSS :
  - CleanCSS : http://www.cleancss.com ;
  - YUI Compressor : http://goo.gl/31CPgc ;
  - CSSCompressor : http://www.csscompressor.com ;
  - CSS Minifier : http://cssminifier.com ;
  - WP Minify pour WordPress (avec son addon WP Minify Fix) : http://wordpress.org/extend/plugins/wp-minify/ (gare aux bugs d'affichage dans le backoffice) ;
  - Better WordPress minify : http://goo.gl/gquVxo ;
- Réduction des ressources HTML :
  - HTML Compressor : http://www.miniwebtool.com/html-compressor/ ;
  - TextFider : http://www.textfixer.com/html/compress-html-compression.php ;
  - HTML Minify pour WordPress : http://goo.gl/DLI4OU ;
- Compression des fichiers JavaScript :
  - JavaScript Compressor : http://JavaScriptcompressor.com ;
  - YUI Compress : http://refresh-sf.com/yui/ ;
  - JS Compress : http://jscompress.com ;
  - JavaScript Minifier : http://JavaScript-minifier.com ;

Il existe également des outils qui permettent de réaliser les trois types de compressions :

- TinyFier : http://www.tinyfier.com ;
- Compress My Code : http://compressmycode.com ;
- HTML Minifier : http://www.willpeavy.com/minifier/.

La manipulation est simple puisqu'elle se réduit à réaliser un simple copier-coller des codes sources dans ces outils, puis à récupérer le code compressé. L'idéal est de se créer un fichier de travail sans compression et un fichier final compressé afin de l'envoyer sur le serveur. Ainsi, le fichier d'origine permet de mettre à jour les codes si nécessaire, et il suffit alors de compresser à nouveau les codes pour obtenir de bons résultats.

Figure 2-12

Compression du code CSS avec CSS Minifier

Attention aux outils de compression

Il arrive parfois que la compression des codes CSS pose des problèmes au niveau des fonctions CSS comme `@media` qui permet de développer un site au design adaptatif. En effet, certains outils comprennent mal les fonctions et coupent en partie le code, il faut donc veiller au bon fonctionnement du code final et à la qualité de la compression...

Optimiser les images

L'optimisation des images est relativement simple à comprendre mais pas toujours évidente à mettre en application, notamment pour tous les utilisateurs de frameworks ou CMS. En effet, plusieurs facteurs sont à prendre en compte ici.

- Les images doivent avoir une taille équivalente à celle affichée dans les pages web, ce qui signifie que les outils ou les créateurs de sites doivent penser à adapter les dimensions en fonction de la version de site utilisée (mobiles, tablettes…).
- Les illustrations doivent être compressées au maximum en fonction de leurs dimensions initiales afin de réduire leur poids et donc le temps de chargement. Une fois encore, le fait d'avoir des images à la taille de chaque type de support permet de ne désavantager aucun utilisateur dans sa visite du site web.
- L'usage des sprites CSS s'avère primordial pour gagner en légèreté et en rapidité d'exécution dans les pages. Cela consiste à créer une ou plusieurs images qui regroupent une multitude de petites icônes. Par exemple, nous pouvons créer un fichier avec la méthode des sprites CSS qui contient l'ensemble des petits boutons de partage vers les réseaux sociaux et leur effet de survol. Ainsi, un seul fichier (plus léger) est chargé au démarrage du site et affiche l'ensemble des boutons grâce au code CSS. Cette technique évite l'effet désagréable de « blanc » lors du survol d'un bouton qui aurait été dissocié en deux images : une pour l'état normal, une pour l'état survolé.

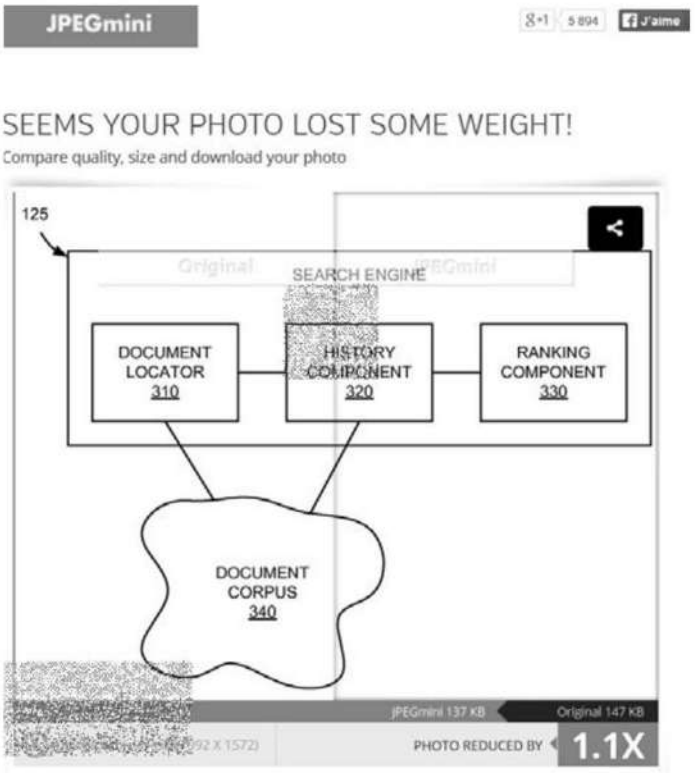
La compression des fichiers peut se faire à l'aide de logiciels, dont certains sont même proposés dans la documentation de Google :

- Jpeg-Optimizer : http://jpeg-optimizer.com ;
- JPEGmini : http://www.jpegmini.com ;
- Compressnow : http://compressnow.com/fr/ ;

- RIOT : http://luci.criosweb.ro/riot/ ;
- ImageOptim : http://imageoptim.com ;
- PNGGauntlet : http://pnggauntlet.com ;
- OptiPNG : http://optipng.sourceforge.net ;
- TinyPNG : https://tinypng.com ;
- PunyPNG : http://www.punypng.com ;
- PNGOUT : http://www.advsys.net/ken/util/pngout.htm.

Figure 2-13

Compression d'un fichier .jpg avec JPEGmini



Le recadrage des images peut aussi s'effectuer avec n'importe quel logiciel graphique comme Adobe Photoshop, Gimp, Paint Shop Pro, Pixen, Krita, PhotoFiltre Studio ou encore Picasa. Il existe également des alternatives proposées directement dans certains CMS comme WordPress ou par le biais d'extensions sur Joomla. Ainsi, vous pouvez à tout moment créer des images à la taille souhaitée sans aucune difficulté.

Enfin, la technique des sprites CSS s'applique en créant une image regroupant plusieurs images ou plusieurs boutons en une seule. Il faut faire attention à la gestion des transparences de certaines images et il est donc recommandé d'enregistrer au format PNG (le format GIF étant conseillé uniquement pour des images de moins de 10 × 10 pixels comme le préconise Google à juste titre).

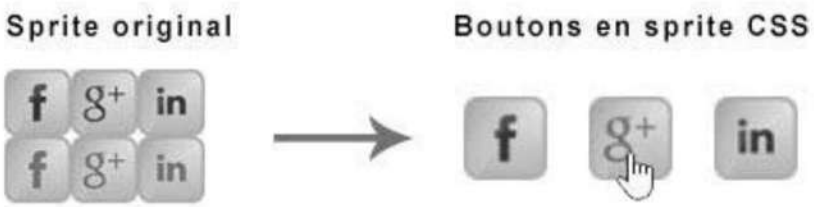
Prenons l'exemple d'une image réalisée en sprite CSS contenant trois boutons pointant vers des réseaux sociaux. Il faut tout d'abord ajouter le code HTML correspondant afin de rendre les boutons cliquables, puis un code CSS adapté à l'image globale. Il s'agit ici de petites icônes de 32 × 32 pixels pour les états non survolés et survolés, ce qui représente une image finale de 96 × 64 pixels une fois tous les boutons ajoutés.

Figure 2-14

Redimensionnement et recadrage en natif dans WordPress

Figure 2-15

Exemples de boutons avec rendu final en sprite CSS



Le code HTML suivant permet de rendre les boutons fonctionnels, et les identifiants (`id`) vont nous permettre de réaliser la technique des sprites CSS :

```
<div id="social">
    <p id="facebook"><a href="URL_FACEBOOK"><span>Facebook</span>
</a></p>
    <p id="googleplus"><a href="URL_GPLUS"><span>Google+</span>
</a></p>
    <p id="linkedin"><a href="URL_LINKEDIN"><span>LinkedIn</span>
</a></p>
</div>
```

Ensuite, voici un exemple de code CSS permettant de mettre en œuvre la méthode des sprites CSS. Il repose sur l'emploi de la propriété `background-position` ou de sa propriété mère `background` (comme c'est le cas ici) avec des positionnements adaptés selon la zone de l'image générale que nous souhaitons afficher :

```
#facebook a span, #googleplus a span, #linkedin a span {display:none;}
#facebook a {display:block; float:left; height:32px; width:32px; background:url(sprite.png)
no-repeat 0 top; margin-right:1em;}
#facebook a:hover {background:url(sprite.png) no-repeat 0 bottom;}
#googleplus a {display:block; float:left; height:32px; width:32px;
background:url(sprite.png) no-repeat -32px top; margin-right:1em;}
#googleplus a:hover {background:url(sprite.png) no-repeat -32px bottom;}
#linkedin a {display:block; float:left; height:32px; width:32px;
background:url(sprite.png) no-repeat -64px top;}
#linkedin a:hover {background:url(sprite.png) no-repeat -64px bottom;}
```

L'ensemble de ces optimisations permet de gagner des millisecondes voire des secondes de chargement pour chaque page du site. La note du PageSpeed sera ainsi améliorée. Veillez à apporter beaucoup de soin au traitement des images sur le Web afin d'optimiser l'expérience utilisateur et les performances globales.

Utiliser des scripts asynchrones optimisés

Plusieurs méthodes permettent d'utiliser des scripts asynchrones, nous ne pourrons pas toutes les décrire ici tant elles sont variées et parfois complexes à mettre en œuvre pour les plus débutants. L'objectif est d'éviter au maximum de charger des codes qui n'ont pas d'intérêt direct pour le bon fonctionnement et l'affichage des pages web. Par exemple, nous pouvons citer le cas des codes JavaScript permettant de faire fonctionner Google Analytics. En effet, il n'est pas gênant du tout que ces scripts soient exécutés après que toute la page a été chargée.

Notons que le PageSpeed est parfois un peu « bête et méchant » et son intolérance va à l'encontre du bon fonctionnement général de certaines pages. En effet, si nous respectons à la lettre les règles édictées par la documentation de Google, nous risquons parfois d'avoir des surprises désagréables en matière d'affichage et de fonctionnalités. Il faut donc être prudent et effectuer des tests progressifs en actualisant les pages web concernées pour savoir si nous n'allons pas trop loin dans l'optimisation.

C'est l'un des défauts du PageSpeed car la quasi-totalité des sites web ne pourra jamais atteindre la note de 100/100 malgré toute la bonne volonté du monde. En effet, il existe presque dans tous les cas un script qui doit être chargé au démarrage et non de manière asynchrone ou en fin de code source.

Il faut donc accepter que l'expérience utilisateur passe avant la note attribuée par Google et bien que nous perdions des points de façon purement subjective, ce n'est pas un drame. Les meilleurs exemples de chargements asynchrones à éviter concernent les sliders, carrousels ou encore la tendance des *smooth scrolls* (pages découpées en sections distinctes avec un effet de scroll continu) comme le montre la figure 2-16.

Pour respecter au maximum les règles fixées par le PageSpeed, nous pouvons utiliser plusieurs méthodes indépendamment ou conjointement quand c'est possible.

- Placer les balises `<script type="application/JavaScript">...</script>` en fin de code source, avant la balise `</html>`. Cela peut également s'appliquer aux balises de style `<link rel="stylesheet" type="text/css" href="style.css" />` si vous voulez être encore plus performant.
- Utiliser des requêtes Ajax asynchrones avec ou sans jQuery (conseillé si vous débutez en code). Nous étudierons un cas par la suite. Retenez qu'il s'agit juste de paramétrer l'option `async` sur `true` pour déclencher les scripts Ajax de manière asynchrone.

Figure 2-16

Exemple de chargement asynchrone d'un script de slideshow qui détériore l'affichage.

- Utiliser les options propres à chaque script pour exécuter les codes de façon asynchrone. Chaque bibliothèque, plug-in ou script détient sa propre vérité, il suffit de lire en détail la documentation pour trouver la méthode d'exécution du code de manière asynchrone. Cela sous-entend qu'il est parfois préférable d'opter pour un plug-in plutôt qu'un autre qui ne proposerait pas ce type d'option. Le choix des scripts est donc essentiel en matière d'optimisation du PageSpeed ou plus généralement d'expérience utilisateur.
- Opter pour HTML 5 et les nouveaux attributs `async` ou `defer` qui permettent d'effectuer des chargements asynchrones avec peu d'efforts. Toutefois, il faut retenir que les anciens navigateurs ne sont pas compatibles avec ces attributs et donc qu'il s'agit davantage d'une solution d'avenir très intéressante et simple à mettre en place.
- Utiliser des bibliothèques et chargeurs JavaScript qui permettent de charger facilement des scripts de manière asynchrone. Les plus connus sont LabJS (source : http://labjs.com), Head.js (source :

http://headjs.com), ControlJS (source : http://stevesouders.com/controljs/) et RequireJS (source : http://requirejs.org). Nous détaillerons la méthode avec LabJS par la suite.

- Passer par la fonction `load()` de jQuery qui permet dans certains cas de sauver les meubles. Cependant, cette technique n'est pas la meilleure dans la plupart des circonstances, il ne faut donc pas la considérer comme essentielle. Il en va de même pour toutes les techniques qui visent à exécuter les codes après quelques secondes à l'aide de timeout en JavaScript ou jQuery. Ces méthodes sont trop limitées et provoquent même parfois l'effet inverse en conduisant à des pertes de performances.

Maintenant que nous avons présenté plusieurs techniques d'optimisation pour le chargement des scripts, nous pouvons détailler celles qui nous semblent les plus pratiques et les plus efficaces. Commençons tout d'abord par la mise en place de requêtes Ajax asynchrones avec la bibliothèque jQuery. Le code suivant présente une requête extrêmement simple qui porte l'option `async` :

```
jQuery(document).ready(function () {
    $.ajax({
        url: "code-ajax.php",
        type: "POST",
    async: true, // Lancement asynchrone
    data: ({
        donnee1: $("#champ1").val(),
        donnee2: $("#champ2").val(),
        donnee3: "texte d'exemple"
    }),
    success: function(data) {
        // Retourne les données transmises en Ajax
        $("#zoneaffichageresultat").append(data);
    },
    complete: function(data){
        console.log(data);
    }
    });
});
```

L'HTML 5 avec ses nouveaux attributs `async` et `defer` peut aussi être considéré comme une bonne solution, tout du moins si vous souhaitez vous tourner vers l'avenir et les nouvelles méthodes de codage. Il s'agit sans aucun doute de la technique la plus simple à appliquer puisqu'il s'agit d'ajouter l'attribut dans la balise ouvrante `<script>`, comme dans l'exemple suivant :

```
<script type="application/JavaScript" src="jquery-min.js" async></script>
<script type="application/JavaScript" src="jquery-min.js" defer></script>
```

Il convient d'indiquer `async="async"` ou `defer="defer"` si vous optez pour la syntaxe xHTML (en développement), sachant que `defer` existait dans d'anciennes versions d'Internet Explorer.

Il est difficile de savoir quel est le meilleur attribut pour l'optimisation. Par exemple, `defer` est plus compatible, charge les scripts dans l'ordre désiré mais peut bloquer temporairement l'affichage des contenus de la page en attente du chargement final.

En revanche, `async` permet de charger le contenu directement et rapidement mais il n'est pas compatible partout et ne charge pas toujours les scripts dans l'ordre, ce qui peut s'avérer très gênant si un script nécessite le chargement préalable d'une bibliothèque, par exemple. Libre à chacun de se faire sa propre opinion, Microsoft a même recommandé d'ajouter les deux attributs et dans ce cas, les navigateurs compatibles optent pour `async`.

Une autre méthode intéressante pour charger les scripts JavaScript de manière asynchrone s'appuie sur des plug-ins comme LabJS. Ces méthodes sont relativement simples à exécuter, il suffit d'utiliser la méthode `wait()` ou d'ajouter une option (`AlwaysPreserveOrder`) pour préserver l'ordre de chargement des scripts.

```
<!-- Première méthode avec wait() -->
<script src='/js/LAB.min.js'></script>
<script>
$LAB
.script("js/jquery-min.js").wait()
.script("js/code-jquery-min.js")
</script>

<!-- Seconde méthode avec AlwaysPreserveOrder -->
<script src='/js/LAB.min.js'></script>
<script>
$LAB
.setOptions({AlwaysPreserveOrder:true})
.script("js/jquery-min.js")
.script("js/code-jquery-min.js")
</script>
```

Enfin, la dernière étape du travail d'optimisation des scripts consiste à réduire les ressources et surtout le nombre de fichiers CSS et JavaScript, par exemple. Le plus souvent, cela se fait manuellement par de simples copier-coller mais il existe aussi une méthode dynamique avec une bibliothèque PHP comme avec YUI Compressor ou Minify (source : https://code.google.com/p/minify/). Consultez la documentation si ces méthodes vous intéressent car elles permettent de combiner les fichiers en deux appels seulement pour tous les fichiers JS et CSS.

Mod_pagespeed

Nous ne pouvons pas parler d'optimisation du PageSpeed sans mentionner `mod_pagespeed` (source : http://goo.gl/bz0P68), un module créé de toutes pièces par Google pour améliorer les performances côté serveur (sur Apache).

Le principal problème est que cette méthode n'est applicable que pour les détenteurs de serveurs dédiés car il faut installer le module sur le serveur Apache, puis le paramétrer en fonction des besoins. Si vous avez cette possibilité, faites-le sans hésiter. Vous gagnerez grandement en termes de performances et cela vous évitera parfois certaines optimisations sur les fichiers `.htaccess`. À noter toutefois qu'il vous faudra de bonnes bases techniques pour réaliser la démarche sans difficulté.

La documentation de Google est plutôt bien faite à ce sujet. Vous trouverez un site dédié (source : http://www.modpagespeed.com) qui permet de mieux comprendre comment installer le module Apache sous Debian/Ubuntu ou CentOS/Fedora. Voici comment procéder pour installer les packs .deb ou .rpm sur Debian/Ubuntu (64 bits) :

```
wget https://dl-ssl.google.com/dl/linux/direct/mod-pagespeed-stable_current_amd64.deb
sudo dpkg -i mod-pagespeed-*.deb
sudo apt-get -f install
```

Il est également possible de télécharger manuellement le fichier .deb et de l'installer sans la commande `wget`, cela revient au même.

Ensuite, il est courant de devoir appliquer deux autres commandes :

- recharger le serveur Apache avec `service apache2 restart` pour l'activer (bien que cela soit fait par défaut en général) ;
- supprimer le pack téléchargé avec `rm mod-pagespeed-*.deb (optionnel)`.

Ensuite, il convient de configurer le module grâce au fichier `pagespeed.conf` (accessible avec la commande `nano /etc/apache2/mods-available/pagespeed.conf`) installé avec le module `mod_pagespeed`. C'est ce dernier qui permet d'appliquer nombre d'options utiles à l'optimisation de la vitesse et des performances du serveur. Le module apporte quelques améliorations visibles :

- mise en cache des illustrations ;
- compression de tous les fichiers CSS et JavaScript ;
- modification des noms de fichiers CSS et JavaScript.

Il est aussi possible de configurer certaines directives à l'aide d'un fichier `.htaccess`, mais cette méthode implique des chargements répétés de requêtes. Il est conseillé de configurer `pagespeed.conf` en natif pour optimiser réellement les performances.

Voici une liste de paramètres intéressants qui peuvent être modifiés pour booster votre site avec le module de Google (il faut redémarrer Apache pour que les effets soient pris en compte).

- Démarrer le module `mod_pagespeed` : `ModPagespeed on`.
- Activer des filtres complémentaires :

```
ModPagespeedEnableFilters filtre1,filtre2
```

- Désactiver des filtres installés :

```
ModPagespeedDisableFilters filtre1,filtre2
```

- Autoriser la réécriture de certains types de fichiers (exemple des fichiers HTML ici, même si cela est le cas par défaut) :

```
ModPagespeedAllow "http://*site.fr/*.html".
```

- Supprimer la réécriture de certains types de fichiers (ici, tous les types) :

```
ModPagespeedDisallow "*".
```

- Optimiser la bande passante lors de la réécriture d'URL :

```
ModPagespeedRewriteLevel OptimizeForBandwidth.
```

- Configurer le fichier de gestion du cache comme ceci :

```
ModPagespeedFileCachePath "/var/cache/pagespeed/"
ModPagespeedFileCacheSizeKb 102400
ModPagespeedFileCacheCleanIntervalMs 3600000
ModPagespeedFileCacheInodeLimit 500000
```

Beaucoup d'autres plug-ins permettent d'obtenir de meilleurs résultats mais ce module créé par Google est mis à jour fréquemment et il faut suivre son actualisation pour optimiser idéalement vos performances de site.

Il est certain que la note du PageSpeed est rehaussée lorsque le module `mod_pagespeed` est bien installé et configuré. Mais vous pouvez aussi obtenir de très bons résultats sans avoir de serveur dédié et ce module spécifique, rassurez-vous…

Gérer des redirections

Les redirections font partie des techniques essentielles à maîtriser lorsque nous créons un site ou que nous devons l'optimiser à des fins de référencement. En général, les nouveaux sites n'ont besoin de redirections que pour éviter les contenus dupliqués ou pour relier les différents noms de domaines représentant le même site.

Au contraire, les sites anciens ou ceux qui ont subi de lourdes refontes sombrent souvent face à la masse de contenus dupliqués ou d'URL disparues qui ne trouvent plus preneur. En effet, il arrive fréquemment que ces sites aient encore un nombre incalculable de pages indexées dans les SERP et que ces résultats soient des liens morts puisque les pages ont été détruites et remplacées par les nouvelles versions issues de la refonte du site. Il est également possible que l'ancienne version du site soit encore en place et que le visiteur ne soit donc pas dirigé vers la bonne information. Dans tous les cas, les résultats d'une refonte peuvent être catastrophiques en termes de SEO et entraîner plusieurs problèmes :

- perte importante de trafic (et conséquences relatives comme la baisse des ventes, etc.) ;
- multiplication de pages en doublon (avec ou sans *duplicate content*) ;
- perte de qualité en termes d'indexation.

Il convient donc de remédier à tout cela et les redirections sont là pour ça. Certains CMS proposent des extensions de qualité tels que WordPress avec Redirection (source : http://goo.gl/xe3PFl), Joomla (source : http://goo.gl/9Uc9Mo), Prestashop avec Duplicate URL Redirect (source : http://goo.gl/GQ87Uo), Magento avec Optimise Web's Mass 301 Redirect (source : http://goo.gl/jjGU8M) ou Drupal avec Global Redirect

(source : http://goo.gl/iLvMxK). Néanmoins, il arrive fréquemment que ces outils ne répondent pas à tous nos besoins et que nous devions effectuer le travail manuellement.

La première règle à retenir est que les redirections doivent absolument être permanentes (code 301) et non temporaires (code 302) car Google pourrait comprendre cela comme une méthode de triche (du *cloaking*, notion sur laquelle nous reviendrons en détail dans le prochain chapitre). Il faut donc veiller à réaliser des redirections de qualité pour ne pas être pénalisé et surtout rendre le renvoi fonctionnel vers les nouvelles pages.

Cela peut se faire simplement à l'aide de fonctions PHP. Ainsi, il suffit d'ajouter un code comme le suivant dans les codes sources des anciennes pages. Cependant, cela peut vite s'avérer fastidieux.

```
<?php
header("HTTP/1.1 301 Moved Permanently");
header("Location:http://www.nouveausite.fr");
exit;
?>
```

La meilleure solution reste une nouvelle fois la création d'un fichier `.htaccess` qui recense l'ensemble des redirections permanentes utiles à la racine de l'ancien site web, comme ceci :

```
# Redirect 301, Redirect permanent et RedirectPermanent sont identiques
Redirect 301 /vieille-page-1.html http://www.site.fr/nouvelle-page-1.html
RedirectPermanent /vieille-page-2.html http://www.site.fr/nouvelle-page-2.html
Redirect permanent /dossier http://www.site.fr/dossier/
```

Écriture raccourcie des redirections

Si le répertoire de la page d'origine et celui de la page cible est le même (donc la racine dans notre exemple), il n'est pas nécessaire d'inscrire le nom de domaine pour l'ancienne page.

Cette méthode est parfaite pour rediriger d'anciennes pages web voire des dossiers complets, mais cela ne répond pas toujours à nos besoins. Sachez également qu'il est possible d'indiquer aux robots des moteurs de recherche qu'un document n'est plus accessible de manière définitive grâce à la directive `Redirect gone`.

```
Redirect gone / fichier-supprime.html
Redirect gone /dossier-supprime/
```

Nous pouvons aller encore plus loin dans les redirections à l'aide de la directive `RedirectMatch` qui accepte des expressions régulières. Ainsi, nous pouvons rediriger des multitudes de fichiers d'un seul coup vers un dossier précis, par exemple, ou une page de destination (comme une page d'erreur personnalisée pour indiquer qu'un nouveau site a été créé). Voici deux exemples de redirections permanentes avec `RedirectMatch` :

```
# Redirection de tous les fichiers HTML vers leur équivalent (du même nom)
# portant désormais l'extension .php
RedirectMatch permanent /(.*)\.html$ http://www.site.fr/$1.php

# Déplacement vers le nouveau site pour les pages d'un dossier
RedirectMatch permanent /dossier/(.*)$ http://www.site.fr
```

Nous venons de le voir, il arrive parfois que ce soit seulement l'extension des pages web qui change de .htm à .html ou de .html à .php, par exemple. Dans ces conditions, les redirections classiques ont peu d'intérêt et dans les exemples, le mot-clé `seeother` devrait remplacer `permanent` pour être plus précis :

```
# Avec RedirectMatch pour changer les .jpg en .png
RedirectMatch seeother /images(.*)\.jpg$ http://www.site.fr/images/$1.png

# Avec Redirect pour changer un document DOC en PDF
Redirect seeother /document.doc gttp://wwwsite.fr/document.pdf
```

Le mot-clé `seeother` est l'équivalent du code 303 de redirection ; cela correspond donc à `RedirectMatch 303`.

Il arrive également que nous déplacions le site sur le serveur de la racine vers un sous-répertoire. Dans ce cas, toutes les pages s'en ressentent, mais il est assez simple d'effectuer les redirections. En effet, il suffit d'écrire la ligne suivante :

```
RedirectMatch 301 (.*) http://www.site.fr/dossier-site/
```

Équivalence d'écritures

Il existe des écritures équivalentes comme `Redirect 301 / http://www.site.fr/dossier-site/`.

En revanche, il peut arriver que les déplacements de fichiers soient plus subtils au sein du serveur. Nous déplaçons parfois uniquement les pages web, les images, les scripts et autres fichiers restants à la racine. Dans ce cas, il faut uniquement déplacer les types de fichiers correspondant à des pages web avec `RedirectMatch` :

```
RedirectMatch permanent /(.*)\.(html|htm|php|py|asp|aspx)?$ http://www.site.fr/
dossier-site/
```

Une fois encore, d'autres nombreuses subtilités peuvent concerner certains sites web. Il serait impossible de toutes les lister, mais en voici quelques-unes pour vous permettre de bien maîtriser les redirections permanentes avec les fichiers `.htaccess` :

- déplacer un dossier à la racine sans impliquer les sous-dossiers :

```
RedirectMatch permanent /dossier/([^/\\]*)$ http://www.site.fr/$1
```

- déplacer un dossier à la racine, mais pas certains fichiers qu'il contient, à l'aide de l'assertion négative (?!) :

```
RedirectMatch 301 /dossier/(?!page\.php|img\.png)(.*)$ http://www.site.fr/$1
```

- éviter les problèmes de casse dans le nom des fichiers avec l'assertion (?i) :

```
RedirectMatch 301 ^/(?i)Sans-Casse\.html$ http://www.site.fr/page.html
```

Désormais, vous connaissez l'essentiel des règles de redirection propres aux fichiers .htaccess afin d'éviter tout problème de contenus dupliqués ou d'URL erronées. Force est de constater que certains cas ne sont pas simples à mettre en œuvre, mais ils vont souvent plus loin que les extensions disponibles dans les divers CMS du marché. Nous pouvons parfois gagner beaucoup de temps en couplant des types de redirections différentes grâce aux extensions de fichiers, aux exclusions des sous-répertoires ou encore aux exclusions de certains fichiers.

Sur ce point, la documentation de Google manque nettement de précision et ne permet pas de pousser aussi loin les redirections (source : http://goo.gl/dDfK4T). Il convient donc de s'y intéresser et d'effectuer des tests approfondis pour trouver des solutions adéquates en cas de refonte ou de déplacement d'un site. Nous n'avons pas tout traité ici mais il existe également des écritures similaires et tout aussi fonctionnelles sur le Web. N'hésitez pas à vous renseigner en cas d'extrême nécessité plutôt que de prendre le risque d'être pénalisé en termes de trafic et de contenus dupliqués.

Gérer les redirections spécifiques et les codes d'erreurs

Nous venons de détailler l'usage des redirections permanentes. Nous allons à présent nous intéresser au traitement de certains codes d'erreurs, de même qu'à des redirections moins connues mais qui peuvent s'avérer intéressantes dans certains cas.

Pour ceux qui ne connaissent pas en détail les codes d'erreurs, voici une typologie simple à retenir :

- 100 à 101 : codes d'information (sur l'état de la requête et du protocole) ;
- 200 à 206 : codes de succès (réussite de la requête) ;
- 300 à 305 (et 307) : codes de redirection (permanentes, temporaires, déplacements, non modifié, usage d'un proxy) ;
- 400 à 417 : codes d'erreurs du client (dont les très connues erreurs 403 et 404) ;
- 500 à 505 : codes d'erreurs du serveur (erreur interne, service indisponible…).

Maintenant que nos idées sont claires, commençons par les redirections spécifiques qui peuvent être utilisées dans certains cas et dont le rôle peut être important en termes de référencement :

- redirections temporaires (codes 302 et/ou 307 parfois), utiles notamment en cas de test sur les moteurs de recherche ou lorsqu'une page ne va pas exister longtemps (bien que ce soit déconseillé dans ce cas) :

```
# Deux instructions équivalentes avec 302 et temp
Redirect 302 /dossier http://www.site.fr/nouveau-dossier
Redirect temp /page.html http://www.site.fr/page2.html
```

- redirections pour des pages non modifiées afin d'indiquer aux moteurs de recherche que les pages concernées n'ont pas subi de mises à jour. Cela demande parfois une configuration du serveur pour renvoyer l'en-tête HTTP `If-Modified-Since` qui sera lu par les robots et leur permettra d'économiser de la bande passante (source : http://goo.gl/KlySZx). Cette fonctionnalité est très rarement utilisée alors que Google n'en a jamais dit de mal et confirme même dans sa documentation le gain de ressources pour Googlebot. Bien qu'il préfère de loin les redirections permanentes, il peut être intéressant de les utiliser à bon escient.

```
# Déplacement de la page d'accueil (non mise à jour) dans un dossier
Redirect 304 /index.html http://www.site.com/dossier-site/index.html
```

Les codes d'erreurs 400 à 417 concernent les problèmes de chargement des pages ou plutôt des soucis côté client. Toutes ne nécessitent pas une intention particulière mais voici une liste des quelques erreurs qui peuvent avoir un intérêt pour le référencement et l'expérience utilisateur :

- 401 « access denied » : accès non autorisé pour les personnes qui ne sont pas authentifiées (seulement si vous utilisez une connexion à l'aide du fichier `.htaccess`) ;
- 403 « request forbidden » : accès interdit ou refusé par le serveur lorsque nous souhaitons, par exemple, protéger des répertoires ou quand un serveur plante dans certains cas (parfois ce sont des virus qui génèrent ce type de problème…) ;
- 404 « object not found » : page introuvable pour les utilisateurs, c'est l'erreur la plus courante qui apparaît à chaque fois qu'une page est inaccessible ou manquante ;
- 410 « the resource is no longer available » : identique à l'erreur 404 sauf que la page n'existe plus, c'est-à-dire que le serveur sait qu'elle a existé mais ne la retrouve plus et affiche donc une erreur (elle peut avoir un rôle pour le référencement) ;
- 413 « request entity was too large » : le serveur ne peut pas traiter la requête car elle est trop volumineuse (erreur rare) ;
- 414 « request URI too long » : l'URI (une chaîne de caractères qui sert à identifier une ressource, soit l'URL dans notre cas) est trop longue et ne peut pas être traitée correctement. Ce type d'erreur peut se produire lorsque nous avons trop de paramètres dans les URL (cela dit, le problème est rarissime).

Pour afficher des pages d'erreurs en fonction des codes rencontrés sur le Web, il suffit de saisir des lignes telles que les suivantes avec l'URL de la page d'erreur de destination :

```
ErrorDocument 403 http://www.site.fr/403.html
ErrorDocument 404 http://www.site.fr/404.html
ErrorDocument 410 http://www.site.fr/410.html
ErrorDocument 503 http://www.site.fr/503.html
```

Le point essentiel à retenir est l'erreur 410 qui devrait être plus fréquemment utilisée que l'erreur 404 en cas de refonte ou de suppression de pages web. En effet, si Google tombe sur une erreur 404 classique (page manquante ou supprimée), il va mettre un certain temps à la désindexer, même si vous possédez un fichier `robots.txt` correctement conçu. En revanche, si vous lui renvoyez un code erreur 410, il va

accélérer le processus de désindexation car il saura désormais que la page ne reviendra pas et n'existe plus. Pour ce faire, il existe deux méthodes (nous en avons déjà évoqué une dans la partie précédente).

- Renvoyer un en-tête HTTP avec PHP pour préciser que la page n'existe plus. Cela doit être indiqué dans les pages concernées et peut rapidement s'avérer fastidieux :

```
<?php
header("Status:410 Gone", false, 410);
// header('location:410.html'); si vous voulez renvoyer vers une page d'erreur
spécifique pour les utilisateurs
exit();
?>
```

- Utiliser une redirection avec le mot-clé `gone` dans un fichier `.htaccess` qui correspond à l'erreur 410 afin de déclarer un contenu désormais obsolète :

```
Redirect gone /dossier-disparu/
Redirect gone /fichier-disparu.html
```

L'autre code d'erreur à surveiller de près est le 503 car il indique aux moteurs de recherche qu'un site est en maintenance. Cette méthode est recommandée et largement préférable aux classiques « pages en construction » créées de toutes pièces en HTML et qui renvoient un code 200 (donc des pages qui peuvent être indexées alors qu'elles ne proposent aucun contenu).

Dans un autre cas, il arrive parfois que nous procédions à des mises à jour et que le site soit inaccessible temporairement, ce qui peut avoir un impact extrêmement négatif sur le référencement si les robots passent pendant ce laps de temps, aussi court soit-il…

Pour renvoyer une erreur 503 dans une page en maintenance ou en construction, il suffit d'envoyer des en-têtes HTTP via PHP :

```
header('HTTP/1.0 503 Service Temporarily Unavailable');
// ou header('HTTP/1.1 503 Service Temporarily Unavailable');
// ou header('Status: 503 Service Temporarily Unavailable');
header('Retry-After: 3600'); // Retenter après 3600 secondes (1 heure)
// ou header('Retry-After: Sun, 21 Sep 2014 12:00:00 GMT'); // Après une date précise
```

Si nous souhaitons aller plus loin, nous pouvons également utiliser une méthode plus technique avec un fichier `.htaccess` grâce à des réécritures d'URL (sur lesquelles nous allons revenir par la suite). Voici un code complet et commenté pour expliquer le processus :

```
<IfModule mod_rewrite.c>
# Active la réécriture d'URL
RewriteEngine On
# Exclusion de notre propre adresse IP
RewriteCond %{REMOTE_ADDR} !^192\.168\.1\.1
# Vérification de l'existence du fichier de maintenance
RewriteCond %{DOCUMENT_ROOT}/maintenance.html -f
```

```
# Annule l'exécution des règles si nous sommes dans la page de maintenance
RewriteCond %{SCRIPT_FILENAME} !maintenance.html
# Redirection vers la page de maintenance (erreur 503)
RewriteRule ^.*$ /maintenance.html [R=503,L]
ErrorDocument 503 /maintenance.html
</IfModule>
```

Cibler les user-agents

Il est possible d'ajouter une condition supplémentaire pour appliquer les règles uniquement s'il s'agit des moteurs de recherche. Par exemple, l'instruction suivante redirige Googlebot et Bingbot vers une page de maintenance : `RewriteCond %{HTTP_USER_AGENT} (Googlebot|Bingbot) [NC]`.

Il n'existe pas de méthode idéale pour gérer les codes d'erreurs mais il est important de les utiliser avec soin pour se prémunir contre les problèmes d'indexation voire de positionnement. Il peut être utile de créer ses propres fonctions pour activer ou désactiver les pages de maintenance, si nécessaire à l'aide de scripts PHP par exemple. Cela n'est pas compliqué mais les codes précédents vous donneront déjà satisfaction dans la majorité des cas.

Enfin, nous terminerons cette partie sur les erreurs HTTP par un code qui peut amuser certains d'entre vous, à savoir un système de redirection aléatoire en fonction des erreurs rencontrées. Certes, son rôle pour le référencement est limité mais moins en matière de communication ou d'intérêt technique car nous pouvons ainsi renvoyer les internautes vers d'autres pages web de contenus ou produits qui pourraient davantage les sensibiliser (ce même principe se retrouve avec des pages d'erreurs personnalisées afin de propager un message par exemple).

L'objectif est de proposer aux internautes ou aux moteurs de recherche des pages de destination variées dès qu'une erreur se produit. Toute la subtilité se situe au niveau de la gestion du « hasard ». De nombreuses autres méthodes peuvent s'appliquer, mais gardez en tête le principe si cela vous intéresse.

```
<?php
// Tableau contenant une liste d'URL
$URLS = array("http://www.site.fr", "http://www.site.fr/404.html",
"http://www.site.fr/contact.html", "http://www.site.fr/services.html");
// Gestion du hasard
$random = mt_rand(0, count($URLS)-1);
// Redirection aléatoire à l'aide de l'en-tête HTTP Location
header('Location: $URLS[$random]');
?>
```

Vous savez désormais comment mieux gérer les redirections et les erreurs courantes pour contrer les problèmes de SEO. Nous allons à présent optimiser la réécriture d'URL, ce qui peut s'avérer parfois très complexe et technique pour les plus débutants…

Maîtriser la réécriture d'URL

La réécriture d'URL (ou *URL rewriting*) constitue certainement l'étape la plus complexe à mettre en œuvre à l'aide des fichiers `.htaccess` pour un site web dynamique. Souvent, nous oublions ce point fondamental du référencement car nous sommes habitués à ce que des outils ou les CMS gèrent cette réécriture nativement. Cependant, il est important de bien connaître la technique qui se dissimule derrière afin de maîtriser pleinement nos URL optimisées.

Tout se dit autour des URL depuis de nombreuses années. D'un côté, nous savons que les mots-clés inclus dans les adresses web impactent quelque peu le positionnement mais aussi les aspects sensoriels de la page (mémorisation, compréhension...). D'un autre côté, il semblerait qu'à la sortie de Google Panda, les mots-clés des noms de domaines et URL n'étaient plus pris en compte comme l'indiquait les descriptifs de l'époque. En réalité, la sortie du filtre anti-EMD montre que ce point n'a jamais été pleinement négocié et que les URL ont encore un vrai rôle à jouer en matière d'indexation et de positionnement des pages.

La problématique de la réécriture d'URL existe donc depuis les origines des sites dynamiques. De nos jours, des CMS comme WordPress proposent un système avancé de réécriture qui nous fait oublier à quel point ce facteur était sensible quelques années auparavant. Cependant, nombre de frameworks ou CMS ne sont pas aussi poussés et ont une réécriture limitée voire unique des pages qui ne correspond pas toujours à nos attentes réelles.

La structure idéale est d'avoir une adresse vraiment unique pour une page, quelles que soient les catégories auxquelles elle est rattachée.

Par exemple, si nous créons un article intitulé « Techniques de référencement », nous voudrons idéalement obtenir une URL claire et composée de mots-clés telle que http://www.site.fr/techniques-de-referencement. Souvent, comme cet article est rattaché à une ou plusieurs catégories, nous nous retrouvons avec des adresses différentes comme http://www.site.fr/techniques-web/techniques-de-referencement ou encore http://www.site.fr/seo/techniques-de-referencement. Dans ce cas, nous sommes confrontés à un exemple flagrant de contenu dupliqué, deux pages « différentes » ayant le même article à proposer via deux URL différenciées.

Cet exemple est très fréquent dans les CMS courants du marché, ce qui explique le nombre incalculable de cas de duplicate content. Nous devons donc parfois retoucher la structure de la réécriture d'URL voire nous l'approprier complètement pour obtenir des résultats fiables.

L'unicité des réécritures proposées dans ces outils nous rend totalement dépendant des systèmes mis en place, qu'ils soient bons ou mauvais, et nous n'avons plus que nos yeux pour pleurer lorsque nous constatons les défauts inhérents des techniques imposées. Par conséquent, nous allons voir comment procéder pour nettoyer ou construire notre réécriture d'URL, et même si ce n'est pas une mince affaire, cela est nécessaire et mérite le détour.

Tout d'abord, retenons que la réécriture impose deux principes :

- la moindre erreur dans les fichiers `.htaccess` va créer un crash du serveur (blocage) et rendre le site totalement inaccessible ;

- l'ensemble des liens hypertextes présents dans la structure des pages doit être retravaillé pour correspondre aux nouveaux liens réécrits. En d'autres termes, toutes nos URL mal écrites et enregistrées dans nos pages (ou dans notre système dynamique pour être exact) vont devoir reprendre la structure des nouvelles adresses que nous souhaitons afficher. C'est souvent la partie la plus laborieuse, c'est pourquoi il faut y réfléchir dès le départ pour éviter tout problème d'affichage.

La réécriture d'URL agit sur la partie appelée *query string* dans les URL, ce qui correspond à la section qui contient tous les paramètres d'URL. Voici comment se décompose une URL afin de bien comprendre la partie sur laquelle nous allons agir :

```
protocole://nom-de-domaine/chemin/page.extension?query_string
```

Dans les sites dynamiques, nous générons des adresses web dynamiques qui prennent un ou plusieurs paramètres, comme dans les exemples suivants :

- cas de l'Ajax : https://www.google.fr/#q=seo&start=10 ;
- cas du PHP : http://www.site.fr/page.php?categorie=2&article= 27 ;
- cas d'une page avec l'extension de la technologie ASPX : http://www.site.fr/page.aspx?idCategorie=2&idArticle=27.

Avec ou sans nom de page ?

Il arrive que les URL ne contiennent pas les page.aspx ou page.php, par exemple, et enchaînent directement avec la query string après le slash.

Force est de constater que les URL sont peu lisibles nativement et peu mémorisables. Qui plus est, nous savons que les moteurs de recherche peuvent peiner à lire des URL à rallonge quand les paramètres s'enchaînent dans la query string. Si cela ne constitue pas un risque de pénalité en soi, cette accumulation d'options d'URL empêche souvent la bonne indexation des pages, ce qui rend la réécriture d'URL encore plus intéressante.

Enfin, il faut savoir que la sécurité des sites web entre aussi en ligne de compte. En effet, les paramètres d'URL contiennent souvent le titre de la page mais si ce dernier est composé d'espaces, nous générons des risques de mauvaises lectures selon les navigateurs mais aussi des failles dans lesquelles peuvent s'engouffrer des pirates du Web (bien que cela ne soit qu'une infime partie des failles accessibles en réalité). Pour contrer ce problème, nous utilisons souvent les fonctions PHP `url_encode()` et `url_decode()` qui permettent de remplacer les espaces et les caractères spéciaux par des codes hexadécimaux ASCII. Voici comment une URL classique peut se transformer une fois encodée :

- URL de base :

> http://www.site.fr/page.php?id=13&titre=le référencement est super !

- URL encodée :

> http://www.site.fr/page.php?id=13&titre=le%20referencement%20est%20super%20%21

Cette solution évite les problèmes mais n'est pas idéale en matière de SEO ou même de communication auprès des internautes. Nous ne pouvons pas considérer que ces adresses soient très lisibles donc il faut souvent ruser et créer une fonction de réécriture des URL pour éviter tout problème. L'idéal est de le faire au sein du code du site si cela n'existe pas afin d'envoyer dans la base de données des URL propres et finalisées, puis de procéder à la réécriture d'URL via les fichiers `.htaccess` pour rendre l'ensemble opérationnel.

Si nous résumons, les URL dynamiques contiennent plusieurs caractères à réécrire :

- les lettres accentuées doivent être remplacées par leur équivalent sans accent ;
- les espaces doivent être comblés, souvent par un tiret pour faciliter la lecture des robots ;
- les caractères spéciaux doivent être remplacés ou supprimés.

Lorsque nous réécrivons de manière dynamique, il arrive que l'adresse obtenue ne soit pas parfaite. C'est pourquoi des outils comme WordPress ou Joomla proposent de réécrire les alias d'URL, ce qui nous permet de proposer l'URL qui nous intéresse réellement. Quand ce champ est rempli, l'outil le prend en priorité, et si ce n'est pas le cas, il réécrit l'adresse avec la fonction par défaut. Ce système est simple à créer en PHP, en ASP ou en Python, par exemple, puisqu'il s'agit uniquement d'une condition `if (condition) { … } else { … }` en réalité.

Prenons l'exemple d'une URL récupérant un titre contenant une apostrophe et un point d'interrogation, soit http://site.fr/page.php?id=7&titre=l'idée est-elle géniale ?. Voilà à quoi elle devrait ressembler pour bien préparer le travail de réécriture :

```
http://site.fr/page.php?id=7&titre=lidee-est-elle-geniale
```

Nous remarquons que ce n'est pas une URL parfaite à cause de l'apostrophe supprimée. Dans ce cas, il vaudrait donc mieux proposer une alternative dans notre système pour personnaliser l'URL ou au moins pour nettoyer les cas comme « lidee » en supprimant le « l ».

La fonction suivante permet de nettoyer les URL rapidement avant l'ajout dans une base de données, par exemple :

```
function cleanURL($url = '') {
    // Nettoyage des accents
    $accents = 'ÀÁÂÃÄÅàáâãäåÒÓÔÕÖØòóôõöøÈÉÊËéèêëÇçÌÍÎÏìíîïÙÚÛÜùúûüÿÑñ';
    $noAccents = 'aaaaaaaaaaaaooooooooooooeeeeeeeeecciiiiiiiiuuuuuuuuynn';
    $cleanUrl = strtr($url, $accents, $noAccents);

    // Nettoyage des caractères spéciaux
    $cleanUrl = preg_replace('#([^a-zA-Z0-9-]+)#iU', '-', $cleanUrl);

    // Nettoyage des tirets en trop
    $cleanUrl = preg_replace('#([-]{2,50})#iU', '', $cleanUrl);

    // Nettoyage des cas d'apostrophes
    $cleanUrl = preg_replace('#^([a-zA-Z0-9-]+[-]+)#iU', '', $cleanUrl);
```

```
    $cleanUrl = preg_replace('#([-]+(qu|t|s|d|j|l|m|c|n)+[-]+)#iU', '-', $cleanUrl);

    // Nettoyage d'un tiret de début ou de fin
    if($cleanUrl[0] == '-') {
    $cleanUrl = substr($cleanUrl, 1);
    }
    if(substr($cleanUrl, -1, 1) == '-') {
    $cleanUrl = substr($cleanUrl, 0, -1);
    }

    return $cleanUrl;
}
```

Il suffit ensuite de lancer la fonction avant l'ajout dans la base de données ou même dans les liens hypertextes pour avoir toujours la même composition d'URL dans le site. Ainsi, une adresse qui porterait un titre comme « Qu'est-ce qu'une URL réussie ? » dans la query string deviendrait automatiquement « est-ce-une-url-reussie » une fois réécrite.

Une fois cette URL protégée et refondue, nous pouvons procéder à la réécriture des liens proprement si cela n'est pas déjà le cas, puis à la réécriture d'URL côté serveur avec les fichiers `.htaccess`.

Le principe de la réécriture d'URL dans les fichiers `.htaccess` consiste à respecter plusieurs étapes.

1. Ajouter la ligne `RewriteEngine On` (obligatoire) pour préciser au serveur que la réécriture d'URL est active. La valeur `off` désactive la réécriture.
2. Ajouter si besoin l'instruction `RewriteBase /` (optionnelle) pour indiquer l'URL d'origine qui sert de préfixe à toutes les adresses utilisées dans le fichier. Si vous entrez par exemple `RewriteBase /categorie/`, toutes les URL de la page commenceront automatiquement par le répertoire `categorie`.
3. Ajouter la règle `Options +FollowSymlinks` (optionnelle) afin d'indiquer au serveur qu'il doit suivre les liens symboliques réécrits dans le fichier `.htaccess`.
4. Écrire des règles de réécriture grâce à l'instruction `RewriteRule`. La structure définitive ressemble à la ligne suivante, les URL étant séparées par des espaces :

```
RewriteRule NOUVELLE-URL-REECRITE URL-A-REECRIRE [drapeau]
```

Autres options de réécriture

Il existe beaucoup d'autres instructions méconnues comme `RewriteOptions` pour accroître le nombre de redirections autorisées, `RewriteLog` pour gérer un journal d'erreurs ou encore `RewriteCond` pour gérer des conditions...

Toute la complexité de la réécriture d'URL se situe justement dans les règles composées d'expressions régulières (`regex`) parfois complexes et de paramètres dynamiques utiles pour réceptionner les informations importantes des adresses web. Qui plus est, nous devons à tout prix maîtriser l'usage du « drapeau » présent en fin de règle de réécriture. Par exemple, le drapeau `[L]` indique que la réécriture d'URL doit s'arrêter après l'application de la règle en cours, afin d'éviter une boucle infinie et d'éventuelles erreurs.

Héritage du dossier parent

Les fichiers `.htaccess` sont spécifiques au répertoire dans lequel ils s'appliquent. Il est possible de relancer des règles d'un niveau supérieur avec la règle `RewriteOptions Inherit`.

Il est parfois utile de rajouter des conditions avant des règles de réécriture sur le principe suivant :

```
RewriteCond %{VARIABLE_DE_TEST} condition_testee [drapeau]
```

Il existe de nombreuses variables de test, dont voici uniquement les plus courantes :

- variables d'en-têtes :
  - `HTTP_USER_AGENT` : indique le navigateur utilisé ;
  - `HTTP_REFERER` : précise l'adresse de la page web précédente (le « référent ») ;
  - `HTTP_COOKIE` : indique la chaîne de caractères cryptée qui contient les cookies ;
  - `HTTP_HOST` : indique le nom du serveur utilisé ;
- variables de serveur :
  - `DOCUMENT_ROOT` : indique le dossier racine du site ;
  - `SERVER_NAME` : récupère le nom du serveur ;
  - `SERVER_ADDR` : récupère l'adresse IP du serveur ;
  - `SERVER_PORT` : retourne le port du serveur ;
  - `SERVER_PROTOCOL` : indique le protocole utilisé ;
- variables inclassables :
  - `REQUEST_URI` : récupère l'URI complète qui correspond à la page visée ;
  - `THE_REQUEST` : retourne la requête HTTP complète qui contient par exemple le protocole en cours ainsi que la méthode utilisée ;
  - `REQUEST_FILENAME` : récupère le chemin local complet d'accès aux ressources ;
  - `HTTPS` : retourne `on` ou `off` en fonction de l'utilisation du protocole SSL ou non ;
- variables de connexion et de requête :
  - `REMOTE_ADDR` : récupère l'adresse IP du visiteur (comparable à `HTTP_FORWARDED` pour savoir si la personne utilise un proxy ou non) ;
  - `REMOTE_PORT` : récupère le port utilisé par le client ;
  - `REMOTE_USER` : renvoie un nom d'utilisateur envoyé par le client ;
  - `REQUEST_METHOD` : retourne la méthode utilisée (`GET`, `POST`…) ;
  - `QUERY_STRING` : récupère la query string complète.

Une fois la variable de test mise en place, il faut ajouter la condition, et elle peut prendre plusieurs formes :

- expression régulière classique ;

- comparaison avec les signes <, > ou = et différenciation avec le caractère !. Par exemple, la condition inverse `!index\.php` correspond à toutes les URL exceptées `index.php` ;
- `-d` : vérifie le chemin vers un répertoire et s'il existe ;
- `-f` : vérifie le chemin vers un fichier et s'il existe ;
- `-s` : vérifie le chemin vers un fichier dont la taille est non nulle, et contrôle s'il existe ;
- `-l` : vérifie le chemin vers un lien symbolique et s'il existe ;
- `-x` : vérifie le chemin si le client a l'autorisation de l'exécuter et s'il existe ;
- `-F` : vérifie si le fichier est valide ou non et accessible ;
- `-U` : vérifie si l'URL est valide et accessible.

Les deux conditions suivantes, que l'on peut retrouver par exemple dans le fichier `.htaccess` d'un site réalisé avec WordPress, signifient à la règle qui suit qu'elle ne doit pas faire de redirections automatiques vers le fichier `index.php` lorsqu'il s'agit d'un fichier ou d'un sous-répertoire réel :

```
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
```

Les réécritures sont techniques et demandent une vraie application pour être certain du bon fonctionnement final. Il n'est pas rare de tester des solutions et de connaître des échecs, notamment lorsque nous souhaitons procéder à des réécritures avancées.

Nous devons finir cette large introduction à la réécriture d'URL par l'utilisation des drapeaux avant d'expliquer ou de rappeler rapidement le principe des expressions régulières. La liste des drapeaux est relativement longue et mérite d'être détaillée car elle peut avoir une forte incidence sur nos règles de réécriture :

- `[B]` *(escape)* : force l'échappement des caractères spéciaux dans l'URL ;
- `[C]` *(chain)* : indique que la règle de réécriture est directement liée à la suivante ;
- `[F]` *(forbidden)* : impose au serveur de retourner une erreur 403 si la règle est respectée ;
- `[G]` *(gone)* : force le serveur à renvoyer l'erreur 410 si besoin ;
- `[H]` *(handler)* : impose au serveur de traiter les données avec le type spécifié (par exemple, la règle `RewriteRule !\. - [H=application/x-httpd-php]` indique que tous les fichiers sans extensions doivent être traités comme des fichiers PHP) ;
- `[L]` *(last)* : stoppe le processus après l'instruction en cours ;
- `[N]` *(next)* : relance de manière récursive l'instruction tant qu'elle est vraie ;
- `[NC]` *(nocase)* : ignore la casse dans la règle de réécriture ;
- `[NE]` *(noescape)* : empêche la conversion ASCII des caractères spéciaux ;
- `[P]` *(proxy)* : force le serveur à traiter la requête via un proxy ;
- `[QSA]` *(qsappend)* : permet au serveur de combiner les options d'URL plutôt que de les supprimer lorsqu'elles s'ajoutent aux paramètres de l'instruction (très utile si vous avez des arguments optionnels qui ne seront pas réécrits) ;

- `[R]` *(redirect)* : indique une redirection 302 par défaut ou un type spécifique si nous le précisons comme `[R=301]` pour une redirection permanente ;
- `[S]` *(skip)* : permet de sauter un nombre d'instructions si nécessaire (par exemple, `[S=2]` évite le lancement des deux instructions suivantes si la règle active est vérifiée) ;
- `[T]` *(type)* : applique la règle uniquement au type MIME précisé (exemples : `[T=image/png]` ou `[T=text/html]`) ;
- `[OR]` *(or)* : applique la règle en cours ou la suivante au lieu des deux comme c'est le cas par défaut.

Combiner des drapeaux

Il est possible de combiner plusieurs drapeaux en les séparant par des virgules, c'est notamment souvent le cas avec des exemples tels que `[P, L]`, `[NC, R]` ou `[QSA, L]`.

Maintenant, avançons dans notre initiation à la réécriture d'URL et intéressons-nous à l'assemblage des expressions régulières pour effectuer les redirections. En effet, nous allons devoir expliquer au serveur quels types d'URL nous souhaitons réécrire grâce à des caractères spéciaux et des instructions définies (que nous appelons *pattern* ou « motif »).

Par exemple, le pattern `\w` signifie que nous acceptons toutes les lettres, tous les chiffres ainsi que l'underscore. Nous pouvons écrire l'équivalent de manière plus lisible et plus mémorisable sous la forme `[a-zA-Z0-9_]`. Sans rentrer dans le détail des expressions POSIX ou PCRE, sachez qu'il existe des écritures qui doivent prendre un délimiteur pour être fonctionnelles mais cela n'est pas le cas dans les fichiers `.htaccess`.

Retenons que nos règles de réécriture vont devoir être composées de plusieurs facteurs :

- des caractères de début (`^`) et de fin de ligne (`$`) ;
- un point (`.`) pour indiquer que tous les caractères sont tolérés ;
- des ensembles de caractères tolérés compris dans un bloc `[caractères…]` :
  - `[…]` correspond aux caractères acceptés ;
  - `[^…]` exclut tous les caractères indiqués ;
- des groupes de données compris entre des parenthèses `(…)` ;
- des répétiteurs :
  - `?` placé après un motif signifie qu'il doit exister une fois ou non ;
  - `*` placé après un pattern indique qu'il doit exister une ou plusieurs fois ;
  - `+` placé derrière un motif précise qu'il doit exister au moins une fois ;
  - `{n, n}` indique que le motif doit être respecté un nombre défini de fois (par exemple, `{1,2}` correspond à une ou deux fois, `{3}` pour de zéro à trois fois ou encore `{3,}` pour un minimum de trois fois) ;
  - `|` placé dans un motif indique un choix entre des règles (équivalent de « ou » en quelque sorte).

Toute la stratégie de réécriture se situe dans la gestion des groupes de motifs et dans l'écriture des motifs eux-mêmes. La liste suivante présente des variantes de caractères avec des équivalences afin de pouvoir déterminer plus précisément nos règles de réécriture :

- `[a-zA-Z0-9]` ou `[:alnum:]` : ensemble des caractères alphanumériques, quelle que soit la casse des lettres (écrire seulement « a-z » ou « A-Z » pour gérer la casse) ;
- `[a-zA-Z0-9_]`, `\w` ou `[:word:]` : ensemble des caractères alphanumériques avec l'underscore en plus. L'inverse s'écrit `\W` ou `[^a-zA-Z0-9_]` ;
- `[a-zA-Z]`, `\a` ou `[:alpha:]` : ensemble des caractères alphabétiques (hors accents selon l'encodage) ;
- `[ \t]`, `\s` ou `[:blank:]` : caractères « espace » et « tabulation » ;
- `[ \t\r\n\v\f]`, `\s` ou `[:space:]` : ensemble des caractères « blancs ». L'inverse s'écrit `\S` ou `[^ \t\r\n\v\f]` ;
- `[0-9]`, `\d` ou `[:digit:]` : caractères digitaux. L'inverse s'écrit `\D` ou `[^0-9]` ;
- `[][!"#$%&'()*+,./:;<=>?@\^_`{|}~-]` ou `[:punct:]` : ensemble des caractères de ponctuation.

Obsolescence des POSIX

Les caractères POSIX sous la forme `[: ... :]` sont obsolètes aujourd'hui mais vous pourrez les rencontrer, ce qui explique pourquoi ils ont été indiqués.

Une fois que nous maîtrisons bien les motifs, il suffit de les insérer dans des groupes et d'ajouter les caractères de répétition utiles pour obtenir le résultat escompté. Voici quelques exemples de groupes complets de caractères :

- `([a-zA-Z0-9-_]+)` indique que nous souhaitons une URL qui contient au moins une fois un caractère alphanumérique, un tiret ou un underscore ;
- `(asp|php|html|htm|aspx|py)` précise que la chaîne de l'URL doit contenir une des extensions placées entre les parenthèses ;
- `([a-zA-Z]+[-][0-9]+)` indique que l'URL doit être de la forme « lettres-chiffres » (parfois utile pour caler un identifiant chiffré dans l'adresse, par exemple) ;
- `(.*)` signifie que nous acceptons toutes sortes de caractères, de zéro à plusieurs fois.

Maintenant, il ne reste plus qu'à composer nos réécritures pour qu'elles ressemblent à ce que nous désirons. Commençons par une règle simple avec une URL à réécrire du type fiche.php?id=23 en fiche-23.html :

```
RewriteEngine On
Options +FollowSymlinks
RewriteBase /
RewriteRule ^fiche-([0-9]+)\.html$ fiche.php?id=$1 [L]
```

Échappement des caractères des expressions régulières

Nous écrivons `\.` car il faut échapper le point afin de ne pas être confondu avec le caractère universel (le point signifiant que la totalité des caractères est tolérée).

Nous pouvons aller plus loin en essayant de récupérer le titre de la fiche produit en ajoutant un motif supplémentaire pour avoir une URL plus précise comme titreFiche-ID.html. Le serveur Apache identifie les mots-clés correspondant à l'ID en question automatiquement dans la règle suivante :

```
RewriteRule ^([a-z0-9_-]*)-([0-9]+)\.html$ fiche.php?id=$2 [L]
```

Sachez qu'il est également possible de supprimer les ID (ou autre paramètre) dans la réécriture d'URL, mais en réalité cela n'est pas directement géré par le côté serveur avec PHP par exemple. Si vous utilisez des URL dynamiques, il faudra absolument fournir un argument fixe et unique à Apache afin qu'il puisse différencier les URL puisqu'il ne pourra pas deviner l'ID correspondant à chaque page. Souvent, nous utilisons un alias d'URL *(slug)* pour procéder comme tel.

Par exemple, admettons que nous voulions une URL telle que http://www.site.fr/titre-du-produit, le paramètre `id` caché était nécessaire à Apache pour savoir de quelle page il s'agissait. Dorénavant, nous devrons utiliser un alias d'URL basé sur le titre (ici, titre-du-produit) qui est unique par page. Ainsi, nos URL seraient plutôt de la forme ...?titre=titre-du-produit au sein du code PHP et dans la base de données, et notre réécriture utiliserait donc le paramètre `titre` plutôt que `id`. Cacher les arguments est souvent complexe mais le rendu est plus agréable pour les visiteurs et efficace pour les moteurs de recherche. La règle de réécriture deviendrait alors :

```
RewriteRule ^([a-z0-9_-]*)$ fiche.php?titre=$1 [L]
```

L'inconvénient de cette technique est qu'elle doit être parfaitement maîtrisée pour fonctionner. En effet, si le titre change, l'URL devient invalide et nous dirigeons donc vers une adresse erronée. Pour contrecarrer ce risque, l'idéal est de créer un champ « alias » ou « slug » dans la base de données du site qui sera indépendant du titre réel. Ainsi, nous pourrons très bien modifier le titre sans changer le slug et sans causer ce problème de page perdue.

Nous avons ici abordé des cas simples et courants, mais il arrive parfois d'être confronté à des réécritures plus complexes. Nous allons montrer un dernier exemple afin de prendre la mesure de ce qu'il est possible de faire avec des conditions et des règles strictes. Pour le reste, chacun d'entre nous aura des cas particuliers à régler, il est impossible de présenter ici la quantité incommensurable de variantes existantes.

Prenons l'exemple d'un site multilingue conçu avec plusieurs sous-domaines. Souvent, nous ajoutons un paramètre dans l'adresse du type `lang` pour déterminer quelle est la langue choisie. Mais dans notre exemple, il faut créer une règle de réécriture pour passer du domaine principal vers le sous-domaine de la langue choisie. Il nous faut également le slug ou l'ID de la page en paramètre pour pouvoir donner un nom définitif à la page traduite. Ainsi, une URL comme http://www.site.com/index.php?lang=en&slug=notre-titre deviendra http://en.site.com/notre-titre.html, ce qui est bien plus efficace en matière de SEO.

Il faut ajouter une condition pour récupérer la langue du sous-domaine ciblé, laquelle sera récupérée dynamiquement dans le fichier `.htaccess` par une variable `%chiffre`. Ensuite, nous appliquons la réécriture pour rediriger les paramètres vers le sous-domaine visé.

```
# S'il s'agit d'un sous-domaine valide…
RewriteCond %{HTTP_HOST} ^(fr|en)\.site\.com$ [NC]
```

```
# … qui réécrit l'adresse proprement
RewriteRule ^([a-zA-Z0-9-_]+)\.html /index.php?lang=%1&page=$1 [NC, L]
```

Il existe une multitude de possibilités, toutes sont dépendantes de la structure du site et du langage dynamique utilisé mais aussi du rendu final désiré. Il est toutefois important de faire très attention aux cas particuliers et aux pages dupliquées à cause d'une mauvaise réécriture ou redirection, par exemple. Généralement, il faut effectuer des tests parfois laborieux pour obtenir les résultats escomptés, cela fait partie du jeu en quelque sorte…

Autres astuces avec les fichiers .htaccess

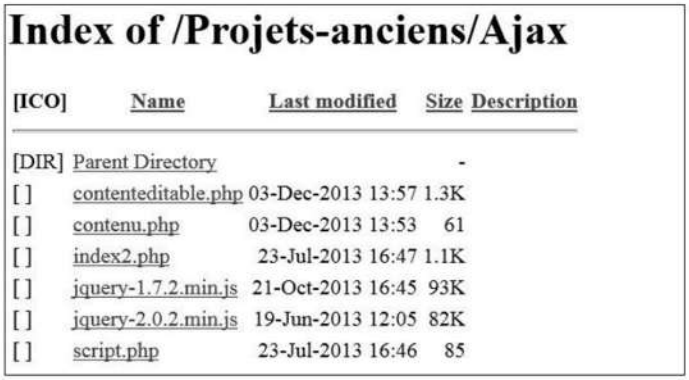
Les fichiers `.htaccess` regorgent d'autres fonctionnalités qui peuvent nous être utiles, en voici quelques-unes qui pourront peut-être nous dépanner par moment.

- Ajout en direct de l'encodage des caractères avec `AddDefaultCharset utf-8`. Cette ligne permet de forcer l'usage de l'encodage UTF-8 et n'impose pas l'utilisation des balises meta pour spécifier le jeu de caractères utilisés (mais il est conseillé d'ajouter les deux). Cette ligne peut vous sauver la vie quand PageSpeed Insights n'arrive pas à comprendre le `charset` pour lequel vous avez opté, même quand il est parfois précisé…
- Utilisation de la directive `DirectoryIndex` pour indiquer le nom de la page d'accueil d'un site si vous préférez cette option à celle du simple nom de domaine. Dans ce cas, écrivez la ligne : `DirectoryIndex page-accueil.html`.
- Blocage de l'accès aux répertoires qui n'ont pas de fichiers index. En effet, il arrive souvent sous Apache que les répertoires dévoilent la totalité des fichiers contenus s'il n'existe pas de page d'accueil, il faut donc les protéger avec l'instruction `Options All -Indexes`.
- Suppression de l'extension des pages pour masquer le langage utilisé. Il s'agit d'une règle parfois anodine mais qui permet de d'éviter que des personnes mal intentionnées puissent savoir avec quel langage tester un piratage. Qui plus est, il existe un mythe éternel en référencement qui prétend que l'extension du fichier peut jouer un rôle sur le positionnement car les moteurs auraient des préférences. Sur le principe, l'idée n'est pas totalement fausse puisque nous savons que les fichiers Flash .swf sont dépréciés par rapport à des fichiers .html. Mais dans les faits, il ne faut pas comparer des fichiers multimédias et des pages web au sens propre. Et sur ce point, force est de constater qu'une page PHP, ASP ou HTML bénéficient de la même considération par les moteurs. Si toutefois vous doutez encore, cachez vos extensions pour protéger votre site et limiter la distinction des langages :

```
# Autorise la lecture des URL sans extension (optionnel)
Options +MultiViews
# Règles de réécriture (une des possibilités)
RewriteCond %{SCRIPT_FILENAME} !-f
RewriteCond %{REQUEST_URI} ^/(.*).(htm|html|php|asp|aspx|py) [NC]
RewriteRule ^(.*)$ $1.%2 [L]
```

Figure 2-17

Exemple de répertoire sans page index visible par les utilisateurs mal intentionnés



Index of /Projets-anciens/Ajax

[ICO]	Name	Last modified	Size	Description
[DIR]	Parent Directory		-	
[ ]	contenteditable.php	03-Dec-2013 13:57	1.3K	
[ ]	contenu.php	03-Dec-2013 13:53	61	
[ ]	index2.php	23-Jul-2013 16:47	1.1K	
[ ]	jquery-1.7.2.min.js	21-Oct-2013 16:45	93K	
[ ]	jquery-2.0.2.min.js	19-Jun-2013 12:05	82K	
[ ]	script.php	23-Jul-2013 16:46	85	

- Blocage des robots de spam ou des aspirateurs de site avec une règle restrictive :

```
RewriteEngine On
# Exemple avec un nom précis
RewriteCond %{HTTP_USER_AGENT} ^nom-robot [NC,OR]
# Exemple avec une chaîne de caractères contenue dans le nom du robot
RewriteCond %{HTTP_USER_AGENT} .*nom-robot* [NC]
RewriteRule .* - [F]
```

- Protection de la bande passante et des images *(hotlinking)* en bloquant l'accès aux liens directs aux sites web externes (en retournant une erreur 403) avec le code suivant :

```
RewriteEngine On
# Ajout d'exceptions pour nos domaines et sous-domaines
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^https?://site.fr/(.*)$ [NC]
RewriteCond %{HTTP_REFERER} !^https?://([a-zA-Z0-9-_]+).site.fr/(.*)$ [NC]
ReWriteRule .*\.(gif|png|jpe?g)$ - [NC, F]
```

- Bannissement d'adresses IP pour protéger le site :

```
Allow from all
# Ajouter autant de règles " Deny from IP " que besoin
Deny from 192.168.1.10
```

Après ce long périple pour maîtriser les fichiers `.htaccess`, sachez qu'il est important de préciser qu'il peut vraiment réduire les performances du serveur s'il est trop complexe ou qu'il génère trop de traitements. Il s'agit donc d'un fichier qui peut vraiment nous aider si nous l'optimisons mais qui peut aussi générer un effet pervers de perte de bande passante. Qui plus est, certains hébergements mutualisés ne proposent pas l'ensemble des fonctionnalités, ce qui explique parfois le mauvais fonctionnement de certains codes, au grand dam des référenceurs…

ASP, ASP.Net et configuration des serveurs IIS de Microsoft

La majorité des référenceurs se focalisent sur les serveurs Apache très déployés dans le monde des hébergements mutualisés, mais dès que nous nous intéressons de près à la question des serveurs dédiés, nous remarquons que Microsoft n'est pas en reste et répond à une demande accrue de la part des entreprises. Par conséquent, tout ce que nous venons d'étudier à propos des fichiers `.htaccess` n'a plus lieu d'être car ils n'existent pas sur les serveurs IIS de Microsoft.

Chez Microsoft, la configuration est différente puisque au moins trois fichiers sont utilisés par le système : `ApplicationHost.config`, `Machine.config`, et `web.config`. Nous insisterons sur le dernier d'entre eux pour parfaire la réécriture d'URL pour Microsoft. Ainsi, nous pourrons aussi bien nous dépatouiller en référencement sur les deux serveurs, bien qu'il faudra parfois retranscrire certaines fonctions du livre avec les technologies ASP ou ASP.Net.

Le fichier `web.config` contient un code balisé sémantiquement à la manière d'un fichier XML. Nous verrons comment optimiser ces balisages pour effectuer des actions similaires à celles disponibles sur les serveurs Apache.

Faire des tests avec un serveur IIS installé localement

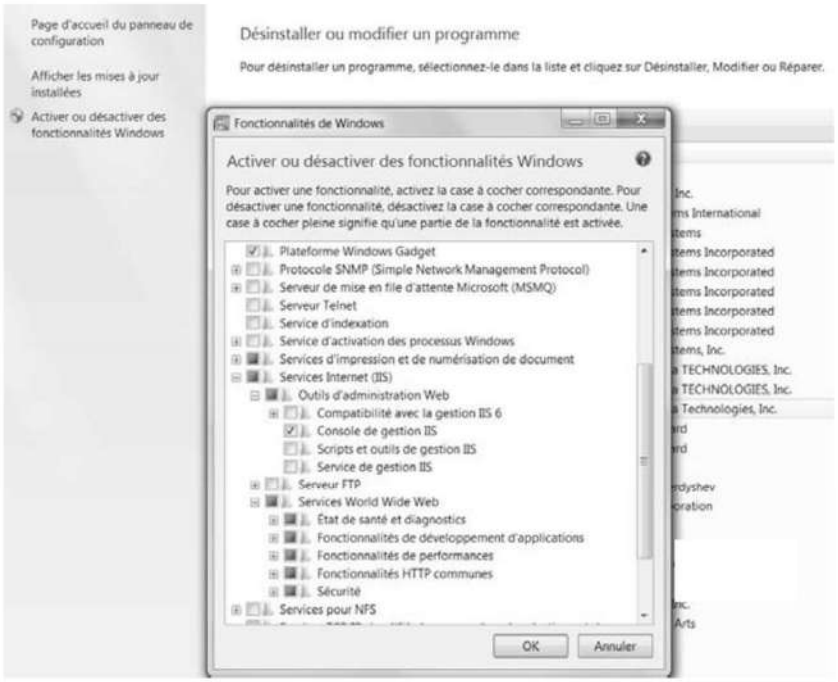
Habitués des serveurs Apache, nous avons tendance à utiliser des serveurs locaux pour installer nos sites web avant de les mettre en ligne sur la Toile. Généralement, nous optons pour des logiciels tels que WampServer ou EasyPHP pour répondre à nos besoins mais chez Microsoft, ces outils ne sont pas parfaitement adaptés et ne permettent pas de travailler efficacement.

Fort heureusement, Microsoft a pensé à tout et intègre depuis des années les services IIS au sein de ces versions de Windows. Il suffit de faire quelques recherches dans l'aide de Windows pour trouver comment installer un serveur IIS local dans la machine puisqu'il est déjà implanté mais désactivé par défaut. Ceci est possible depuis Windows NT donc nous pouvons être rassurés et le retrouver dans la plupart de nos machines sans aucun soucis.

Par exemple, sur Windows Vista et Windows 7, il faut se rendre dans le menu *Panneau de configuration*, cliquer sur *Désinstaller un programme*, puis *Activer ou désactiver des fonctionnalités Windows*. Ensuite, il suffit de cocher et d'activer la section *Services Internet (IIS)* dans la liste des services disponibles. Le procédé est le même sur Windows 8, sauf que la section s'intitule *Internet Informations Services (IIS)*.

Figure 2-18

Installation d'un serveur IIS local sur Windows 7



Une fois le serveur installé, il suffit de se rendre dans les options d'administration du *Panneau de configuration* et de cliquer sur *Gestionnaire des services Internet (IIS)* pour administrer le serveur local. En cliquant sur *Sites*, puis en sélectionnant un site, il est possible de modifier le port du serveur local avec l'option *Liaisons…* située dans la colonne de droite, mais aussi le chemin d'accès aux fichiers (répertoire dans lequel les sites sont installés) dans les paramètres avancés.

Figure 2-19

Paramétrage et utilisation d'un serveur IIS local sur Windows 7

Enfin, nous pouvons accéder aux sites du serveur en tapant http://localhost/ dans la barre d'adresse si vous n'utilisez aucun autre système local. Si WampServer ou EasyPHP sont installés et activés, saisissez également le nom du port pour éviter les conflits, par exemple http://localhost:8080 (ou autre numéro de port).

Effectuer des redirections avec IIS, ASP et ASP.Net

Nous avons vu que la configuration Apache et les redirections PHP étaient relativement simples à mettre en place. Qu'on se le dise, cela reste possible chez Microsoft et n'est pas beaucoup plus compliqué dans les faits. Par exemple, les redirections 301 peuvent s'effectuer simplement avec ASP comme le montre le code suivant :

```
<%@ Language=VBScript %>
<%
Dim NewURL as String
NewURL = 'http://www.nouveau-site.fr'
Response.Status = "301 Moved Permanently"
Response.AddHeader = 'Location', NewURL
%>
```

En ASP.Net (fichiers .aspx), l'écriture varie quelque peu :

```
<script runat="server">
private void Page_Load(object sender, System.EventArgs e) {
    Response.Status = "301 Moved Permanently";
    Response.AddHeader("Location", "http://www.nouveau-site.fr");
}
</script>
```

Le type de redirection ne varie qu'en fonction du statut `Response.status` intégré dans les codes mais comme Google et consorts nous conseillent vivement l'usage des redirections permanentes (301), nous utiliserons ces scripts en règle générale.

Les redirections peuvent également être effectuées via le fichier `web.config` à l'aide d'un code simple. Il faut utiliser la balise `<location>` pour indiquer l'URL à rediriger ainsi que la balise `<httpRedirect>` pour indiquer le chemin de destination, comme ici :

```
<configuration>
<location path="page-a-rediriger.html">
<system.webServer>
<httpRedirect enabled="true" destination="http://www.site.fr/page-redirigee.html"
httpResponseStatus="Permanent" />
</system.webServer>
</location>
</configuration>
```

Équivalences des codes d'erreurs

L'attribut `httpResponseStatus` permet d'indiquer la valeur `Permanent` afin de procéder à une redirection 301, les valeurs `Found` et `Temporary` correspondant aux codes 302 et 307. Il suffit ensuite d'ajouter autant de blocs `<location>…</location>` que nécessaire pour effectuer les redirections utiles.

Nettoyer les URL avec VBScript

Nous avons vu en PHP comment nettoyer les URL de tous les caractères spéciaux et espaces qui peuvent poser des problèmes d'interprétation dans les navigateurs. Apprenons à faire la même manipulation avec ASP, par exemple, pour pouvoir ajouter et utiliser des adresses web propres dans les bases de données. Nous allons donc reprendre la fonction `cleanURL()` que nous avions créée dans un fichier .asp en la réadaptant en format VBScript pour ASP.

Malheureusement, le VBScript ne dispose pas des mêmes fonctions que PHP. Il faut donc créer deux fonctions, `strtr()` et `preg_replace()`, pour obtenir les équivalents des fonctions PHP. Voici les codes de ces deux fonctions très utiles traduites en VBScript :

```
<%
Function strtr(chaine, strFrom, strTo)
Dim c0, c1, i
for i = 1 to len(strFrom)
    c0 = mid(strFrom, i, 1)
    if i > len(strTo) Then
        c1 = ""
    else
        c1 = mid(strTo, i, 1)
    end if
    chaine = Replace(chaine, c0, c1)
next
strtr = chaine
End Function

Function preg_replace(regexp, chars, str)
    Set regex = New RegExp
    With regex
        .Pattern = regexp
        .IgnoreCase = True
        .Global = True
    End With
    preg_replace = regex.Replace(str, chars)
    Set regex = nothing
End Function
%>
```

Désormais, nous pouvons reprendre notre fonction `clearURL()` comme en PHP en modifiant uniquement la syntaxe pour l'adapter au langage VBScript pour ASP. Il faudrait bien sûr reprendre la même procédure pour que le code soit fonctionnel avec ASP.Net.

```
<%
Function cleanURL(url)
Dim accents, noAccents

' Nettoyage des accents
accents = "ÀÁÂÃÄÅàáâãäåÒÓÔÕÖØòóôõöøÈÉÊËèéêëÇçÌÍÎÏìíîïÙÚÛÜùúûüÿÑñ"
```

```
noAccents = "aaaaaaaaaaaaaooooooooooooooeeeeeeeeeccciiiiiiiiuuuuuuuuuynn"
cleanerUrl = strtr(url, accents, noAccents)

' Nettoyage des caractères spéciaux
cleanerUrl = preg_replace("([^a-zA-Z0-9-]+)", "-", cleanerUrl)

' Nettoyage des tirets en trop
cleanerUrl = preg_replace("([-]{2,})", "", cleanerUrl)

' Nettoyage des cas d'apostrophes
cleanerUrl = preg_replace("#^([a-zA-Z0-9-]+[-]+)#iU", "", cleanerUrl)
cleanerUrl = preg_replace("#([-]+(qu|t|s|d|j|l|m|c|n)+[-]+)#iU", "-", cleanerUrl)

' Nettoyage d'un tiret de début ou de fin
If Left(cleanerUrl, 1) = "-" Then
    cleanerUrl = Replace(cleanerUrl,"-","",1,1)
End If
If Right(cleanerUrl, 1) = "-" Then
    cleanerUrl = Replace(Right(cleanerUrl, 1),"-","",1,1)
End If

' On retourne le résultat
cleanUrl = cleanerUrl
End Function
%>
```

Pour utiliser la fonction, il suffit ensuite de recourir à la commande `Response.write(cleanURL(url))` ou le code balisé et explicite `<%= cleanUrl(url) %>`, `url` étant une variable contenant l'adresse à nettoyer.

Ainsi, si vous avez une page web dont la query string reprend un titre comme « Qu'est-ce que le référencement ? », l'adresse sera réécrite proprement en « qu-est-ce-que-le-referencement ». Nous pourrons ensuite travailler bien plus proprement pour obtenir des URL optimisées.

Réécrire des URL avec un serveur Microsoft

La réécriture d'URL, tout comme la redirection, peut s'effectuer directement dans les options de configuration des serveurs IIS mais aussi via les fichiers `web.config` placés à la racine des répertoires ciblés. La technique n'est pas très compliquée et reprend le principe des balisages. Il convient uniquement de maîtriser ces blocs balisés et les expressions régulières pour faire fonctionner le système. Voici la liste des blocs à connaître :

- `<configuration>…</configuration>` englobe toutes les options du fichier `web.config` ;
- `<system.webServer>…</system.webServer>` contient les règles de réécriture ou les redirections, par exemple (disponible par défaut depuis IIS 7) ;
- `<rules>…</rules>` sont des blocs généraux comprenant les règles de réécriture d'URL ;

- chaque bloc `<rule>...</rule>` comporte une règle de réécriture définie. Il englobe un ou plusieurs types de balises :
  - `<match />` pour définir l'expression régulière de la réécriture ;
  - `<action />` pour réaliser la redirection vers la page réécrite ;
  - un bloc `<condition>...</condition>` (optionnel) pour définir des conditions à l'aide des balises `<add />`.

Chaque bloc `<rule>...</rule>` contient des informations obligatoires : la balise `<match />` reçoit l'expression régulière qui compose l'URL réécrite à la fin du traitement tandis que l'élément `<action />` contient l'URL d'origine à réécrire dans laquelle il faut ajouter l'attribut `type="rewrite"` pour activer la réécriture.

Il est possible d'ajouter des conditions si nécessaire pour s'assurer, par exemple, qu'il s'agit d'un fichier ou d'un répertoire existant comme nous pouvons le faire avec les serveurs Apache. Voici un exemple concret de réécriture d'URL avec IIS. La technique n'est en réalité pas plus complexe que celle utilisant les fichiers `.htaccess`. Il suffit de réaliser de bonnes expressions régulières et de respecter la sémantique des fichiers `web.config`.

```
<configuration>
<!-- Autre code placé au-dessus si besoin -->
<system.webServer>
<rewrite>
    <rules>
        <rule name="reecriture de categories">
        <!-- URL réécrite -->
        <match url="^categorie-([0-9]+)/([a-zA-Z0-9-_]+)" />
        <!-- Ajout de conditions (optionnel) -->
        <conditions logicalGrouping="MatchAny">
            <!-- Il doit s'agir d'un fichier valide ! -->
            <add input="{REQUEST_FILENAME}" matchType="IsFile" ignoreCase="true" />
            <!-- Il doit s'agir d'un répertoire valide ! -->
            <add input="{REQUEST_FILENAME}" matchType="IsDirectory" ignoreCase="true" />
        </conditions>

        <!-- URL à réécrire -->
        <action type="Rewrite" url="categorie.aspx?id={R:1}&titre={R:2}" />
        </rule>
    </rules>
</rewrite>
<!-- Autre code placé en-dessous si besoin -->
</system.webServer>
</configuration>
```

En réalité, tout la complexité se situe dans la bonne gestion des références indiquées à l'aide des écritures `{C:N}` et `{R:N}` (où `N` est un nombre de 0 à 9). Les blocs contenant `C` correspondent au numéro du pattern des conditions et les blocs contenant `R` correspondent aux règles classiques du pattern. Voici un exemple pour comprendre le principe avec le pattern `^(www\.)(.*)$` :

- {R:0} correspond à l'expression régulière complète, soit une URL comme www.site.fr ;
- {R:1} correspond au premier bloc entre parenthèses, soit www. ;
- {R:2} correspond au second bloc entre parenthèses, soit par exemple site.fr.

Il suffit donc de recomposer les URL avec les blocs de règles ou de conditions appropriés pour procéder à des réécritures propres.

Dans le même esprit, voici une réécriture d'URL pour rediriger un nom de domaine sans les www vers le nom de domaine qui possède le préfixe.

```
<configuration>
<system.webServer>
<!-- Autre code placé au-dessus si besoin -->
<rewrite>
    <rules>
        <rule name="ajout du www" stopProcessing="true">
            <match url=".*" />
            <conditions>
                <add input="{HTTP_HOST}" negate="true" pattern="^site.fr$" />
            </conditions>
            <action type="Redirect" url="http://www.site.fr/{R:0}"
            redirectType="Permanent" />
        </rule>
    </rules>
</rewrite>
<!-- Autre code placé en-dessous si besoin -->
</system.webServer>
</configuration>
```

Autres spécificités techniques du fichier web.config

Définir la page d'accueil par défaut

Comme avec les fichiers .htaccess des serveurs Apache, il est possible de définir les pages d'accueil d'un site avec le fichier web.config des serveurs IIS de Microsoft. Pour ce faire, il suffit d'ajouter les lignes suivantes et de modifier les balises <add /> à votre guise :

```
<system.webServer>
<!-- Autre code placé au-dessus si besoin -->
    <defaultDocument enabled="true">
    <files>
        <add value="index.asp" />
        <add value="index.aspx" />
        <add value="index.html" />
    </files>
    </defaultDocument>
<!-- Autre code placé en-dessous si besoin -->
</system.webServer>
```

Gérer les pages d'erreurs

La gestion des pages d'erreurs reste relativement simple à mettre en œuvre. Toute la documentation officielle de Microsoft à ce sujet est claire pour faciliter la configuration (source : http://www.iis.net). Dans les faits, il suffit de bien connaître la sémantique du code des fichiers `web.config` et d'ajouter les codes d'erreurs intéressants (attribut `statusCode`) avec la page de destination concernée (attribut `path`) dans des balises `<error />`. Il est également possible d'ajouter un préfixe à l'URL pour indiquer le chemin vers le serveur avec l'attribut `prefixLanguagePath` selon les cas.

```
<system.webServer>
<!-- Autre code placé au-dessus si besoin -->
<httpErrors>
    <error statusCode="401" prefixLanguageFilePath="%SystemDrive%\CHEMIN-SERVEUR"
    path="401.asp" />
    <error statusCode="403" prefixLanguageFilePath="="%SystemDrive%\CHEMIN-SERVEUR"
    path="403.asp" />
    <error statusCode="404" prefixLanguageFilePath="="%SystemDrive%\CHEMIN-SERVEUR"
    path="404.asp" />
    <error statusCode="500" prefixLanguageFilePath="="%SystemDrive%\CHEMIN-SERVEUR"
    path="500.htm" />
</httpErrors>
<!-- Autre code placé en-dessous si besoin -->
</system.webServer>
```

Optimiser le cache et le PageSpeed

Il est possible d'activer la compression Gzip ou Deflate avec IIS en paramétrant le bloc `<httpCompression>`. Vous pouvez définir la compression des fichiers statiques mais aussi de données dynamiques. Il existe deux types de balisages distincts pour procéder à la compression (`staticTypes` et `dynamicTypes`) dont le type doit être précisé dans une balise `<scheme />`, comme dans l'exemple suivant :

```
<system.webServer>
<!-- Autre code placé au-dessus si besoin -->
<httpCompression directory="%SystemDrive%\CHEMIN\IIS Temporary Compressed Files">
<scheme name="gzip" dll="%Windir%\system32\CHEMIN\gzip.dll" />

<dynamicTypes>
<add mimeType="text/*" enabled="true" />
<add mimeType="application/JavaScript" enabled="true" />
<add mimeType="*/*" enabled="false" />
</dynamicTypes>

<staticTypes>
<add mimeType="text/*" enabled="true" />
<add mimeType="application/JavaScript" enabled="true" />
<add mimeType="*/*" enabled="false" />
</staticTypes>
</httpCompression>
<!-- Autre code placé en-dessous si besoin -->
</system.webServer>
```

Au-delà de la compression, il est possible d'indiquer des extensions spécifiques en cache à l'aide du bloc `<caching>…</caching>`. Il suffit d'ajouter un bloc `<profiles>…</profiles>` contenant les extensions ciblées dans des balises `<add />` pour placer des documents en cache.

```
<system.webServer>
<!-- Autre code placé au-dessus si besoin -->
<caching enabled="true" enableKernelCache="true">
    <profiles>
        <add extension=".asp" policy="CacheUntilChange"
        kernelCachePolicy="CacheUntilChange" />
        <add extension=".aspx" policy="CacheUntilChange"
        kernelCachePolicy="CacheUntilChange" />
    </profiles>
</caching>
<!-- Autre code placé en-dessous si besoin -->
</system.webServer>
```

La compression peut aussi être effectuée au niveau des URL avec le code suivant :

```
<system.webServer>
    <urlCompression doStaticCompression="true" doDynamicCompression="true" />
</system.webServer>
```

De manière générale, il est également possible de fixer un seuil de cache à mettre en œuvre pour le serveur, au-delà même d'une sélection d'extension en particulier. Le code en est simplifié et s'appuie alors sur les attributs `maxCacheSize` et `maxResponseSize` pour définir respectivement la mémoire de sortie maximale du cache et la taille maximale de la réponse mise en cache. Attention, ce système ne fonctionne bien que dans le fichier `ApplicationHost.config` et non dans le fichier `web.config`.

```
<system.webServer>
    <!-- 1 Go de mémoire allouée au cache pour une réponse maximale de 1 Mo -->
    <caching enabled="true" enableKernelCache="true" maxCacheSize="1000"
    maxResponseSize="1024000"/>
</system.webServer>
```

Enfin, sachez également que vous pouvez paramétrer le cache `Expires` des fichiers avec `web.config`. Il existe plusieurs variantes qui ne trouvent pas réellement de parallèles avec ce qui se fait sur les serveurs Apache, mais les résultats peuvent toutefois être au rendez-vous.

Il est possible de définir à la fois la date d'expiration du cache de manière fixe ou tout simplement de donner un intervalle de temps avant l'expiration du cache. Les deux méthodes sont présentées dans l'exemple qui suit, elles se différencient par l'usage de la valeur `UseMaxAge` ou `UseExpires` dans l'attribut `cacheControlMode` des balises `<clientCache />` :

```
<system.webServer>
<!-- Autre code placé au-dessus si besoin -->
<staticContent>
```

```
    <!-- Cache fixé à une journée avant l'expiration -->
    <clientCache cacheControlMode="UseMaxAge" cacheControlMaxAge="1.00:00:00" />
</staticContent>

<staticContent>
    <!-- Expiration du cache le soir du 28 septembre 2014 -->
    <clientCache cacheControlMode="UseExpires" httpExpires="Sun, 28 Sep 2014 23:59:59 UTC" />
</staticContent>
<!-- Autre code placé en dessous si besoin -->
</system.webServer>
```

Tous ces exemples ne sont qu'une introduction à ce qu'il est possible de réaliser sur un serveur IIS. Il existe plusieurs variantes puisque nous pouvons aussi coder des fonctionnalités similaires en VBScript, en VB.Net ou C#, par exemple, plutôt que de passer par les fichiers de configuration du serveur.

Néanmoins, plusieurs de ces codes permettent d'améliorer considérablement la gestion des serveurs de Microsoft, trop souvent oubliés par les référenceurs. La documentation officielle de Microsoft ou des lectures connexes vous permettront de trouver des réponses appropriées en cas de besoin. N'hésitez pas si vous êtes de fervents utilisateurs des serveurs IIS…

Bloquer l'accès aux robots et aux IP sur IIS

Comme sur Apache, il est possible de bloquer l'accès d'un site à certaines adresses IP pour éviter les robots de spam ou même des utilisateurs. Pour cela, il suffit d'ajouter la liste des adresses à bloquer dans le fichier `web.config`, à l'aide des balises `<security>` et `<add>` notamment, comme dans les cas suivants :

```
<security>
<!-- On autorise toutes les adresses avec la ligne suivante -->
<ipSecurity allowUnlisted="true">
    <!-- On ajoute des restrictions avec le clear -->
    <clear/>
    <!-- Blocage d'une IP précise -->
    <add ipAddress="xxx.xxx.xxx.xxx"/>
    <!-- Blocage d'une plage d'adresses IP : de xxx.xxx.xxx.0 à xxx.xxx.xxx.255 -->
    <add ipAddress="xxx.xxx.xxx.xxx" subnetMask="255.255.255.0"/>
    <!-- Blocage d'une plage d'adresses IP : de xxx.xxx.0.0 à xxx.xxx.255.255 -->
    <add ipAddress="xxx.xxx.xxx.xxx" subnetMask="255.255.0.0"/>
    <!-- Blocage d'une plage d'adresses IP : de xxx.0.0.0 à xxx.255.255.255 -->
    <add ipAddress="xxx.xxx.xxx.xxx" subnetMask="255.0.0.0"/>
</ipSecurity>
</security>
```

Autre méthode de blocage des adresses IP

Pour bloquer par défaut toutes les IP et en autoriser seulement certaines, il faut passer l'attribut `allowUnlisted` sur `false` puis ajouter `allowed="true"` dans les balises `<add>` pour les adresses autorisées.

Compatibilité mobile

Pourquoi posséder un site mobile-friendly ?

Impact des supports mobiles

Les supports mobiles regroupent à la fois les téléphones portables, les smartphones et les tablettes. Les mini-PC ou équivalents sont exclus de cette liste en général. L'usage de ces objets nomades s'est accru exponentiellement depuis de nombreuses années et l'essor ne semble pas près de s'arrêter.

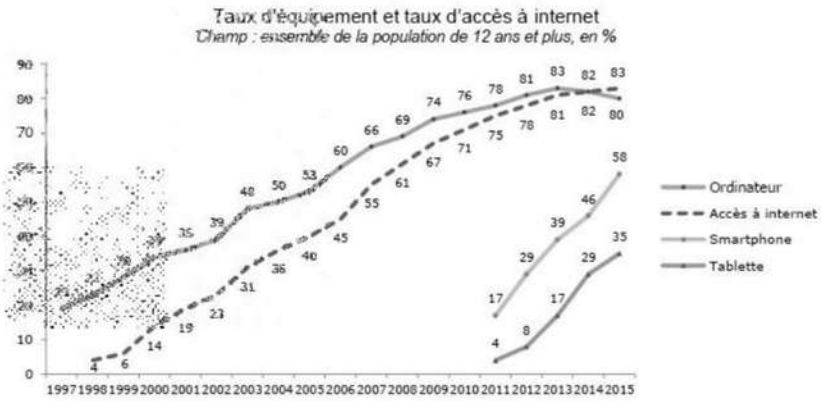
L'utilisation des supports mobiles affecte grandement Internet et le Web, que ce soit pour des raisons techniques, ergonomiques ou graphiques. Les smartphones et tablettes se sont démocratisés, mais la variété des systèmes d'exploitation (Android, iOS, Windows Phone, Windows 10, Symbian, BlackBerry OS…), des navigateurs et des objets (chaque smartphone a ses propres caractéristiques : puissance, résolution, dimensions…) influence directement le développement des pages web pour ces supports.

Dans les faits, il faut admettre que les tablettes offrent des résolutions parfois plus fines et larges que celles des écrans d'ordinateurs ; leur impact sur le design d'un site web est donc à nuancer. Au contraire, les résolutions des smartphones sont assez variées et sont à prendre absolument en compte pour les graphistes afin de restituer au mieux des pages web sur les petits écrans.

Côté chiffres, les voyants sont au vert pour les objets nomades et la mobilité est devenue un facteur clé de développement pour nombre de start-ups et marques. L'Arcep a publié une étude fin novembre 2015 et avance que 58 % des Français possèdent un smartphone et 35 % une tablette, avec une augmentation fulgurante de ces chiffres depuis 2011 (source : http://goo.gl/XfvFq3).

Figure 2-20

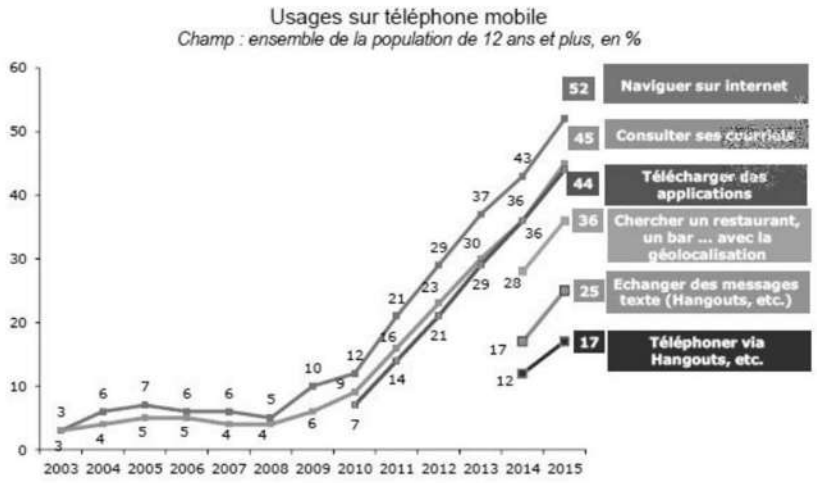
Répartition des équipements fixes et mobiles en France



52 % des sondés ont indiqué qu'ils utilisaient essentiellement les supports mobiles pour naviguer sur Internet, 44 % pour télécharger des applications et 36 % pour chercher des informations géolocalisées (restaurants, bars…). Ces chiffres prouvent qu'avoir des sites web et applications mobiles vraiment adaptés pour les mobinautes est devenu fondamental. Les parts de marché de ces objets nomades sont trop significatives pour être négligées par les webmasters, développeurs et référenceurs.

Figure 2-21

Principaux usages des supports mobiles



Dans le même temps, Google a indiqué que plus de 50 % des recherches proviennent des supports mobiles dans le monde (source : http://goo.gl/Urk5RG). Nous pouvons imaginer que cette croissance forte d'usages mobiles s'applique pour tous les outils de recherche (Bing et Yandex ont également mis en avant la mobilité), même si les proportions varient certainement. De ce fait, ils sont touchés de plein fouet par la mobilité et se sont penchés sur ces questions depuis quelques mois pour améliorer la pertinence des résultats et la navigation dans les SERP mobiles.

Mobilegeddon : nouvelle vie des sites compatibles mobiles

Le nombre de mobinautes croît énormément depuis des années, au point que le nombre de recherches mobiles sur Google a dépassé celui des recherches via ordinateurs en octobre 2015 (source : http://goo.gl/PKDY8M). Google ne s'y est pas trompé et a déployé de nombreux efforts pour valoriser Google mobile et ses applications Android et iOS.

Face à ce tournant historique en matière de recherche, le moteur de Mountain View avait pris les devants en annonçant l'arrivée d'un nouveau critère de positionnement dès le 26 février 2015 (source : http://goo.gl/WqCoK4). Ce premier communiqué officiel avait pour objectif d'accorder un peu de temps aux gestionnaires de sites web pour créer ou obtenir des versions mobiles de leur page web, avant que le critère ne soit officiellement déployé au sein de l'algorithme dans le monde le 21 avril 2015.

D'où vient le nom de « mobilegeddon » ?

En réalité, plusieurs porte-paroles de Google avaient déjà évoqué une transition vers les mobiles dès la fin de l'année 2014, mais sans donner de détails (source : http://goo.gl/U98Mwe). Plusieurs signes avant-coureurs tels que l'ajout de l'analyse mobile dans l'outil PageSpeed Insights ou dans la Search Console à propos de la compatibilité mobile laissaient penser que le mobile allait prendre de l'importance.

C'est pourtant l'officialisation début 2015 qui a eu l'effet d'une bombe, au point que les experts américains lui ont attribué le nom de « mobilegeddon ». Ce terme n'a rien d'officiel, mais démontre tout l'engouement et l'impact de ce nouveau critère de ranking dédié uniquement au moteur de recherche mobile.

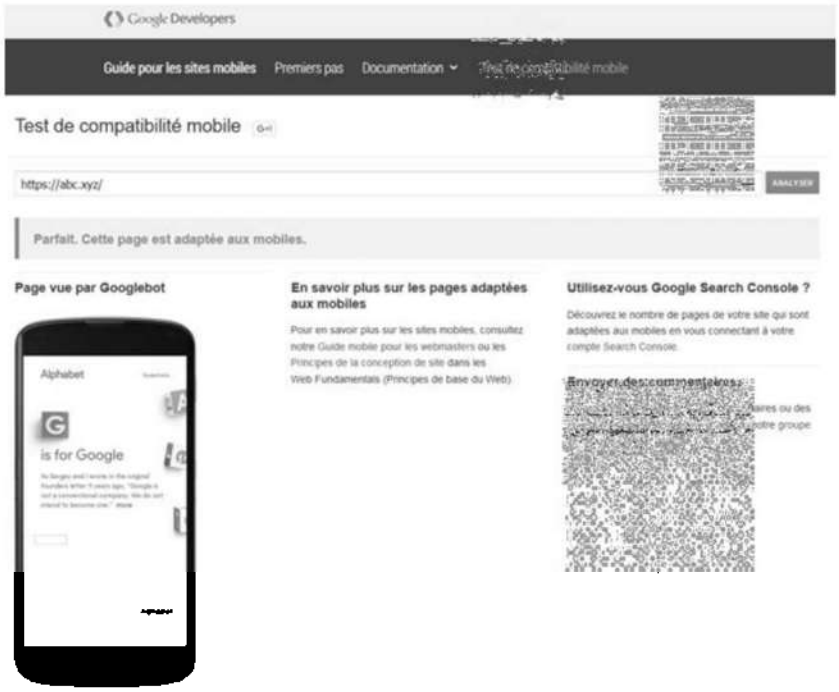
Sur le plan du positionnement, le fait d'avoir des pages compatibles mobiles améliore leur classement dans les SERP mobiles au détriment des pages web inadaptées. Dans les faits, les pages non compatibles mobiles ne sont pas sanctionnées ; elles perdent uniquement des positions à cause du bonus de positionnement que les pages concurrentes obtiennent.

Googlebot-mobile vérifie la compatibilité mobile en direct, c'est-à-dire plusieurs critères déterminant la bonne adaptation des pages web pour les petits et moyens écrans. Un label *mobile-friendly* (ou « site mobile » en français) s'affiche dans les pages de résultats sur Google mobile pour indiquer aux mobinautes si les liens naturels proposés mènent vers des pages adaptées.

Avant d'évoquer les critères principaux analysés par l'algorithme de Google, sachez que la firme a mis à disposition un outil de test de la compatibilité mobile (source : https://goo.gl/niWnyR) et a ajouté une option dédiée au sein de la Google Search Console afin d'aider les développeurs à mieux cerner les points qui ne vont pas dans leur adaptation mobile.

Figure 2-22

Exemple du site d'Alphabet (holding de Google) compatible mobile



Notons que l'outil PageSpeed Insights peut également faire office de bon complément dans la mise en place d'une version adaptée à tous les supports, au même titre que les autres fonctions proposées par Google.

Le communiqué officiel de Google n'a pas indiqué en détail les critères pris en compte pour la compatibilité mobile, mais les outils mis à disposition ainsi que la documentation ont permis d'en savoir davantage à ce sujet. Voici donc certains des facteurs analysés par l'algorithme de Google :

- configurer la fenêtre d'affichage à l'aide d'une balise HTML `<meta> "viewport"`, qui a pour objectif d'indiquer aux navigateurs et robots le format d'adaptation des pages web ;
- ne pas utiliser de technologies bloquantes ou non lues par les mobiles, le Flash par exemple ;
- adapter la taille des polices pour que les utilisateurs n'aient pas à zoomer pour lire, quel que soit l'écran ;
- limiter les dimensions des pages en fonction de l'écran (mise en page fluide ou largeur à 100 % en CSS) et éviter tout défilement horizontal ;
- bien espacer les liens et boutons pour améliorer l'ergonomie et éviter les clics intempestifs dus à l'imprécision du tactile ;
- éviter les interstitiels d'installation d'applications mobiles qui s'affichent devant les contenus ou les plug-ins bloquants. Ceci est pénalisé depuis une mise à jour de l'algorithme *mobile-friendly* du 4 novembre 2015 (source : https://goo.gl/gTHWao).

Figure 2-23

Test de la compatibilité mobile dans PageSpeed Insights

Google mobile aura-t-il un index indépendant ?

John Mueller ainsi que Gary Illyes ont plusieurs fois évoqué la possibilité pour Google de créer un index de pages entièrement consacré aux supports mobiles. Cette question peut en effet se poser puisque nous constatons que de plus en plus de sites proposent des versions parallèles adaptées à la mobilité : soit des sites mobiles à part entière, soit des versions en AMP HTML (nous allons détailler cela par la suite) ou encore avec l'App Indexing.

Le 23 novembre 2015, lors d'un *hangout* sur Google+, Vincent Courson (équipe webspam à Google Dublin) et Zineb Ait Bahajji (webmaster trends analyse à Google Zurich) ont confirmé que Google songe à créer un index mobile mais que rien ne devrait voir le jour avant longtemps (source : https://goo.gl/40mCAh).

Bing, Yandex et les mobiles ?

Google n'est plus le seul moteur de recherche à se pencher sur la compatibilité mobile. C'est notamment le cas de son poursuivant Bing et du leader de la recherche russe Yandex.

En Russie, Yandex a ajouté un label *mobile-friendly* dans ses SERP depuis le 20 novembre 2015 pour montrer aux utilisateurs les sites compatibles mobiles (source : http://goo.gl/Hv7cQt). Si la firme a confirmé qu'aucun boost de ranking n'était encore associé à ces facteurs d'analyse, cela pourrait arriver prochainement. Il est possible de tester la compatibilité mobile d'un site dans les Yandex Webmaster Toolkit, qui se basent sur quelques critères :

- présence ou non de la balise `<meta> "viewport"` ;
- absence de défilement horizontal (donc adaptation de l'écran sur la largeur, quelle que soit la résolution mobile) ;
- absence de technologies bloquantes comme Adobe Flash ou Silverlight de Microsoft ;
- absence de scripts bloquants en JavaScript.

Début 2016, Yandex n'en est encore qu'à ses balbutiements en matière d'analyse de compatibilité mobile, mais cela devrait vite s'améliorer pour devenir un facteur de positionnement du moteur.

De son côté, Bing a réagi plus rapidement, dès le 14 mai 2015 exactement, soit un peu moins d'un mois après Google (source : https://goo.gl/5KJVDc). Dès cette date, un label *mobile-friendly* est apparu dans les SERP sur mobile et Bing a confirmé accorder une valorisation du positionnement pour les pages compatibles avec les supports nomades.

Depuis le 12 novembre 2015, Bing a même déployé un outil de test pour les utilisateurs. Cet outil est disponible via une URL directe (source : https://goo.gl/1OpJCk) ou dans Bing Webmaster Tools, par le biais de l'option *Test d'adéquation à l'utilisation sur appareil mobile* disponible dans l'onglet *Diagnostics et outils*.

Seuls quelques facteurs sont analysés, à l'instar de l'outil de Yandex :

- vérification de configuration de la fenêtre (`<meta> "viewport"`) ;
- vérification du contrôle du zoom (`<meta> "viewport"`) ;
- adaptation des contenus à l'écran (pas de défilement horizontal) ;
- lisibilité des contenus textuels sur les petits écrans ;
- tailles et espacements adaptés des liens dans les pages.

Ces deux moteurs ne vont certainement pas aussi loin que Google actuellement, mais prouvent que la compatibilité mobile est devenue un enjeu essentiel dès 2015. Il n'est plus possible d'imaginer la conception de pages web sans leur pendant sur mobile, au risque de perdre le faible boost de ranking, mais surtout des visiteurs de plus en plus nombreux sur les supports itinérants.

Figure 2-24

Test de la compatibilité mobile dans l'outil de test de Bing

Différentes alternatives mobiles

Méthodes conseillées par Google

Il existe plusieurs possibilités pour créer des versions compatibles avec les mobiles, comme Google l'indique dans sa documentation (source : https://goo.gl/5whBOL).

- **Responsive web design :** création d'une mise en page fluide et qui s'adapte sur tous les supports. Généralement, la mise en place technique est assez aisée et pratique. Ce sont essentiellement des adaptations de propriétés CSS qui modifient la mise en page et la mise en forme selon le support visité.
- **Dynamic serving :** des en-têtes `"Vary"` sont envoyés au serveur et permettent d'afficher une version différente du code selon le robot (`user-agent`) qui crawle la page. Dans ce cas, l'URL reste identique, mais le code HTML et CSS diffère selon la version renvoyée par le serveur (en fonction du `user-agent`). La mise en place demande plus de compétences et il faut prendre garde à ne pas faire d'erreurs, car cela pourrait être interprété comme du *cloaking* (nous évoquerons ce sujet dans le prochain chapitre), fortement pénalisé par Google.
- **URL distinctes :** les spécialistes parlent souvent de « sites mobiles » dans ce cas, à savoir des versions de sites entièrement construites pour les supports nomades. L'objectif est d'offrir des versions pleinement adaptées pour chaque support, mais cela signifie aussi que le référencement est à gérer entièrement pour chaque mouture des pages (URL et contenus différents).

Figure 2-25

Méthodes de création de sites compatibles mobiles

Configuration	Est-ce que mon URL reste la même ?	Est-ce que mon code HTML reste le même ?
Responsive Web Design	✓	✓
Dynamic serving	✓	✗
URLs distinctes	✗	✗

Toutes les techniques ont des avantages et des inconvénients, mais leur niveau de technicité, leur coût et leurs risques diffèrent. Google préconise des versions de sites web en responsive design ; cela a l'avantage d'adapter des contenus déjà indexés et connus par Google. En effet, cela évite de risquer de tomber dans du cloaking comme avec la méthode du *dynamic serving*, ou même de devoir mener deux référencements de front avec un site mobile. Tout est une question d'usage, mais pesez bien le pour et le contre avant de vous lancer dans l'aventure mobile.

Quid des applications mobiles ?

Les applications mobiles constituent des systèmes à part dans le monde des mobiles. Google avait tout de même indiqué dans son communiqué du 26 février 2015 son intérêt pour l'App Indexing, à savoir l'indexation des pages profondes issues des applications mobiles.

Dès le 21 avril 2015, des liens d'applications ont pu mieux ressortir dans les SERP mobiles et être mis en avant. Toutefois, Google n'avait alors pas encore précisé si ces derniers bénéficiaient d'un bonus de positionnement. Il aura fallu attendre le SMX de New York pour que Mariya Moeva, en charge des mobiles chez Google, indique le 30 septembre 2015 qu'une aide au ranking serait accordée aux utilisateurs de l'App Indexing.

Projet AMP HTML

AMP est le sigle du projet open source Accelerated Mobile Pages (source : https://www.ampproject.org) soutenu par une large communauté de webmasters et de marques dont Google, Twitter, Pinterest, LinkedIn, Parse.ly ou encore Adobe Analytics.

L'objectif est d'accélérer considérablement le temps de chargement des pages web sur les supports mobiles, grâce à un système de code HTML réécrit selon les versions à afficher. AMP permet de créer des versions statiques des pages web (un peu comme si elles étaient en cache) qui accélèrent le chargement des contenus sans perdre de temps avec les ressources lourdes (images, vidéos, iframes...) déjà enregistrées de manière statique.

AMP HTML implique des changements techniques dans le code, mais aussi de bonnes méthodes de détection des supports mobiles (en CSS, JavaScript ou PHP par exemple) pour pouvoir être utilisé à bon escient et être pleinement fonctionnel.

Google soutient le projet Accelerated Mobiles Pages. La firme a encouragé les webmasters à proposer ce type de versions mobiles car son chargement est bien plus rapide que celui d'une page classique. Nous pouvons donc imaginer qu'un site mobile ne propose pas de responsive web design mais uniquement une alternative en AMP HTML pour les mobiles (bien que ce soit déconseillé en général).

Figure 2-26

Google propose des SERP avec une démo en AMP HTML

Google a indiqué prendre en compte officiellement les alternatives des pages web en AMP HTML dès 2016 pour les proposer dans les SERP mobiles quand elles existent (source : https://goo.gl/GmwJGQ). De ce fait, un site web peut voir ses pages en AMP HTML indexées par Google et présentées dans les versions mobiles du moteur pour les mobinautes, au détriment des pages classiques réservées aux navigateurs de bureau.

Sur le plan technique, il s'agit uniquement de réécrire le code HTML avec les balises AMP HTML (variantes simples des balises classiques) pour toutes les ressources lourdes à charger (images, vidéos, iframes, fichiers…). Certaines règles sont à respecter et il faut inclure quelques scripts en JavaScript fournis par le projet, notamment sur GitHub (source : https://goo.gl/YdQ9Ci). Nous allons étudier sa mise en application dans la sous-partie suivante.

L'AMP HTML peut aider les référenceurs à mieux indexer des pages web via Googlebot-mobile ou à mieux les positionner puisqu'un boost de positionnement est à l'étude pour 2016 (source : http://goo.gl/KBh5BE). L'AMP HTML accélère le temps de chargement des pages web et nous avons vu que ce critère compte dans la pertinence de Google (notion de PageSpeed), et si un bonus de ranking s'ajoute à ce gain de performance, cela devrait inciter les webmasters à proposer des versions en AMP.

Concernant l'indexation, la seule règle à bien respecter est la mise en place d'URL canoniques pour que Google distingue bien les pages web classiques des versions en AMP HTML. Cela lui permet de favoriser la version adéquate en fonction des supports utilisés.

Exemples de mises en application

Passage en responsive web design

Notre objectif n'est pas d'apprendre complètement à réaliser du responsive web design ; d'autres ouvrages bien plus adaptés vous en apprendront davantage à ce sujet, notamment celui de l'inventeur du concept, Ethan Marcotte, publié aux éditions Eyrolles. Ici, nous allons surtout traiter des points qui sont essentiels dans la mise en place d'un site au design adaptatif pour améliorer la note du PageSpeed :

- l'utilisation idéale de la balise `<meta> "viewport"` ou de la fonction CSS `@viewport` ;
- la création d'une mise en page fluide avec médias flexibles ;
- l'optimisation des images en fonction des supports.

La balise `<meta> "viewport"` correspond grosso modo à la surface occupée par la fenêtre du navigateur, quel que soit le support utilisé. Son usage, initialement mis en place par Apple sur les iPhone, a permis d'indiquer aux supports les dimensions de la fenêtre à respecter. Il est donc possible d'insérer des tailles fixes si nécessaire mais, en général, le concept de design adaptatif recommande l'usage des balises `<meta> "viewport"` relatives sous la forme suivante :

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Après quelques années, le W3C a intégré la fonction `@viewport` dans sa spécification CSS pour qu'elle soit reconnue officiellement, malgré son caractère propriétaire d'origine (Apple).

Cette dernière est compatible avec la plupart des navigateurs mobiles, dont Internet Explorer 10/11. Son usage n'est pas plus complexe que celui de la balise `<meta>` éponyme, mais semble plus logique pour l'avenir du Web car il suffit de l'ajouter dans un fichier CSS :

```
@viewport { width:device-width;zoom:1 }
```

Le concept de responsive web design passe ensuite par l'usage des médias et de mises en page flexibles. Plusieurs règles CSS sont à respecter :

- utiliser des unités relatives comme `em`, `rem`, `ex`, `%` ou `px` dans certains cas ;
- rendre les médias flexibles avec une ligne de code CSS comme `img, object, embed, iframe, video, audio { width:100%; max-width:100%; height:auto }` ;
- utiliser les *media queries* avec la fonction `@media` pour exécuter les CSS en fonction des dimensions des fenêtres de chaque support. Pour ce faire, il suffit d'entrer des conditions comme `@media all and (max-width:480px) { /* code CSS */ }`.

L'ensemble de ces considérations permet d'obtenir une mise en page fluide selon la taille de la fenêtre du navigateur et de l'écran utilisé. Cependant, le problème réside dans l'usage de grandes images non adaptées aux smartphones par exemple. En effet, si nous avons une image initiale de 960 px de largeur, la propriété `width:100%;` l'affiche à 960 px lorsqu'il s'agit d'un écran d'ordinateur, mais la réduit proportionnellement sur petit écran, sauf que l'image chargée reste identique.

Dans ces circonstances, les performances sont amoindries par le chargement d'une trop grande image pour des écrans de taille réduite ; il serait préférable de charger une image adaptée.

Il n'existe aucune méthode parfaite, car nous ne pouvons pas prévoir autant d'images que de tailles d'écrans existant sur le marché. Il faut donc procéder à des choix en fonction des points de rupture les plus courants suivant leurs résolutions d'écran, à savoir 360 px, 480 px, 640 px, 768 px, 1 024 px et 1 280 px.

Pour les grandes résolutions, le problème est réglé par défaut car elles sont assez larges pour afficher l'ensemble des contenus. Ce sont donc les plus petites résolutions qui doivent mériter notre attention. Il suffit alors de créer plusieurs images qui seront chargées en fonction des diverses media queries utilisées, comme ici :

- `image-360.jpg` en dessous de 480 px de largeur ;
- `image 640.jpg` pour les résolutions de 640 px et 768 px de largeur ;
- `image-960.jpg` pour les grands écrans.

Vers du responsive design en Flexbox CSS ?

Les Flexbox CSS offrent une nouvelle forme de design adaptatif fluide et flexible qui risque de s'imposer dans les prochaines années. Il s'agit d'un modèle de boîte flexible utilisant la propriété CSS `display:inline-flex` ou `display:flex`.

Les Flexbox offrent une souplesse inégalée jusqu'à présent pour les intégrateurs web et sont une solution idéale pour mettre en place du responsive design. Voici les grands changements apportés par les propriétés CSS des Flexbox.

- Distribution horizontale ou verticale des blocs HTML, avec ou sans retour forcé à la ligne (et tout ceci sans `float` en CSS), grâce aux propriétés `flex-direction` et `flex-wrap` (ou les deux combinées dans `flex-flow`). Les Flexbox permettent aussi de remplir l'espace vide disponible. Par exemple, il est tout à fait possible de bloquer un élément en pied de page sans utiliser les positions fixes en CSS et tout cela sera fluide. Il suffit d'user de la propriété `justify-content` avec la valeur `flex-end` pour caler un élément à la fin (donc en pied de page selon vos réglages de Flexbox).
- Alignement des blocs entre eux : centré (verticalement ou horizontalement), justifié, réparti (les blocs se répartissent l'espace disponible). Tout ceci s'effectue avec les propriétés `justify-content` et `align-items`.
- Ordonnancement et réorganisation des blocs entre eux grâce à la propriété `order`. Il est ainsi très simple de passer le troisième bloc HTML en première position par exemple, ce qui se révèle très pratique pour du design adaptatif.

D'autres propriétés existent pour aller encore plus loin avec les Flexbox et il est fortement conseillé de se pencher sur cette spécification CSS 3 innovante et tant attendue depuis des années par les intégrateurs du monde entier.

Retenez que tout ceci est rétrocompatible avec de nombreux navigateurs (seul Internet Explorer est un peu à la traîne), mais surtout adapté aux outils mobiles (les navigateurs mobiles prennent tous les Flexbox en compte).

JavaScript/jQuery pour les mobiles

Le langage JavaScript et ses nombreux frameworks comme jQuery aident à optimiser de nombreux facteurs pour les mobiles. Cela peut être de simples déplacements ou masquages de blocs, à l'instar du responsive web design en CSS, mais aussi des techniques plus poussées comme des préchargements d'images, des gestions de contenus côté client, etc.

Nous n'entrerons pas dans tous ces détails techniques ici car les variations sont bien trop nombreuses pour être traitées, notamment à cause de la multitude de bibliothèques et frameworks JavaScript. En effet, il serait impossible d'évoquer le sujet de la même manière pour des utilisateurs de serveurs NodeJS ou pour ceux qui préfèrent Apache ou IIS. De même, un utilisateur de techniques avancées avec BackBoneJS, AngularJS de Google ou ReactJS de Facebook n'aura pas la même approche du Web mobile qu'un gestionnaire de site dit « classique ».

En revanche, il faut absolument rappeler les méthodes de détection des mobiles qui ont déjà été rapidement évoquées dans la sous-partie « Éviter les redirections vers la page de destination » de la section « PageSpeed et vitesse de chargement des pages ».

Plusieurs possibilités permettent de détecter les mobiles, soit via l'analyse des `user-agent` directement, soit par l'analyse de la résolution ou de la taille d'écran. Les deux types sont intéressants, mais pas dans les mêmes cas de figure.

La détection des résolutions ou des écrans peut se faire avec des fonctions JavaScript en natif ou via jQuery, comme ceci :

```
<script>
// Première écriture avec la fonction matchMedia
if(window.matchMedia("(max-width:768px)").matches) {
   // Code effectué si la largeur est inférieure à 768 px
}

// Seconde écriture avec jQuery
if($(window).width() < 768) { // Détection via jQuery
          // Code effectué si la largeur est inférieure à 768 px
}
</script>
```

Cette méthode de détection est pratique car elle fonctionne avec tous les navigateurs et supports, mais elle peut avoir aussi quelques inconvénients selon les usages :

- elle s'active même lorsqu'une fenêtre est réduite sur un écran d'ordinateur ;
- elle dépend surtout des résolutions et non des largeurs d'écrans (mais c'est possible) ; comme les supports mobiles offrent de plus en plus souvent de larges résolutions, cela devient parfois obsolète.

L'autre méthode consiste à détecter les `user-agent` des mobiles avec une expression régulière. Cela fonctionne aussi bien en JavaScript/jQuery qu'avec les autres langages comme Python, PHP, Java... Cette méthode a l'avantage de ne s'appliquer que dans un vrai contexte de mobilité, mais elle a aussi deux défauts.

- Il faut bien connaître la liste des agents mobiles. Or, comme cela évolue rapidement, c'est très compliqué.
- Certains systèmes d'exploitation renvoient de mauvais agents ou sont mal détectés, donc la détection peut parfois mal se faire.

Voici un exemple de fonction JavaScript de détection des `user-agent` réécrite à partir des codes clés en main fournis par le site Detect Mobile Browsers (source : http://detectmobilebrowsers.com) :

```
function isMobile(a){
if(/(android|bb\d+|meego).+mobile|avantgo|bada\/|blackberry|blazer|compal|elaine|
fennec|hiptop|iemobile|ip(hone|od)|iris|kindle|lge |maemo|midp|mmp|mobile.+firefo
x|netfront|opera m(ob|in)i|palm( os)?|phone|p(ixi|re)\/|plucker|pocket|psp|series
(4|6)0|symbian|treo|up\.(browser|link)|vodafone|wap|windows ce|xda|xiino/i.test(a
)||/1207|6310|6590|3gso|4thp|50[1-6]i|770s|802s|a wa|abac|ac(er|oo|s\-)|ai(ko|rn)
|al(av|ca|co)|amoi|an(ex|ny|yw)|aptu|ar(ch|go)|as(te|us)|attw|au(di|\-m|r |s
)|avan|be(ck|ll|nq)|bi(lb|rd)|bl(ac|az)|br(e|v)w|bumb|bw\-
(n|u)|c55\/|capi|ccwa|cdm\-|cell|chtm|cldc|cmd\-|co(mp|nd)|craw|da(it|ll|ng)|dbte
|dc\-s|devi|dica|dmob|do(c|p)o|ds(12|\-d)|el(49|ai)|em(l2|ul)|er(ic|k0)|esl8
|ez([4-7]0|os|wa|ze)|fetc|fly(\-|_)|g1 u|g560|gene|gf\-5|g\-mo|go(\.
w|od)|gr(ad|un)|haie|hcit|hd\-(m|p|t)|hei\-|hi(pt|ta)|hp( i|ip)|hs\-c|ht(c(\-|
|_|a|g|p|s|t)|tp)|hu(aw|tc)|i\-(20|go|ma)|i230|iac( |\-|\/)|ibro|idea|ig01|ikom|i
m1k|inno|ipaq|iris|ja(t|v)a|jbro|jemu|jigs|kddi|keji|kgt( |\/)|klon|kpt |kwc\-
|kyo(c|k)|le(no|xi)|lg( g|\/(k|l|u)|50|54|\-[a-w])|libw|lynx|m1\-w|m3ga|m50\/|ma(
te|ui|xo)|mc(01|21|ca)|m\-cr|me(rc|ri)|mi(o8|oa|ts)|mmef|mo(01|02|bi|de|do|t(\-|
|o|v)|zz)|mt(50|p1|v )|mwbp|mywa|n10[0-2]|n20[2-3]|n30(0|2)|n50(0|2|5)|n7(0(0|1)|
10)|ne((c|m)\-|on|tf|wf|wg|wt)|nok(6|i)|nzph|o2im|op(ti|wv)|oran|owg1|p800|pan(a|
d|t)|pdxg|pg(13|\-([1-8]|c))|phil|pire|pl(ay|uc)|pn\-
2|po(ck|rt|se)|prox|psio|pt\-g|qa\-a|qc(07|12|21|32|60|\-[2-7]|i\-
)|qtek|r380|r600|raks|rim9|ro(ve|zo)|s55\/|sa(ge|ma|mm|ms|ny|va)|sc(0
1|h\-|oo|p\-)|sdk\/|se(c(\-|0|1)|47|mc|nd|ri)|sgh\-|shar|sie(\-|m)|sk\-0|sl(45|id
)|sm(al|ar|b3|it|t5)|so(ft|ny)|sp(01|h\-|v\-|v )|sy(01|mb)|t2(18|50)|t6(00|10|18)
|ta(gt|lk)|tcl\-|tdg\-|tel(i|m)|tim\-|t\-mo|to(pl|sh)|ts(70|m\-|m3|m5)|tx\-
9|up(\.b|g1|si)|utst|v400|v750|veri|vi(rg|te)|vk(40|5[0-3]|\-v)|vm40|voda|vulc|vx
(52|53|60|61|70|80|81|83|85|98)|w3c(\-| )|webc|whit|wi(g
|nc|nw)|wmlb|wonu|x700|yas\-|your|zeto|zte\-/i.test(a.substr(0,4))) {
            return true;
      } else {
            return false;
         }
}
```

Vous pouvez constater la complexité de l'expression régulière interminable qui compose ce type de fonction, mais la détection est généralement de bonne facture. Ensuite, pour l'utiliser, il suffit de l'appeler avant tout code que vous souhaitez appliquer aux mobiles.

```
if(isMobile(navigator.userAgent||navigator.vendor||window.opera)) {
   // code à appliquer pour les mobiles
}
```

Il ne vous reste plus qu'à développer vos propres applications pour les mobiles et pour améliorer votre responsive web design.

Et les applications web mobiles avec jQuery ?

Il est possible de créer des applications web mobiles avec de nombreuses bibliothèques, dont jQuery. En effet, jQuery Mobile facilite la conception des web-apps. Il existe aussi d'autres systèmes plus évolués comme Cordova avec PhoneGap pour ce faire. Le mobile avec JavaScript ne se limite donc pas à des pages web réadaptées pour les supports nomades...

AMP HTML en pratique

Nous avons évoqué la réécriture open source AMP HTML au début de cette partie. Il est grand temps de voir comment cela fonctionne. N'ayez crainte, cela n'est pas si complexe que cela peut le paraître, sauf pour certains cas particuliers (scripts ou médias notamment).

L'idéal est de s'appuyer sur les évolutions du projet, soit sur le site officiel, soit directement sur GitHub (source : https://goo.gl/YdQ9Ci). La bibliothèque complète est fournie, avec des exemples de mise en place. Procédons par étape pour maîtriser AMP HTML.

1. Modifier l'en-tête des pages web avec quelques instructions importantes pour valider l'AMP (en gras), sans oublier la présence du `viewport` et l'ajout de l'attribut `amp` dans la balise `<html>`.

```
<!doctype html>
<html amp>
<head>
<meta charset="utf-8">
<link rel="canonical" href="hello-world.html"/>
<meta name="viewport" content="width=device-width,initial-scale=1">
<style>body{opacity:0}</style><noscript><style>body{opacity:1}</style></noscript>
<script async src="https://cdn.ampproject.org/v0.js"></script>
</head>
...
```

2. Entrer les scripts utiles pour les cas spécifiques et inscrire les balises AMP HTML nécessaires. Deux possibilités s'offrent à vous : soit vous créez des doublons de pages, soit vous utilisez des fonctions de réécriture automatique des URL.

La liste des balises et attributs est fournie dans le projet, mais voici un tableau récapitulatif :

BALISES HTML NATIVES	BALISES EN AMP HTML
`img`	`amp-img`
`video`	`amp-video`
`audio`	`amp-audio`
`iframe`	`amp-iframe`
`iframe YouTube`	`amp-youtube`
`iframe Twitter`	`amp-twitter`

BALISES HTML NATIVES	BALISES EN AMP HTML
iframe Vine	amp-vine
iframe publicitaire	amp-ad
pixel de tracking	amp-pixel
animation GIF	amp-anim
slider et carousel	amp-carousel
modale ou lightbox	amp-lightbox ou amp-image-lightbox
chargement de fonts	amp-font
texte adaptatif	amp-fit-text
listes en JSON	amp-list
svg	Les éléments SVG sont pour la plupart autorisés.

Certains types de contenus ont besoin de scripts JavaScript dédiés pour fonctionner. Il faut ajouter une balise `<script>` avec l'attribut `"custom-element"` associé. C'est notamment le cas des `iframe` ou des vidéos YouTube par exemple. Il faut ajouter ces deux lignes de code dans l'en-tête pour que ces balises fonctionnent en AMP :

```
<script custom-element="amp-iframe" async
    src="https://cdn.ampproject.org/v0/amp-iframe-0.1.js"></script>
<script custom-element="amp-youtube" async
   src="https://cdn.ampproject.org/v0/amp-youtube-0.1.js"></script>
```

Il est également possible de créer des types de balises personnalisés en ajoutant un script spécifique et son type de balise, comme ceci :

```
<script async custom-template="amp-perso"
src="https://cdn.ampproject.org/v0/amp-perso-0.1.js"></script>
<template type="amp-perso" id="template1">
Hello {{World}} !
</template>
```

Retenez que l'essentiel du travail consiste à faire basculer les balises HTML en balises AMP HTML pour améliorer les performances sur les mobiles. Ce sont essentiellement les principaux médias qui sont affectés, comme les images, les vidéos, les animations et les scripts ; donc, cela limite les modifications.

Figure 2-27

Exemple de code transformé en AMP HTML

```
<figure>
  <amp-img class="full-bleed" placeholder
      src="img/sea@1x.jpg"
      srcset="img/sea@1x.jpg 1x, img/sea@2x.jpg 2x"
      layout="responsive" width="360"
      alt="Fusce pretium tempor justo, vitae consequat dolor maximus eget."
      height="216">
  </amp-img>
  <figcaption>
    Fusce pretium tempor justo, vitae consequat dolor maximus eget.
  </figcaption>
</figure>
<hr>

<p>
  Cum sociis natoque penatibus et magnis dis parturient montes,
  nascetur ridiculus mus. Nulla et viverra turpis. Fusce
  viverra enim eget elit blandit, in finibus enim blandit. Integer
  fermentum eleifend felis non posuere. In vulputate et metus at
  aliquam. Praesent a varius est. Quisque et tincidunt nisi.
  Nam porta urna at turpis lacinia, sit amet mattis eros elementum.
  Etiam vel mauris mattis, dignissim tortor in, pulvinar arcu.
  In molestie sem elit, tincidunt venenatis tortor aliquet sodales.
  Ut elementum velit fermentum felis volutpat sodales in non libero.
  Aliquam erat volutpat.
</p>

<div class="ad-container">
  <amp-ad width=300 height=200
      type="adsense"
      data-ad-client="ca-pub-9350112648257122">
  </amp-ad>
</div>
```

Concluons sur la mise en place de l'AMP HTML avec un exemple de fonction PHP qui transforme dynamiquement des balises HTML en code AMP, à l'aide d'expressions régulières. Cette fonction ne gère pas tous les types exposés précédemment, mais donne les bases de ce qu'il est possible de faire pour gagner du temps et automatiser la mutation pour les mobiles.

Il faudrait par exemple créer au moins trois fonctions :

- fonction `isAMP()` pour vérifier que l'URL en cours est une version en AMP, repérée grâce à un suffixe /amp/ ou ?amp=1 dans l'adresse web ;
- fonction `getCaronicalAMP()` pour ajouter automatiquement l'URL canonique sans le suffixe AMP de l'URL ;
- fonction `setTagAMP()` pour modifier dynamiquement les balises HTML natives en AMP HTML. Ici, l'exemple présenté est simple et ne prend pas en compte toutes les modifications d'attributs possibles.

En effet, de nombreux attributs existent uniquement en AMP HTML, comme `layout="responsive"`, et le but est surtout de comprendre le principe.

```
// Fonction de vérification de l'URL pour voir s'il s'agit d'une page en AMP HTML
function isAMP() {
   $url = $_SERVER['REQUEST_URI'];
          $regex = "#([&?]amp=1|/amp/?)$#i";
          return preg_match($regex, $url);
}

// Fonction d'ajout des URL canoniques
function getCanonicalAMP() {
   $url = "//".$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI'];
          $regex = "#([&?]amp=1|/amp/?)$#i";
          $newURL = preg_replace($regex, "", $url);
          return '<link rel="canonical" href="'.$newURL.'"/>';
}

// Fonction principale pour modifier les balises en AMP HTML
function setTagAMP($texte = "") {
   $regexIMG = "#<img([^>]+)/>#i";
          $texte = preg_replace($regexIMG, "<amp-img$1></amp-img>", $texte);

          $regexYouTube = '#</?iframe([^>]+)?src=["\']https?://(www.)?
youtube.com/embed/([a-z0-9]+)["\']([^>]+)>(\s+)?</iframe>#i';
          $texte = preg_replace($regexYouTube, '<amp-youtube$1data-videoid="$3"$4>
</amp-youtube>', $texte);

          $regexIframe = "#</?iframe([^>]+)>(\s+)?</iframe>#i";
          $texte = preg_replace($regexIframe, '<amp-iframe$1 $2></amp-iframe>',
                                $texte);
          return $texte;
}
```

Voici comment cela pourrait être intégré en HTML, en partant du postulat que la variable `$contenu` correspond à l'ensemble du texte récupéré dans une base de données :

```
<!DOCTYPE html>
<html <?php if(isAMP()) { echo "amp"; } ?>>
<head>
       <meta charset="utf-8">
       <meta name="viewport" content="width=device-width,initial-scale=1">
       <?php
          // Ajout de toutes les balises et scripts utiles pour le site web
          // (dont iframe et YouTube ici)
          if(isAMP()) {
       ?>
       <style>body{opacity:0}</style><noscript><style>body{opacity:1}</style></noscript>
       <?php echo getCanonicalAMP(); ?>
       <script async src="https://cdn.ampproject.org/v0.js"></script>
```

```
        <script async custom-element="amp-iframe"
                src="https://cdn.ampproject.org/v0/amp-iframe-0.1.js"></script>
        <script async custom-element="amp-youtube"
                src="https://cdn.ampproject.org/v0/amp-youtube-0.1.js"></script>
        <?php
           }
        ?>
</head>
<body><?php echo setTagAMP($contenu); ?></body>
</html>
```

WordPress et AMP HTML

WordPress a développé rapidement un plug-in pour que les sites web profitent de l'AMP HTML ; ce dernier évolue doucement mais sûrement. Il est conseillé de le suivre plutôt sur Github (source : https://goo.gl/BXv33R) car il évolue plus rapidement que les versions classiques des extensions WordPress (source : https://goo.gl/BY2w31). Une fois le plug-in installé, il suffit d'ajouter un suffixe `/amp` ou `?amp=1` aux URL pour afficher la version AMP HTML (même si cette dernière ne s'applique vraiment que sur les supports mobiles).

L'inconvénient est qu'aucune détection native n'est effectuée ; il faut donc modifier manuellement l'URL sur mobile, ce qui est un vrai frein. L'idéal est de procéder à la détection des supports mobiles pour réécrire automatiquement les URL avec `/amp` (ou `?amp=1`) ou de faire des redirections automatiques vers ces pages optimisées. Vous pouvez trouvez des exemples ici : http://goo.gl/xxOrl8.

D'autres plug-ins sont en cours développement et suivent les directives du projet Accelerated Moblie Pages, il faudra regarder cela de près pour ne pas être dépassé et pour trouver les meilleures solutions.

AuthorShip et AuthorRank

AuthorShip

Les réseaux sociaux ont pris une part importante dans les usages quotidiens au point que nombre de référenceurs se sont dits qu'une présence et une notoriété forte sur ces outils pourraient avoir une incidence sur l'indexation, et surtout sur le positionnement. Nous reviendrons sur ce point juste après, lorsque nous évoquerons l'AuthorRank, mais avant cela, il est important d'expliquer la notion d'AuthorShip mise en place par Google et Bing qui permet de déterminer grâce à des systèmes de reconnaissance les auteurs de contenus sur le web.

En 2011, Google avait décidé de créer un système de liaison entre les pages web (ou sites complets) et ses propres outils en l'intitulant AuthorShip. Historiquement, cette interconnexion se faisait un peu dans l'ombre par le biais des adresses Gmail que nous possédions (certes, cela est toujours le cas) mais les changements impliquaient le fait d'indiquer à Google quels sites étaient en relation avec un profil sur son réseau social Google+ notamment (et surtout son auteur). Ces liaisons permettaient entre autres d'afficher les photos des profils Google+ au sein des SERP, ce qui pouvait constituer un atout considérable en termes de visibilité.

Nous parlons de ce phénomène au passé car la donne a changé le 28 août 2014 dans les systèmes de Google, mais nous expliquerons en détail l'intérêt encore présent de ce phénomène d'interconnexion entre contenus et auteurs dont la mutation ne fait que commencer. John Mueller, l'un des porte-parole attitré de Google, a indiqué ce jour-là la fin du marquage de l'AuthorShip et de l'affichage des photos dans les SERP (source : http://goo.gl/PwfEJd). Relatons l'histoire de l'AuthorShip de Google autour de faits marquants.

- 23 décembre 2013 : Google annonce que de moins en moins de photos seront affichées dans les SERP.
- 25 juin 2014 : John Mueller annonce la fin de l'affichage des photos dans les SERP ainsi que le nombre de « cercles » (abonnements sur Google+). Seul le nom de l'auteur reste alors affiché avec éventuellement la date de la publication.

Figure 2-28

Images de profil anciennement affichées dans les SERP de Google

- 28 août 2014 : l'histoire de l'AuthorShip s'arrête avec la suppression du nom des auteurs mais aussi l'arrêt du système de mise en place dans le code source des pages web.

Comment fonctionnait l'AuthorShip de Google ?

Pour qu'un auteur soit reconnu par Google, il suffisait de rajouter une balise `<link />` dans la section `head` des pages web. Google utilisait l'attribut `rel="author"` accompagné d'un lien vers le profil Google+ de l'auteur pour créer cette notion d'AuthorShip, comme dans l'exemple ci-dessous :

```
<link rel="author" href="https://plus.google.com/id-googleplus" />
```

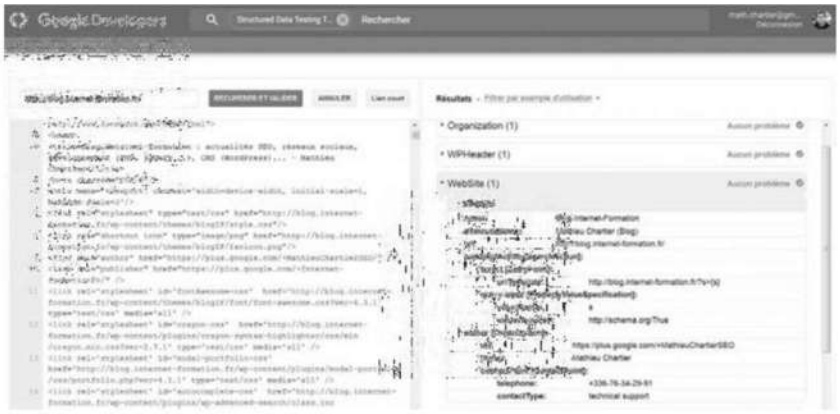
Un autre système passait par des liens classiques (balises `<a>`) en ajoutant en fin d'URL le paramètre `?rel=author`. Ces deux méthodes suffisaient pour être reconnu par le moteur de recherche et afficher les photos dans les SERP.

La disparition de l'AuthorShip sous sa forme existentielle a fait beaucoup parler mais il est important de bien comprendre que cela ne signe pas pour autant l'arrêt de mort du suivi des auteurs ni même d'éventuels systèmes de notation les concernant. Si nous reprenons l'annonce faite par John Mueller, voici ce qui est écrit en américain : « (...) With this in mind, we've made the difficult decision to stop showing authorship in search results. ». En d'autres termes, Google arrête l'affichage de l'AuthorShip dans les résultats de recherche mais cela ne signifie pas que les auteurs n'intéressent plus le moteur...

John Mueller a confirmé qu'il n'est plus utile d'installer la balise `<link rel="author"/>` qui a fait le succès de l'AuthorShip, mais que si cette dernière est encore en place, elle ne posera pas de soucis car elle sera traitée comme n'importe quel autre marqueur sémantique. Il a ensuite renchéri en indiquant que Google misait sur d'autres méthodes (dont Schema.org) pour continuer à faire évoluer ses systèmes de reconnaissance. L'outil de test des données structurées montre que l'auteur est toujours reconnu mais ne bénéficie plus d'un traitement de faveur (source : http://goo.gl/yUNdPM).

Figure 2-29

Récupération des données d'auteur comme tout autre rich snippet



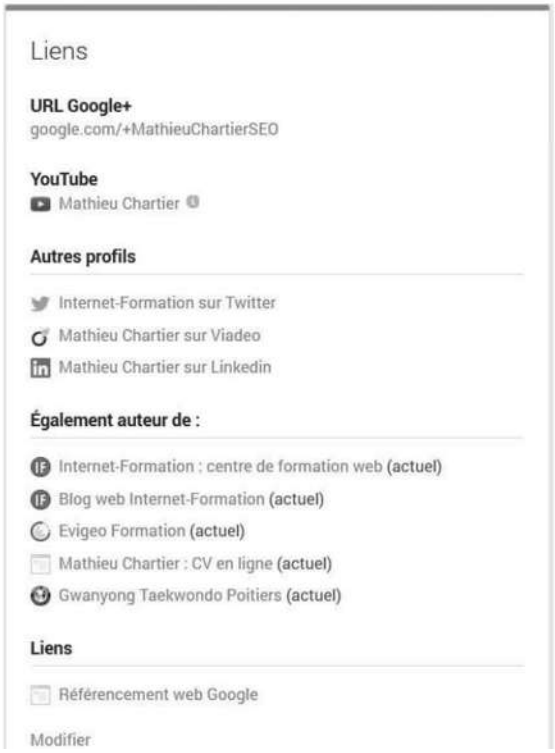
Nous devons donc désormais considérer que l'ancienne méthode n'aboutit plus car Google a estimé que le taux de clics dans les SERP n'était pas impacté par l'affichage des données issues de l'AuthorShip et de ce système (la vraie raison étant sûrement une surabondance de spams dans les SERP). Par conséquent, les informations relatives aux auteurs feront désormais partie d'un tout, au même titre que n'importe quelle autre donnée structurée. Comme nous pouvons nous douter que Google reste intéressé par ces informations, il ne nous reste qu'à savoir comment Google pourrait les récupérer.

- L'inscription à un compte Google Gmail puis à d'autres outils de la firme permet au moteur de recherche de connaître les connexions entre les différents services pour un même auteur.
- Google analyse l'ensemble des sites reliés à un compte Google Analytics ou Google Search Console. Il est donc capable de déterminer quel auteur ou webmaster gère plusieurs sites web. C'est pourquoi il est souvent conseillé de créer plusieurs comptes distincts pour éviter qu'un site pénalisé puisse entraîner dans la foulée d'autres sites d'un même auteur (cette peur est née de Google Panda mais rien ne dit que cet effet boule de neige est appliqué...).
- Google utilise des données structurées différentes de l'attribut `rel="author"` pour capter les informations (notamment avec Schema.org comme John Mueller l'a indiqué sur Google+), bien que cette fonctionnalité HTML fonctionne encore à ce jour comme nous l'avons vu dans l'illustration 2-19.

- Il est encore possible de remplir dans la section « Bio » d'un profil Google+ le bloc « également auteur de » qui permet d'indiquer à Google quels sites appartiennent à un auteur en particulier. Cette fonctionnalité existait déjà dans l'AuthorShip natif du moteur de recherche mais n'a pas disparu comme nous le voyons dans la capture 2-20.
- L'installation d'un badge Google+ permet de connecter le compte d'un auteur à une page (source : http://goo.gl/ZlbT4Z) et donc d'être reconnu comme possesseur d'un site web.
- L'utilisation de l'API Google Sign In pour se connecter par le biais d'un compte Google permet au moteur de recherche de récupérer des informations relatives aux auteurs.

Figure 2-30

Section Liens de Google+ pour indiquer les sites web appartenant à un même auteur



L'AuthorShip a pour objectif de relier un maximum de comptes associés à un auteur ou à un compte afin d'identifier les actions de chacun mais aussi de valoriser les auteurs les plus actifs. Si l'affichage n'est plus présent dans les SERP, rien ne confirme dans les dires de John Mueller que les actions des auteurs n'intéressent plus la firme, surtout quand nous savons que Google souhaite à terme déterminer les intentions des utilisateurs (il lui faudra donc bien les connaître). Du point de vue des réseaux sociaux, il faut imaginer que Google peut s'intéresser à certains d'entre eux mais essentiellement aux principaux ou tout simplement ceux qu'il détient, notamment Google+, YouTube, Orkut (fermé en septembre) ou encore Picasa... Lorsque nous cliquons sur le lien *Modifier* de la section *Liens*, Google+ présente une liste par défaut de réseaux à connecter, nous pouvons supposer qu'il existe des partenariats ou qu'il respecte la logique des réseaux majeurs du marché. Nous retrouvons notamment Twitter, Facebook, Flickr, LinkedIn ou Quora et il nous suffit d'indiquer les liens vers nos profils ou comptes pour être lié.

La notion d'AuthorShip telle qu'elle a été inventée ne se limite pas du tout aux comptes sociaux associés à un compte Google via Gmail ou Google+. En réalité, le système est beaucoup plus puissant et intéressant qu'il n'y paraît car la force de l'AuthorShip est de pouvoir relier des pages web ou des sites à un compte « auteur » afin de valoriser encore davantage son activité et tous les efforts qu'il mène sur la Toile. Certes, Google n'affiche plus les informations dans les SERP mais rien ne confirme que ces interconnexions n'intéressent plus le moteur. Nous pouvons même imaginer qu'il a suffisamment engrangé de données depuis 2011 pour se permettre de changer de méthode et éviter le spam grandissant autour de l'AuthorShip.

Google joue avec les mots

Comme souvent, Google teste des fonctionnalités puis revient dessus quelques mois plus tard. L'AuthorShip initial fait partie de cette longue liste de tests qui n'ont pas résisté à la pression interne, au spam ou au désintérêt des internautes (...). Dans les faits, l'ancien marquage qui permettait d'afficher son nom et sa photo de profil Google+ au sein des SERP a disparu, mais nous devons tout de même lire entre les lignes et bien capter les messages subliminaux publiés dans les différents communiqués pour comprendre que les données des auteurs intéressent toujours le moteur de recherche. C'est davantage la forme qui a changé que le fond. Reste à savoir si cela aura désormais un impact sur le positionnement et si un potentiel AuthorRank sera appliqué, comme nous le verrons dans la section suivante...

Comme cela a été indiqué par Google, il est encore possible d'afficher des informations d'auteur à l'aide des rich snippets, notamment avec Schema.org. Sachez qu'il existe de très intéressantes ressources sur la Toile pour effectuer ces nouveaux marquages d'auteurs en HTML, notamment sur le site SEO Skeptic (source : http://goo.gl/QiZDp9). Nous allons voir deux exemples avec Schema.org pour illustrer nos propos, sachant que l'ancienne technique avec l'attribut `rel="author"` est toujours fonctionnelle si vous l'avez conservée ou déjà mise en place.

Vers un retour de l'ancien balisage ?

Le 1er octobre 2015, Gary Illyes, Trends Analyst à Google Dublin, a déclaré que les webmasters devraient maintenir l'ancien balisage de l'AuthorShip pour permettre à Google de l'utiliser si nécessaire (source : http://goo.gl/ploL0o). Ces propos ont provoqué l'étonnement de la communauté mondiale après l'arrêt du système un an auparavant annoncé par John Mueller. Gary Illyes a alors nuancé ses dires mais a confirmé que le balisage pouvait être laissé ou remis en place car Google pourrait le réutiliser en cas de besoin.

En d'autres termes, Google n'a peut-être toujours pas trouvé de solutions pérennes pour se détourner de l'AuthorShip original ou cherche encore à obtenir d'autres informations d'auteurs pour compléter sa base d'informations. Toutefois, rien ne permet de confirmer un retour de l'AuthorShip bien qu'aucun porte-parole de Google n'ait contredit la déclaration surprenante de Gary Illyes.

Dorénavant, il faut se rendre à l'adresse https://schema.org/author pour obtenir les types de balisage sémantique qui permettent de noter l'existence et le rôle d'un auteur au sein d'une page web. Nous constatons que la notion d'auteur est évoquée dans le type CreativeWork qui indique l'auteur d'un contenu.

Figure 2-31

Marquage des auteurs avec Schema.org et les microdonnées

Voici deux exemples qui montrent comment Google peut obtenir des informations sur un auteur en HTML. N'oubliez pas qu'une fois qu'il détermine qui est l'auteur avec le type CreativeWork (qui s'applique à de nombreux sous-types de marqueurs), il peut déduire que les autres données correspondent à cet auteur. L'AuthorShip n'a pas forcément toujours été l'unique méthode pour recueillir ces données personnelles et les liens entre des pages et des comptes Google.

```
<!-- Exemple 1 avec " CreativeWork " -->
<div itemscope itemtype="http://schema.org/CreativeWork">
    <img itemprop="image" src="creation.jpg" />
    <p>
        <span itemprop="name">Nom de la création</span>
        <span itemprop="author">Nom de l'auteur</span>
    </p>
</div>

<!-- Exemple 2 avec d'autres types de création -->
<body itemscope itemtype="http://schema.org/WebPage">
...
    <div itemscope itemtype="http://schema.org/Article">
        <p itemprop="name">Titre d'un article</p>
        <p>Rédigé par : <a href="page-auteur.html" itemprop=»author»>Nom de l'auteur</a> le
        <span itemprop="datePublished">10-07-2014</span></p>
    </div>
...
</body>
```

Le principal problème posé par l'AuthorShip concerne les sites associés à plusieurs auteurs. Dans ce cas, le site (ou blog) peut appartenir à un auteur en particulier qui fait intervenir d'autres personnes sporadiquement. Malheureusement, les premières techniques ne permettent pas de différencier l'auteur du blog de l'auteur de l'article alors que c'est parfois indispensable pour valoriser les actions de chaque auteur indépendamment et sans fausser les résultats de chacun. Historiquement, Google dissociait les relations entre Google+, l'auteur principal du site (ou l'éditeur) et les auteurs invités ou associés grâce à l'attribut `rel="me"` parallèlement à `rel="author"`. De nos jours, cela n'est plus possible sous cette forme et il suffit de mélanger plusieurs marquages sémantiques avec les microdonnées, les microformats ou RDFa pour obtenir un résultat similaire. Cela ne facilite pas la tâche mais retenez que la mise en place des multi-auteurs est toujours réalisable, bien que nous ne sachions pas si cela fonctionne réellement.

Comment différencier plusieurs auteurs dans une page ?

Google a toujours rencontré des difficultés à identifier les auteurs réels des contenus au sein d'une même page, notamment dans le cas du blogging anonyme. Il est fort probable que cela ait fortement dérangé les robots d'indexation et que ce manque de précision et ces problèmes d'identification soient une des raisons de la suppression de l'AuthorShip. Les marquages créés pour l'occasion n'étaient pas idéaux et ne permettaient sûrement pas d'obtenir les résultats escomptés. Cela a dû peser dans la balance au moment de clôturer l'histoire de l'AuthorShip.

Dans les faits, l'AuthorShip s'avère intéressant car il permet de connecter un maximum de pages et de profils à un auteur voire à une page professionnelle afin de valoriser toutes les actions menées. Bien qu'il

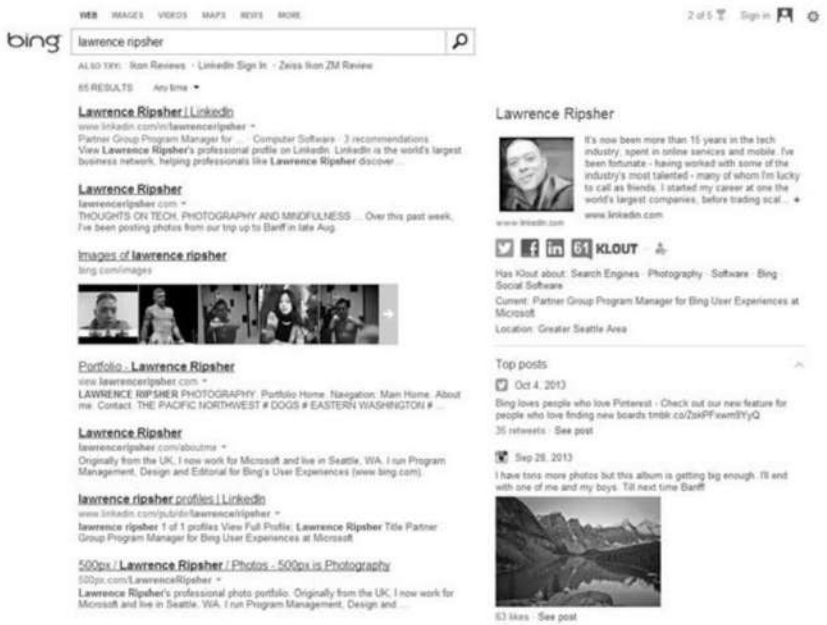
ait changé de visage sur Google, aucun communiqué officiel n'a confirmé un désintérêt pour les informations concernant les auteurs (seul l'affichage des données semble touché par la mutation du système). Nous allons voir dans la prochaine section que la notion d'AuthorRank est la conséquence directe de ces associations mais aussi que les rich snippets risquent à terme d'avoir de plus en plus de poids dans la détermination des liens entre auteurs et contenus.

Bing dispose également de son propre système de liaisons mais ce dernier se fait par l'intermédiaire du service en ligne Klout (source : http://goo.gl/xHu1JQ) depuis fin 2013. Il suffit de se connecter à l'outil Snapshot de Klout pour authentifier et relier les auteurs à leurs réseaux sociaux.

L'AuthorShip de Bing est beaucoup plus spécifique que celui de Google puisqu'il ne semble pas s'intéresser particulièrement aux sites web des auteurs. Il se cantonne uniquement aux réseaux sociaux principaux. La fin de l'AuthorShip natif de Google ne semble pas avoir atteint la détermination de Bing et nous pouvons espérer que Bing continue son association avec le service Klout pour valoriser les auteurs de contenus sur la Toile.

Figure 2-32

AuthorShip via Klout dans Bing aux États-Unis



PublisherShip

La mutation de l'AuthorShip de Google n'a pas causé que des malheurs puisque nous avons appris dans le même temps que l'association d'une entreprise à une page web pourrait fonctionner.

Dans l'intervention de John Mueller du 28 août 2014 sur Google+, qui a généré de nombreux commentaires de spécialistes, le porte-parole de Google Zurich a indiqué que l'attribut `rel="publisher"` de l'ancien AuthorShip était toujours en place. Sa mention exacte (en anglais dans le texte) indique que la

mutation du système de Google n'a affecté que les marquages d'auteur et non celui des éditeurs ou des entreprises : « publisher markup is not affected by this ». Nous pouvons retrouver toutes les techniques pour relier des pages professionnelles à des sites web dans la documentation officielle de la firme (source : http://goo.gl/mlQfdj).

Sur le même principe que l'AuthorShip classique, il est en effet possible de créer des interconnexions entre des pages web et des profils professionnels (ou pages) via l'attribut `rel="publisher"` comme dans l'exemple suivant :

```
<a href=https://plus.google.com/id-page-g+" rel="publisher">Google+</a>
```

Pour que l'interconnexion se fasse proprement, la firme conseille de rajouter l'URL du site dans la section Infos de la page Google+ associée au site web, en tant que lien principal. Cela permet de faire l'échange dans les deux sens et d'assurer la bonne appartenance des informations.

Auparavant, nous savions qu'il était possible d'utiliser également deux autres méthodes, mais rien ne dit que ces dernières soient encore activées dans le moteur de recherche :

- lien direct ajouté dans la section `<head>…</head>` :

```
<link rel="publisher" href="https://plus.google.com/id-page-g+/" />
```

- usage d'un badge Google ciblé vers une page professionnelle.

La documentation officielle indique désormais une nouvelle méthode pour relier un compte professionnel à un site web, à l'aide du service Google My Business (https://business.google.com). Il suffit de sélectionner la page Google+ relative à un site web, d'aller dans le « profil » de cette page (onglet *Page Google+* dans le menu latéral à gauche) puis de se rendre dans la section Infos. Ensuite, il ne reste qu'à entrer l'URL du site web associé ou de cliquer sur l'option *Associer le site web* si cela est proposé (cela change selon les cas). Normalement, une validation de cette association est demandée, il ne reste qu'à la confirmer pour finaliser la demande.

Le PublisherShip est un nom attribué par la communauté mais Google n'a jamais vraiment utilisé ce vocable. Il nous permet juste de le différencier de l'ancien AuthorShip. Auparavant, les sites professionnels étaient peu valorisés par Google mais cela semble avoir changé puisque la documentation officielle mentionne clairement l'intérêt de ce marquage sémantique.

- L'association entre Google+ et un site web permet d'entrer en contact avec nos relations professionnelles (ce qui est le but premier d'un réseau social en somme).
- Le PublisherShip permet à Google de déterminer la pertinence du site web par rapport à des requêtes d'utilisateurs.

En d'autres termes, cette mise à jour du système permet à Google de mieux ou moins bien classer les sites web qu'il connaît (via ces connexions) en fonction des requêtes des utilisateurs. De nombreux exemples trouvés sur le Web montrent même que le logo des pages professionnelles apparaît encore de temps à autre dans les SERP, ce qui signifie que ces extraits enrichis jouent toujours un rôle dans le référencement.

AuthorRank et AgentRank

La notion d'AuthorRank est discutée à tout va depuis de nombreux mois, elle résume le principe qui consiste à mieux positionner dans les SERP des pages relatives aux publications d'un auteur. En d'autres termes, plus un auteur reconnu par Google est actif sur les plates-formes sociales et les sites web, plus les pages web liées à son activité personnelle peuvent être mieux positionnées.

Initialement, le terme « AuthorRank » est une invention des spécialistes SEO qui parlaient d'une notation pour les auteurs actifs et réputés. Désormais, le terme est commun et s'emploient souvent mais il faut savoir que Google mentionne clairement l'AuthorShip mais rarement l'AuthorRank, pour ne pas dire jamais. Dans les faits, l'idée de noter les « agents » remonte à 2005 avec un premier brevet sur la notion d'AgentRank (source : http://goo.gl/WevuYm). Le brevet a évolué et s'est affiné jusqu'à sa dernière version connue et publiée le 22 octobre 2012 (source : http://goo.gl/orx2M5).

Partant de ce constat, l'AuthorRank est sujet à controverse car il n'a jamais été réellement confirmé par la firme, bien qu'un système de notation des « agents » puisse exister. En effet, Google a publié un brevet le 4 février 2014 intitulé « Reputation Scoring of An Author » dont le titre est évocateur (source : http://goo.gl/st96fL). Il s'agit d'un document qui mentionne le fait de revendiquer des contenus (via l'AuthorShip ou un système équivalent comme d'autres rich snippets ou le PublisherShip) et de noter les auteurs et contributeurs en fonction de leur réputation et de leur activité sur les sites web, notamment par la publication de nombreux contenus.

De nombreux spécialistes ont toujours évoqué l'AuthorRank comme la résultante d'une notation des auteurs actifs sur les réseaux sociaux mais si nous lisons les brevets de Google, l'activité sur les réseaux sociaux n'est pas vraiment mise en avant, il s'agit plutôt de l'ensemble des contenus attribués à un auteur, qu'ils soient publiés sur un blog, un site web classique, dans des fiches-produits ou sur des réseaux sociaux par exemple...

Le brevet intitulé « Separating Reputation of Users in Different Roles » publié le 17 février 2011 va un peu dans le même sens puisqu'il décrit un moyen de valoriser les internautes actifs (contenus, avis, commentaires...) sur des sites tiers tels qu'Amazon (source : http://goo.gl/BJStcl). Dans ce cas, nous pouvons imaginer l'équivalent pour les médias sociaux puisque ce sont aussi des sites tiers, mais qu'en est-il réellement ? Il est bien difficile de répondre car rien ne nous permet d'affirmer qu'une notation des auteurs existe vraiment lorsque nous sommes actifs sur les réseaux sociaux.

En fait, nous pouvons nous poser plusieurs questions à propos d'un quelconque intérêt pour Google de noter les auteurs actifs sur les réseaux sociaux.

- Pourquoi Google valoriserait-il les auteurs actifs sur des réseaux concurrents tels que Facebook, Twitter ou LinkedIn ?
- Comment Google pourrait-il connaître tous les réseaux sociaux du marché et valoriser les auteurs avec neutralité, partialité et égalité dans ce cas ?
- Comment Google pourrait-il techniquement reconnaître des auteurs identiques mais connectés avec des adresses e-mails différentes et portant des pseudos variés ?

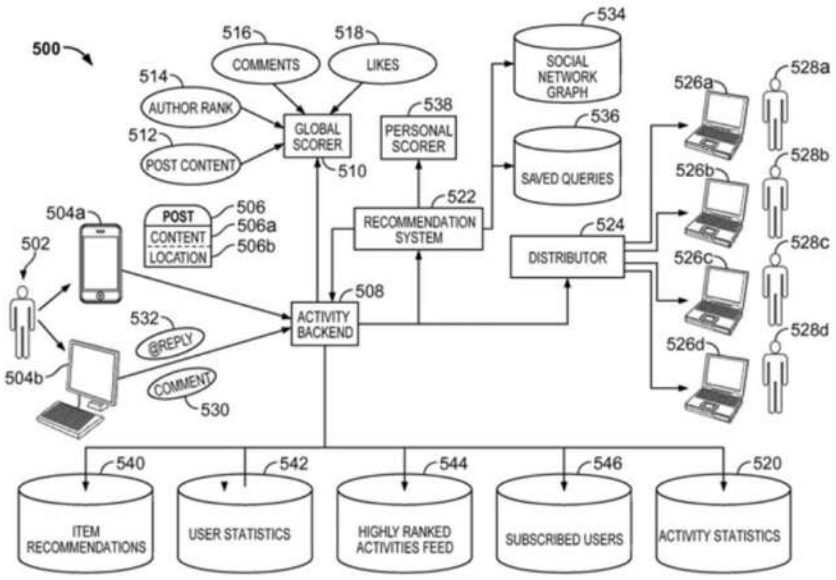
Il ne s'agit que de quelques questions menant à un raisonnement par l'absurde mais il est vrai que nous pouvons nous demander pourquoi Google aurait intérêt à valoriser des internautes actifs ailleurs que sur ces propres réseaux sociaux… Nous pouvons comprendre que la firme cherche à valoriser les auteurs en fonction de leurs interventions sur la Toile par le biais de contenus riches, et de ce fait, les publications sur les réseaux sociaux pourraient également avoir un intérêt, mais nous ne pouvons pas réellement savoir si un AuthorRank existe et si sa portée va jusqu'aux réseaux des concurrents (et si oui, lesquels ?)…

Tous les avis existent sur le Web car ce sujet n'a jamais vraiment été tranché de la part des porte-parole de Google, nous pouvons aisément imaginer que l'AuthorShip avait un intérêt puisque plusieurs témoignages démontraient que le fait de déconnecter des sites d'un profil Google+ les faisaient quelque peu chuter dans les SERP. Désormais, l'AuthorShip ayant disparu sous sa forme initiale sur Google, il faut imaginer que les rich snippets (voire le PublisherShip) vont prendre le relais pour développer cet éventuel AuthorRank. En revanche, aucune étude totalement fiable n'a permis d'affirmer qu'être actif sur Twitter ou Facebook pouvait avoir un impact direct sur le positionnement via un AuthorRank. En réalité, nous pouvons imaginer qu'il existe des effets indirects ou un effet boule de neige qui permet d'améliorer l'indexation, le netlinking et donc quelque peu le positionnement, mais rien ne permet de confirmer qu'il s'agit d'une note attribuée par le moteur de recherche…

Dans les faits, un seul brevet récent mentionne clairement l'AuthorRank, il s'agit d'un document publié le 10 décembre 2013 intitulé « Scoring Authors of Posts » dans lequel Todd Jackson et huit autres employés de Google mentionnent la notation des auteurs mais aussi le nom des réseaux sociaux Google+ et Twitter (source : http://goo.gl/IshVf7). Nous allons détailler ce qu'il en ressort mais nous pouvons enfin parler d'AuthorRank puisqu'il s'agit de la première mention officielle et d'un système correspondant à nos attentes. Il ne fait donc plus aucun doute depuis sa parution que Google a mis en place une méthode de notation des rédacteurs et des auteurs. Seule son application peut être remise en cause.

Figure 2-33

Première mention de l'AuthorRank dans le brevet « Scoring Authors of Posts » (numéro 514)



Ce brevet explique tous les facteurs pris en compte dans la notation des auteurs et indique par la même occasion que ce critère n'est pas spécifique aux réseaux sociaux mais bien à toutes les publications faites par des auteurs reconnus et affiliés à un compte Google. Voici la liste des éléments analysés et pris en compte par le système du moteur (source : http://goo.gl/S5bu7x) :

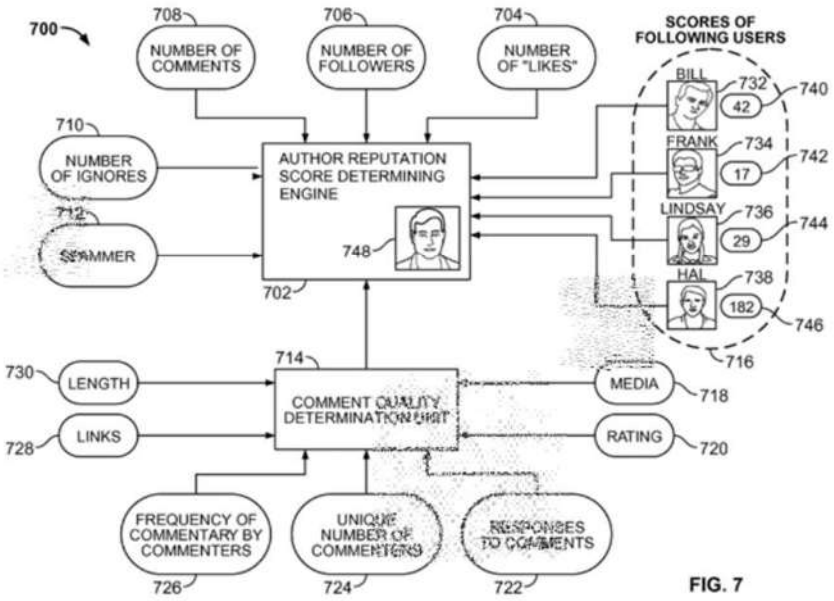
- nombre de commentaires obtenus pour la totalité des publications (réseaux sociaux, articles, fiches produits...) ;
- nombre d'abonnés et de contributeurs différents. Google compte les abonnés et les personnes qui commentent les contenus en ligne mais en prenant garde de vérifier qu'il doit s'agir d'un maximum d'internautes variés pour booster la note ;
- nombre de « Likes » ou « +1 » sur les publications. Ici, il s'agit nettement d'un critère social et même si le brevet parle de « Likes », le nom de Facebook n'est jamais mentionné et nous ne pouvons pas confirmer qu'il est pris en compte, au contraire de Twitter dont le nombre d'abonnés jouent un rôle ;
- notation en fonction des contributeurs qui interviennent sur nos publications et des abonnés qui nous suivent. En d'autres termes, Google comptabilise le nombre d'abonnés et de contributeurs mais analyse également leur note et leur influence. Si nous avons 1 000 abonnés inactifs, notre note sera mal évaluée tandis que si nous avons 500 abonnés dont une centaine de membres actifs et reconnus, notre note sera rehaussée. Le système est donc profond, il ne se limite pas à notre propre compte mais analyse également les comptes affiliés qui interagissent avec nous pour voir si nous méritons une meilleure note ou non. Une fois encore, Google a mis en place une forme de « note de confiance » puisqu'il suffit que des membres influents et réputés nous suivent et contribuent parfois à nos publications pour que notre note soit majorée ;
- note de qualité des interactions avec les autres membres. Ce point est très important et risque de faire la différence pour beaucoup d'entre nous. En effet, l'objectif n'est pas de compter uniquement les partages, les Likes ou les commentaires. Google va plus loin d'une part en différenciant les membres qui ont réalisé ces interactions et d'autre part, en scrutant en détail le contenu des commentaires. Nous allons revenir sur ce point ;
- nombre de publications « ignorées », c'est-à-dire masquées ou non suivies par les abonnés. Il s'agit d'une note « négative » qui vient contrebalancer la note positive obtenue par les actions des internautes. En d'autres termes, Google dévalue quelque peu la note finale de l'auteur en fonction du nombre de ses publications qui ont notamment été ignorées ou supprimées ;
- identification des spammeurs grâce à leur IP pour dévaluer leur note de contributeur. Google explique que les membres qui « aiment » tout, qui suivent tout et n'importe quoi, qui utilisent des phrases *spammy* comme « Comment devenir riche » ou « Super article » subiront une forte dévaluation de leur notation. L'objectif est d'éviter dès à présent le spam abusif et les commentaires sans intérêt qui polluent les blogs et les plates-formes sociales dans l'unique but d'obtenir des *backlinks*.

Comme nous venons de le voir, Google analyse en profondeur les interactions entre les auteurs. Cela s'explique car ces agissements ont un impact sur la notation de l'auteur « éditeur de contenu » mais aussi sur la notation des auteurs « contributeurs » qui laissent des commentaires. Google ne peut donc pas ignorer les avis et commentaires car ce sont également des « publications » au sens du moteur. Ils ont donc un rôle à jouer dans la note finale. Globalement, voici les points pris en compte et à respecter pour être un bon contributeur et obtenir une meilleure note :

- l'utilisation du caractère @ pour mentionner un membre (une réponse à un avis) a un rôle primordial puisque Google sait reconnaître les auteurs affiliés si nous respectons la forme `@pseudo`. Plus une publication reçoit de réponses avec une telle mention, plus les auteurs concernés sont valorisés ;
- plus une publication reçoit d'avis et de partages de contributeurs différenciés, plus la note est rehaussée pour l'éditeur des contenus ;

Figure 2-34

Critères pris en compte pour le calcul de l'AuthorRank de Google



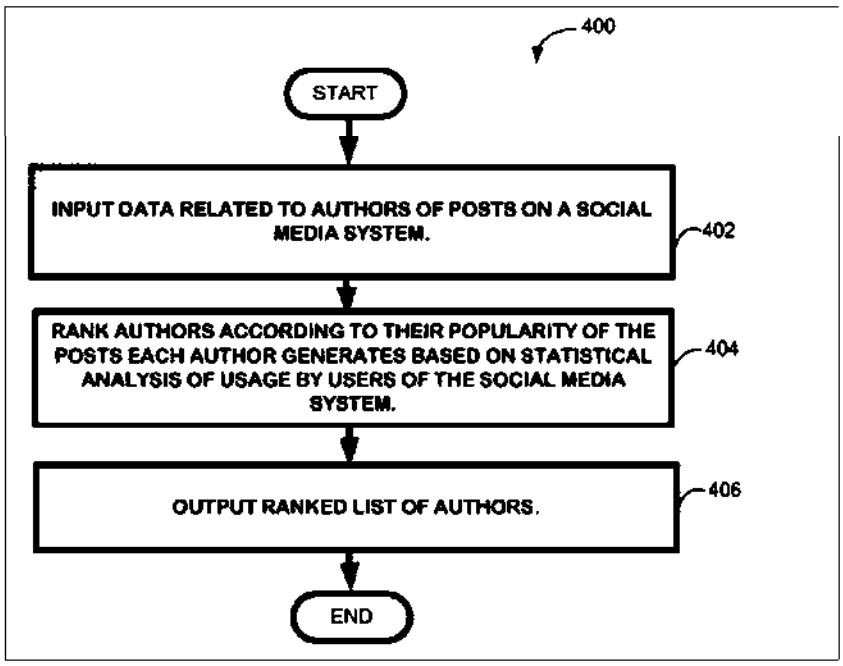
- plus une discussion s'installe entre des contributeurs identifiés, plus la note est valorisée. Par exemple, si une publication se transforme en questions-réponses entre l'auteur et un membre, l'auteur sera mieux considéré car il interagit et s'investit avec ses membres. Il faut toutefois être prudent et s'assurer que Google puisse bien comprendre qu'il s'agit de discussion, c'est là où le rôle du `@pseudo` est important, par exemple ;
- la fréquence des commentaires ainsi que leur contenu sont étudiés en détail pour surévaluer ou sous-évaluer la note finale :
  - la longueur des commentaires impacte la note finale. Un simple « très bon article » n'a aucun poids alors qu'un commentaire constructif est valorisé ;
  - un commentaire à forte plus-value (vidéo, illustration, lien vers une ressource de valeur...) est mieux noté pour le contributeur (donc sa note sera augmentée) et pour l'éditeur de la publication commentée ;
- les notes des commentateurs impactent la note définitive de l'auteur ;
- les interactions entre les contributeurs et l'auteur ont également un impact. Cela passe par les réponses mais aussi par le fait qu'un contributeur soit abonné ou non. Par exemple, s'il s'agit d'un visiteur non abonné, la note est encore rehaussée car elle fait intervenir un nouveau membre. Cela signifie pour Google que la publication a joui d'une bonne visibilité et d'une plus-value importante pour que des lecteurs non affiliés s'y réfèrent ;

- le fait qu'une publication ait été partagée par d'autres membres et que ces partages aient suscité des interactions avec d'autres membres augmentent la note de l'auteur initial. Par exemple, si nous postons un statut sur Google+, que ce dernier est partagé par un visiteur et que son partage obtient des commentaires ou Likes, Google valorise l'auteur initial du contenu partagé.

Parallèlement, il semblerait que Microsoft ait également mis au point un système entièrement consacré à la notation des auteurs en fonction de leur activité sur les réseaux sociaux. En effet, un brevet intitulé « Ranking Authors in Social Media Systems » et publié le 10 mai 2012 (source : http://goo.gl/Yx4pw2) décrit un système de notation des internautes actifs sur les diverses plates-formes communautaires via des calculs statistiques.

Figure 2-35

Description d'un système de notation des auteurs sur les réseaux sociaux par Microsoft



Le brevet explique que l'autorité des internautes peut être calculée grâce à ces points précis :

- analyse temporelle des partages de liens dans laquelle l'autorité est calculée sur la base de la propension de l'utilisateur à fournir des liens innovants vers des pages web qui deviennent par la suite rapidement populaires ;
- calcul basé sur les auteurs de liens et de mises à jour de contenus dans des domaines pour lesquels ils font autorité ;
- popularité et influence basées sur les liaisons et relations entre auteurs. Par exemple, des indicateurs tels que le nombre d'abonnés, de publications, de *retweets*, de mentions ou le nombre d'amis en ligne peuvent jouer sur la notation finale.

Sachez que Yahoo! a également déposé un brevet le 16 octobre 2012 pour valoriser les contributeurs actifs sur les sites web (source : http://goo.gl/sE8K1c), wikis et forums qui exploitent les *User Generated Content* (UGC). Ainsi, Yahoo! pourrait valoriser des auteurs reconnus par des métadonnées et qui contribuent

beaucoup sur ces types de sites web. Toutefois, rien ne dit si le système existe réellement et s'il est en place sur le moteur de recherche…

En définitive, Google, Microsoft, Yahoo! voire d'autres moteurs de recherche commencent petit à petit à développer leur système de notation des internautes (auteurs, contributeurs, rédacteurs…) mais cela ne se limite pas uniquement à certains réseaux sociaux dans la majorité des cas, notamment chez Yahoo! et Google.

Globalement, nous devons être actifs sur les sites web (blogs notamment) mais aussi sur les réseaux sociaux. Il convient de commenter les publications en apportant de la plus-value sans oublier de créer des interactions avec les autres contributeurs et les auteurs par le biais d'abonnements, de partages et de réponses.

Le système semble plutôt convaincant et il ne reste qu'à voir les résultats sur le long terme pour savoir si ce nouveau facteur de notation sera pleinement appliqué et aura un fort impact sur le positionnement, que ce soit sur Bing, Yahoo! ou Google.

Vers une disparition prochaine de l'AuthorRank ?

À ce jour, et notamment depuis les modifications récentes de l'AuthorShip, de nombreux spécialistes pensent que l'AuthorRank devrait disparaître (s'il existe et est vraiment appliqué) mais les divers commentaires de John Mueller et Gary Illyes ont laissé entendre que la fin de l'AuthorShip originel ne signait pas pour autant l'arrêt de mort du suivi des auteurs, ne soyons donc pas si catégoriques… Qui plus est, la mise en avant discrète du PublisherShip prouve que les sites web professionnels pourraient connaître un second souffle sur Google+ en matière de SEO, ce qui signifierait qu'un éventuel AuthorRank pourrait être assimilé à cette nouvelle fonctionnalité. Seul l'avenir nous le dira…

Créer un système de hashtags optimisé SEO avec PHP

L'objectif de ce chapitre est de vous montrer comment optimiser de nombreux critères de référencement, mais nous pouvons aussi nous intéresser à des cas plus rares mais néanmoins intéressants. Dans cette section, nous allons tenter de réaliser un système de *hashtags* optimisé pour le référencement, bien que la technique puisse être apparentée à du *Black Hat SEO* à cause d'un critère technique.

Les principaux problèmes des systèmes de tags ou hashtags sont de générer une page de résultats variables en fonction du mot-clé cliqué, mais aussi d'ajouter une multitude de liens au sein des pages qui vont diviser le jus de liens transmis aux pages réellement intéressantes. Pour contrecarrer en partie ces soucis, quatre alternatives sont possibles :

- réaliser le système en Ajax car ce langage peut s'avérer bloquant pour les robots, ce qui évitera l'indexation de contenus dupliqués dans les pages de résultats (nous reviendrons en détail sur ce point dans le prochain chapitre) ;
- établir des liens classiques vers les pages de tags afin de lister les résultats correspondants, puis ajouter une règle dans un fichier `robots.txt` voire dans un fichier `.htaccess` pour bloquer la page de résultats et donc éviter l'indexation des contenus dupliqués ;
- transformer les hashtags en boutons HTML (balises `<input type="button" />` de formulaire, par exemple) pour bloquer la lecture des robots et éviter la perte de jus de liens pour les autres liens hypertextes contenus dans la page ;

- générer des liens cliquables qui ne sont pas réellement des liens `<a>…</a>` en HTML, c'est-à-dire qu'une fonction JavaScript va agir pour rendre les hashtags cliquables en empêchant l'indexation de la page de résultats, sans bloquer la lecture des contenus pour les robots et enfin en permettant une division plus judicieuse du jus de liens.

Il est important de partir du bon pied pour bien comprendre comment fonctionne un système de hashtags conçu en PHP. Comme les codes présentés ici sont pour la plupart réalisés avec la méthode procédurale, les fonctions suivantes resteront basées sur ce modèle. Cependant, il est tout à fait possible voire plus pratique de transformer ce système en PHP orienté objet.

Pour mettre en place un système de hashtags, il faut lire les textes au moment de l'affichage ou lors de l'envoi dans la base de données et détecter les hashtags présents. Pour cela, il suffit de créer une fonction qui recense toutes les occurrences de la forme `#hashtag`. Une fois ce code créé, il doit être appliqué systématiquement lors de l'affichage des pages (en récupérant les contenus à partir d'une base de données, par exemple) ou directement lors de l'ajout des données dans la base (dans ce cas, le code des hashtags est entré en « dur » directement, ce qui limite les traitements pour le visiteur).

Créer un système de mentions

Le même principe que tout ce qui va suivre pourrait s'appliquer pour les mentions comme nous les trouvons sur les réseaux sociaux tels que Twitter avec `@utilisateur`. Au fond, seul le caractère de départ change : il ne s'agit plus du caractère `#` mais de `@` ou `+` sur Google+, par exemple.

La méthode de détection des hashtags résulte d'une expression régulière précise couplée à la fonction `preg_match_all()` ou plutôt `preg_replace()` afin de modifier le hashtag par ce même mot-clé, cliquable cette fois-ci. La fonction propre mais moins optimisée SEO ressemble à la suivante :

```
function replaceHashtags($texte = '', $url = '') {
$regex = "/#(.*)([ ]|[!\"#$%'&()*+,.\/:;<=>?@\_`{|}~-])+/iU";
$replace = '<a href="'.$url.'?hash=$1">#$1</span>$2';
$texte = preg_replace($regex, $replace, $texte);
return $texte;
}
```

Pour chaque hashtag, nous obtenons un résultat HTML qui ressemble au code suivant :

```
<a href="traitement.php?hash=hashtag">#hashtag</a>
```

La fonction prend deux paramètres (dont un optionnel) :

- le texte à analyser, qui peut être le contenu envoyé dans la base de données avec une requête SQL ou encore le texte affiché dans la page pour le visiteur ;
- le nom de la page (ou l'URL) du fichier de traitement, qui permettra d'afficher les résultats correspondant au hashtag cliqué (ici, il s'agit de `traitement.php`). Il suffit de laisser ce second paramètre vide pour que le traitement soit attendu dans la page en cours, ce qui est parfois le plus pratique…

La fonction s'utilise donc de cette manière :

```
<?php
// $texte correspond au contenu à traiter ou à afficher
// Il s'agit souvent de variables provenant d'une base de données
$texte = "Texte avec #hashtag par #milliers…";
echo replaceHashtags($texte, 'traitement.php');
?>
```

Il ne nous reste plus qu'à voir la page de traitements des résultats pour terminer notre système de hashtags. Dans notre exemple, cette page s'appelle `traitement.php` mais elle pourrait porter un nom plus évocateur voire subir une réécriture pour être plus optimisée.

En réalité, nous pouvons faire à peu près ce que nous voulons dans la page de résultats. Il faut aller chercher dans la base de données les résultats qui répondent aux hashtags cliqués. Cela fonctionne donc comme un moteur de recherche interne en quelque sorte. Nous n'aurons qu'à afficher les données qui nous intéressent, comme bon nous semble.

Il existe de nombreuses méthodes pour faire des requêtes de recherche, la plus connue étant la méthode `LIKE` en SQL. Dans notre cas, nous allons utiliser une méthode peu usitée mais pourtant efficace, appelée `REGEXP`, qui utilise des expressions régulières dans les requêtes SQL, ce qui peut être intéressant pour notre code. La requête suivante récupère les hashtags à la volée à l'aide du paramètre `GET` passé dans l'URL de traitement (`hash` dans l'exemple) :

```
// Récupération du paramètre d'URL hash
$word = htmlspecialchars($_GET['hash']);
// Requête sélective des résultats correspondant au hashtag cliqué
$query = 'SELECT colonneSQL FROM tableSQL WHERE colonneSQL REGEXP "#'.$word.'"';
// Ensuite, nous réalisons le traitement comme bon nous semble…
```

Nous pouvons par exemple lister les titres ou les extraits de texte qui répondent au hashtag cliqué dans la page de résultats. La fonction suivante est compatible jusqu'à PHP 5.5 et vous permet de faire rapidement un traitement de ce type, en sachant qu'il faudra la modifier à votre guise pour obtenir le résultat escompté :

```
// 4 paramètres obligatoires
function resultsHashtags($word = '', $connexion, $table, $column) {
$requeteSelect = mysqli_query($connexion, 'SELECT '.$column.' FROM '.$table.'
       WHERE '.$column.' REGEXP "#'.$word.'"');
while($result = mysqli_fetch_assoc($requeteSelect)) {
    echo $result[$column]."<br/>";
}
}
// Il suffit de la lancer ainsi pour la rendre fonctionnelle
if(isset($_GET['hash'])) {
resultsHashtags($word, $connexion, $tableSQL, $colonneSQL);
}
```

Faire tourner le système en boucle comme sur Twitter

Si vous voulez faire perdurer le système, il faudra également utiliser la fonction `replaceHashtags()` dans cette page (si vous l'utilisez au moment de l'affichage) pour que les hashtags présents soient également cliquables, et ainsi de suite…

Ce système permet donc de générer des hashtags à la volée lors de l'envoi des données ou au moment de l'affichage, puis d'afficher les informations correspondantes dans une page de résultats une fois ces tags cliqués. Ce premier système est propre mais n'est pas le plus optimisé pour le référencement. Il faudrait passer par une technique proche du Black Hat SEO pour obtenir de meilleurs résultats. En effet, notre première fonction génère des liens hypertextes classiques en HTML. Nous allons donc uniquement modifier cette dernière et ajouter une subtilité pour contourner le problème à nos risques et périls…

La technique consiste à ne pas générer des balises `<a>` mais plutôt des `<span>`, neutres en HTML mais que l'on va rendre cliquables grâce à une simple astuce en JavaScript. En effet, nous pouvons ajouter l'événement JavaScript `onclick` dans tous les éléments HTML existants et faire des redirections grâce à la commande `window.location.href`. En couplant ces facteurs, nous allons modifier la fonction `replaceHashtags()` pour créer des `<span>` cliquables qui éviteront le perte de PageRank et BrowseRank dans les pages de contenus.

La fonction s'utilisera de la même manière que la première version, sauf que nous ajoutons ici un troisième paramètre pour créer une classe CSS afin de donner un rendu visuel assimilable à un vrai lien pour les balises `<span>`. Voici la fonction modifiée :

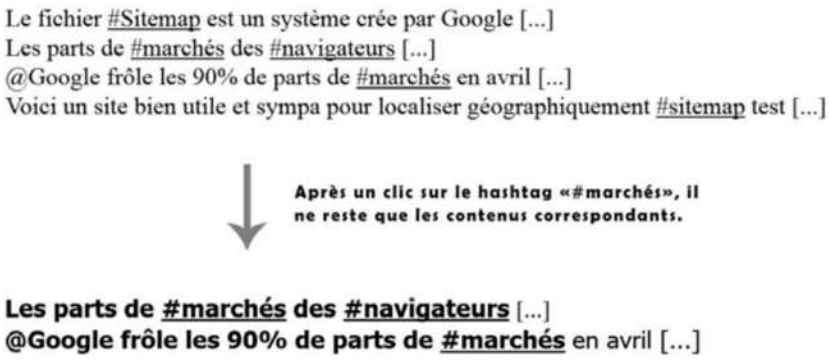
```
function replaceHashtags($texte = '', $url = '', $class = "classLink") {
$regex = "/#(.*)([ ]|[!\"#$%'&()*+,.\/:;<=>?@\_`{|}~-])+/iU";
$replace = '<span onclick="window.location.href=\''.$url.'?hash=$1\'"
class="'.$class.'">#$1</span>$2';
$texte = preg_replace($regex, $replace, $texte);
return $texte;
}
```

Il suffit ensuite de spécifier un style aux faux liens en `<span>` grâce à la classe CSS ajoutée (appelée `classLink` par défaut). Par exemple, le code CSS suivant met des liens soulignés en noir, puis sans soulignement au survol de la souris.

```
.classLink {
text-decoration :underline;
color:#000;
cursor:pointer;
}
.classLink:hover {
text-decoration:none;
}
```

Figure 2-36

Système de hashtags avec cliquables et CSS



Retenons que ce système fonctionne parfaitement quelle que soit la méthode utilisée. D'un côté, la technique propre facilite l'indexation des contenus grâce aux liens classiques et, d'un autre côté, la version Black Hat SEO optimise davantage la transmission du jus de liens. Ce qu'il faut retenir, c'est que la page de destination *(landing page)* qui contient les contenus relatifs au hashtag cliqué ne doit pas générer trop de contenus dupliqués, ce qui reste compliqué dans la majorité des cas. C'est pourquoi la gestion de son indexation (désindexation ou gestion d'une URL canonique, par exemple) représente le point crucial pour obtenir de meilleurs résultats.

Ce système n'est pas unique et n'est pas forcément le meilleur ou le plus efficace du marché, mais il était important de développer l'idée d'un mécanisme de tags, mentions ou hashtags optimisé pour le référencement. Nous allons désormais aborder la fin de ce chapitre avec quelques techniques qui pourront peut-être nous aider pour réaliser un référencement encore plus abouti.

Optimiser le Rank Sculpting et le Bot Herding

PageRank Google

Le PageRank est un critère utilisé par Google pour calculer la popularité d'une page web et donc son classement dans les pages de résultats. Il s'agit d'une note fixée entre 0 et 10 et attribuée par le moteur à chaque page web pour sa popularité. Ce point est important, ce sont bien chaque page de manière indépendante qui obtienne un PageRank donné, et non le site au complet !

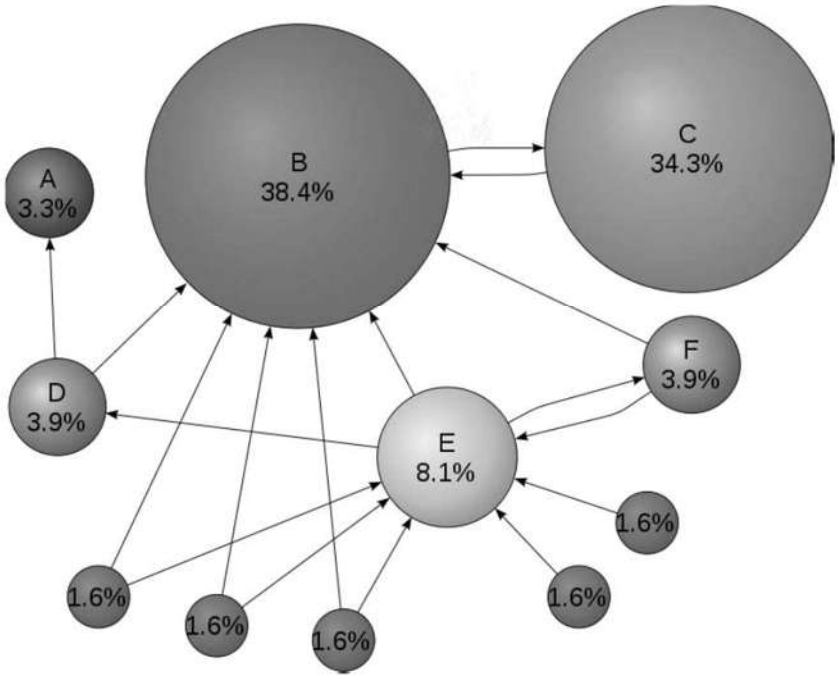
Le PageRank a été inventé par Larry Page et analyse plus d'une centaine de variables pour attribuer cette note finale, dont voici quelques exemples :

- quantité et qualité des liens entrants et sortants ;
- ancres de liens ;
- trafic, popularité et notoriété de la page ;
- comportement des internautes…

Ce qu'il faut absolument comprendre c'est que les liens n'ont pas la même valeur pour Google. Plus la source est pertinente et de qualité (avec un bon trafic, une forte notoriété, un PageRank déjà important), plus le lien sera de qualité et aura un poids dans le calcul final. Le PageRank est mouvant et il est réactualisé plusieurs fois par an au fil des modifications effectuées sur les sites web.

Figure 2-37

Illustration mise à disposition dans le domaine public par son auteur, 345Kai, dans le projet anglais Wikipedia



La formule initiale du calcul du PageRank a été donnée jadis par l'université Stanford dans un document intitulé « The Anatomy of a Large-Scale Hypertextuel Web Search Engine… »

```
PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))
```

La formule originelle est certainement bien différente aujourd'hui puisque Google a ajouté de nombreux facteurs complémentaires dans le calcul du PageRank comme la qualité des liens ou encore la variation des ancres, le nombre de nofollow, etc.

Nous devons retenir l'essentiel, à savoir que les liens constituent encore un élément fondamental du positionnement mais qu'il ne faut absolument pas fonder tout son travail sur ce seul et unique critère si nous ne voulons pas faire d'erreur, notamment depuis l'arrivée fracassante des pénalités de Google Penguin.

Différences entre liens externes et internes

Retenons également que Google différencie nettement les liens obtenus via des sites externes ou par le biais de liens internes. Les deux jouent un rôle pour la note finale du PageRank mais leur valeur est pondérée selon le type de lien dont il s'agit.

TrustRank : indice de confiance

Le TrustRank est un indice de confiance qui a vu le jour dès mars 2004 dans un article rédigé par un duo de chercheurs de l'université de Stanford intitulé « Combating Web Spam With TrustRank ». L'objectif du TrustRank est d'attribuer une note de qualité (ou confiance) aux liens entrants obtenus par les sites web. Ainsi, ce n'est plus seulement le nombre de liens qui est pris en compte mais bien leur qualité intrinsèque. Il est devenu de plus en plus important d'obtenir des liens sûrs…

Le TrustRank était une note allant de 0 à 1, attribuée par des humains chez Google selon une batterie de critères définis. Aussi, les sites les mieux considérés partaient avec un fort TrustRank et cela se rétribuait indéfiniment en fonction des échanges de liens naturels ou non. En réalité, la marque TrustRank n'existe plus depuis le 29 février 2008 car Google l'a tout bonnement abandonnée comme le confirme la figure 2-38.

Cela ne signifie pas pour autant que l'indice de confiance est tombé aux oubliettes. Nous pouvons presque nous assurer du contraire d'ailleurs car Google Penguin est capable de dissocier des liens de mauvaise ou de bonne qualité. Il existe donc encore un référentiel ou un algorithme qui intègre cette notion de confiance, mais elle est directement intégrée dans le PageRank actuel.

Il est difficile de savoir si des liens ou des sites web sont de qualité ou non mais l'AlexaRank, même s'il est indépendant des moteurs, peut s'avérer une bonne source de confiance pour savoir s'il est risqué ou non de faire des échanges de liens avec certains sites. C'est l'une des seules méthodes qui nous permet de nous rassurer pour avoir une approche de l'ex-TrustRank.

Figure 2-38

Abandon de la marque TrustRank par Google

TARR Status | ASSIGN Status | TDR | TTAB Status *(Use the "Back" button of the Internet Browser to return to TESS)*

TRUSTRANK

Word Mark	TRUSTRANK
Goods and Services	(ABANDONED) IC 042. US 100 101. G & S: Computer services, namely organizing information, sites and other resources available on computer networks
Standard Characters Claimed	
Mark Drawing Code	(4) STANDARD CHARACTER MARK
Serial Number	78588592
Filing Date	March 16, 2005
Current Filing Basis	1B
Original Filing Basis	1B
Published for Opposition	December 6, 2005
Owner	(APPLICANT) Google Inc. CORPORATION DELAWARE 1600 Amphitheatre Parkway Mountain View CALIFORNIA 94043
Type of Mark	SERVICE MARK
Register	PRINCIPAL
Live/Dead Indicator	DEAD
Abandonment Date	February 29, 2008

Chez Bing, un système équivalent au PageRank existe avec le BrowseRank, qui intègre lui aussi la notion de qualité des backlinks. Il faut donc employer les mêmes méthodes pour valoriser sa note et son positionnement à l'aide des critères off page. Pour Bing, les notes sont essentiellement fondées autour du BrowseRank pour les liens et du StaticRank pour juger les textes (source : http://goo.gl/mXSotJ), dont la qualité orthographique des documents, ce qui donne une note moyenne de qualité.

Qu'est-ce-que l'indice UPR ?

Il arrive également d'entendre parler de la notation UPR chez Microsoft. Elle rassemble les notions d'indice de confiance et d'indice de popularité prises en compte par le BrowseRank.

Ce qu'il faut retenir du TrustRank et des autres indices de confiance, c'est qu'il est désormais primordial d'obtenir des liens en masse, mais surtout de cibler leur qualité avant toutes choses sous peine d'être pénalisé par Google Penguin notamment. Le PageRank et le BrowseRank sont des algorithmes avancés qui savent très bien déterminer les liens de piètre qualité. Il convient donc de les limiter au maximum pour obtenir un profil valorisant pour les sites web à positionner.

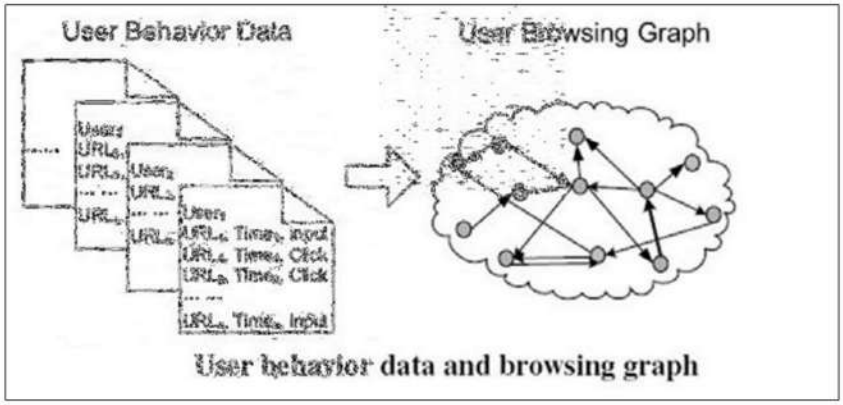
BrowseRank de Bing

La technologie de recherche de Microsoft est axée autour du BrowseRank (source : http://goo.gl/rdxuqr), un algorithme créé dès 2008 et qui reprend globalement les grandes lignes du PageRank de Google, à quelques différences près. En effet, Microsoft avance que Bing traite mieux les résultats avec le BrowseRank que Google et son PageRank car il prend en compte des critères comportementaux et relatifs à la qualité des liens :

- nombre de liens entrants ;
- qualité des liens entrants en fonction de la thématique abordée par la requête ou encore selon le poids attribué à certains liens plutôt qu'à d'autres ;
- taux de rebond dans les pages ;
- nombre de clics sur les liens entrants ;
- temps moyen de visite.

Figure 2-39

Schéma technique du BrowseRank



L'ensemble de ces facteurs permet de mieux valoriser les liens et les pages web en fonction de leurs réelles qualités. Si le PageRank et Google ont de nos jours développés ce genre de pratique, il faut avouer que le BrowseRank initial offre un panel assez intéressant et les résultats proposés par Bing semblent souvent pertinents.

Rank Sculpting et Bot Herding

Le PageRank et le BrowseRank sont des critères importants basés sur le maillage des liens externes mais également internes, ce qui signifie qu'il peut être opportun de bien organiser ses contenus et surtout ses liens internes pour favoriser le transfert du jus de liens. Le *PageRank Sculpting* (communément appelé ainsi grâce au succès légendaire du PageRank de Google) est la conséquence de cette idée. La méthode consiste à optimiser le maillage interne pour que les pages à fort potentiel récupèrent plus de jus de liens.

Le Rank Sculpting consiste donc à utiliser à bon escient le potentiel de popularité des pages pour favoriser les pages web secondaires ou profondes qui ont davantage de mal à obtenir des backlinks. Cette technique a longtemps été appliquée par les référenceurs mais les moteurs n'aiment pas spécialement être dupés de la sorte. Il faut donc veiller à créer un maillage interne optimisé et le plus naturel possible. Ne perdez jamais de vue que la réussite d'une bonne architecture interne de site web présente avant tout un avantage pour les visiteurs, les robots doivent absolument passer au second plan sous peine de se tromper de cible…

PageRank Sculpting vs Bot Herding ?

Il nous arrive parfois de parler de *Bot Herding* dont le résultat est très approchant du PageRank Sculpting dans les faits. Le terme « herding » signifie « mener un troupeau » en anglais, la technique est donc utilisée pour amener les robots d'indexation sur les pages qui nous intéressent. Le rôle du Bot Herding est un peu plus vaste que celui fixé par le PageRank Sculpting puisque c'est la gestion du maillage et de l'ergonomie interne qui est mise à contribution pour améliorer le crawl, ce n'est pas seulement pour un objectif de transfert interne de jus de liens.

Historiquement, le PageRank Sculpting se travaillait à l'aide de l'attribut `rel="nofollow"` que l'on plaçait dans les liens internes (balises `<a>…</a>` en HTML pour rappel) car ce dernier indiquait aux robots qu'il ne fallait pas suivre les liens ni leur transmettre du jus de liens. Désormais, la donne a changé puisque l'attribut `rel="nofollow"` a perdu de sa superbe. Les robots suivent les liens que l'attribut soit présent ou non, mais il semblerait que le PageRank ne soit pas transmis.

Nous ne pouvons pas garantir que le PageRank soit transmis lorsqu'un attribut `nofollow` est placé dans une balise de lien mais qu'en serait-il d'un éventuel TrustRank ou BrowseRank ? Après plusieurs tests, il s'avère que certains sites dont les liens entrants sont majoritairement des nofollow arrivent à obtenir un PageRank convenable et un positionnement de qualité. Certes, d'autres facteurs sont pris en compte et rien ne permet d'affirmer qu'il s'agirait d'un transfert de popularité, mais cette éventualité peut s'envisager. Peut-être que Google dévalue un peu la note mais il attribue tout de même un peu de PageRank aux pages cibles.

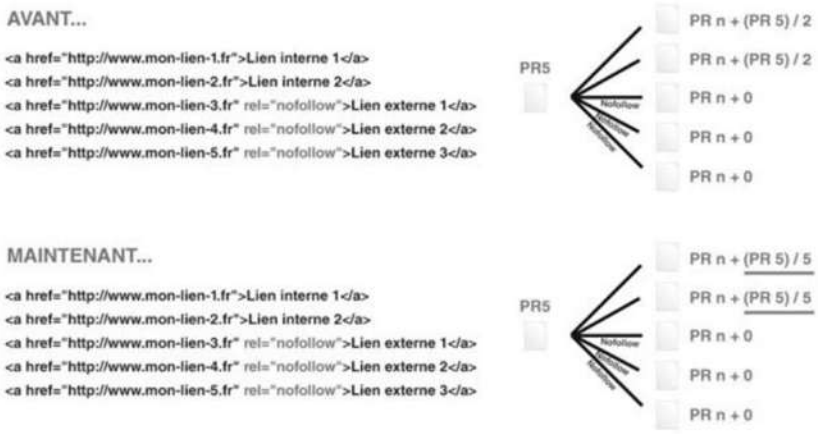
Qu'en est-il vraiment du rôle des nofollow ?

Google ne communique pas vraiment à ce sujet ou reste évasif car chaque déclaration pourrait faire l'effet d'une bombe. En laissant l'idée que l'attribut `rel="nofollow"` empêche le transfert de PageRank, la firme s'assure de ne pas être noyée par des spammeurs. Si elle avouait le contraire, elle modifierait le comportement des référenceurs du monde entier. Il est fort probable que nous ne sachions jamais vraiment le vrai du faux…

L'attribut `rel="nofollow"` était pratique puisqu'il permettait de « diviser » le PageRank en fonction des liens internes qui le possédait ou non, ce qui permettait de transmettre davantage de jus de liens aux pages que nous souhaitions optimiser. En d'autres termes, si une page contenait dix liens dont trois en nofollow, le PageRank des sept liens restants était de 1/7e et non de 1/10e. De nos jours, Google lutte contre cette pratique et divise la note de popularité en fonction du nombre de liens, qu'il existe des nofollow ou non. En reprenant notre exemple, cela signifie que la note pour chaque page serait de 1/10e mais pour celles qui sont ciblées par un lien avec `rel="nofollow"`, la note transmise sera de 0 tout simplement. Par conséquent, la valorisation du PageRank par le maillage et l'usage des nofollow ne présentent plus d'intérêt particulier…

Figure 2-40

Nouvelle interprétation de la transmission du jus de liens pour Google



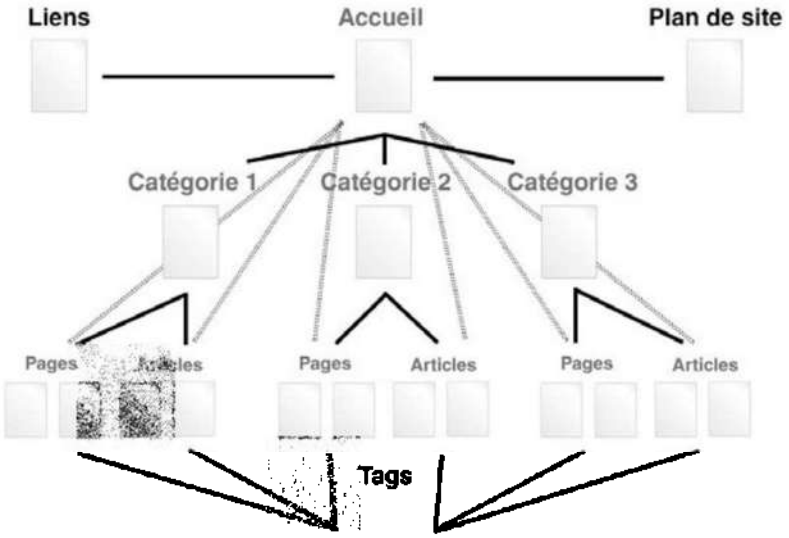
Est-ce pour autant la mort du Bot Herding ou du PageRank Sculpting ? Dans la majorité des cas, cette mise à jour de l'interprétation des attributs `rel="nofollow"` par Google constitue un réel frein, il faut donc passer par d'autres moyens pour réaliser ce type de technique.

Méthode du siloing

La méthode du *siloing* consiste à organiser de façon structurelle et ergonomique des catégories et des pages en « silos ». Les robots sont donc « dirigés » au sein des sites en respectant une logique de navigation et d'indexation tout en évitant les pages « cul de sac » *(dangling pages)*. La figure 2-30 décrit une forme de siloing pour WordPress.

Figure 2-41

Exemple d'architecture en silos optimisée pour WordPress



- Limiter l'indexation de certaines pages notamment avec un fichier `robots.txt` pour favoriser le crawl des pages majeures et à valoriser.
- Éviter à tout prix le problème du DUST avec les adresses web doublonnées. Les contenus recopiés et les URL dupliquées peuvent subir des sanctions mais aussi diviser encore plus la note de popularité. Il serait dommage de donner du jus de liens inutilement pour des URL doublons…
- Utiliser des facteurs bloquants au profit du référencement. Nous évitons souvent d'utiliser des codes en JavaScript, Ajax ou ActionScript (Flash) car ils bloquent le crawl des robots. Mais pourquoi ne pas les utiliser pour optimiser le maillage interne ?

Si votre site est bien conçu, avec un plan de site réussi et une bonne organisation, l'indexation a de fortes chances de bien se passer. Dans ce cas, l'usage de quelques facteurs bloquants lorsque c'est opportun peut permettre de limiter le transfert de PageRank vers des pages secondaires.

Prenons un exemple simple et concret d'un site e-commerce : il convient d'avoir des liens en bas de pages qui mènent vers les conditions générales de ventes (CGV), la page Partenariats, Revue de presse, etc. Est-ce que toutes ces pages méritent d'obtenir un fort PageRank ? Pas nécessairement, et elles prennent également de la place dans le site web. Peut-être serait-il intéressant de créer des listes déroulantes cliquables ? En effet, elles sont bloquantes pour les robots et surtout elles ne constituent pas des liens hypertextes au sens propre.

Aussi, le jus de liens ne leur serait pas transmis mais serait surtout moins divisé qu'auparavant sans pour autant gêné réellement l'ergonomie et l'efficience du site pour les usagers. Le code suivant, en JavaScript et HTML, est un exemple très optimisé de ce qui peut se faire pour bloquer les robots :

```
<!-- Script à placer dans la section <head> ou <body> pour gérer les redirections -->
<script type="text/JavaScript">
```

```
function changeMenu(nameFormulaire, url, extension) {
    document.forms[nameFormulaire].action = url+extension;
    document.forms[nameFormulaire].submit();
}

</script>
<!-- Code HTML de deux listes déroulantes optimisées SEO -->
<form name='formulaire'>
<select>
    <option value="conditions" onClick="changeMenu('formulaire', this.value, '.html')">CGV
    </option>
    <option value="aide" onClick="changeMenu('formulaire', this.value, '.html')">Aide
    </option>
    <option value="mentions-legales" onClick="changeMenu('formulaire',
    this.value, '.html')">Informations légales</option>
</select>
<select>
    <option value="partenariats" onClick="changeMenu('formulaire', this.value,
    '.html')">Devenez partenaires</option>
    <option value="affiliation" onClick="changeMenu('formulaire', this.value,
    '.html')">Affiliation</option>
    <option value="revue-presse" onClick="changeMenu('formulaire', this.value,
    '.html')">Espace Presse</option>
</select>
</form>
```

- User de codes techniques pour contrecarrer le crawl des moteurs. Il faut toutefois prendre garde à ne pas tomber dans l'excès et risquer de se faire pénaliser. Vous pourrez trouver un exemple avec le code appelé *jQueryRank Sculpting* dans le chapitre 3 traitant du problème de *cloaking* et qui montre comment duper un moteur, favoriser le Bot Herding et le PageRank Sculpting par la technique.

Passer en HTML 5 plutôt qu'en xHTML ou HTML 4 ?

Depuis 2007 se développe la dernière mouture de l'HTML, il est donc évident que des nouveautés ont été apportées et que les moteurs ont su en profiter. Bien qu'aucune communication officielle ne mentionne l'avantage du passage à l'HTML 5, nous pouvons nous baser sur le fait que Google a appuyé le projet et a très rapidement développé ses outils et ses API autour de cette nouvelle technologie.

Google préconise l'usage de l'HTML 5 car la sémantique du code est plus précise et permet l'utilisation des attributs `role` ou encore des extraits de code enrichis. Mais rien ne permet d'affirmer qu'un site construit en HTML 5 serait mieux positionné que son équivalent en HTML 4 ou xHTML. Étant donné que les validations du W3C ne sont pas encore totalement terminées à propos de l'HTML 5 et de son futur HTML 5.1, il est fort probable que les balises n'aient pas encore d'incidence directe sur le positionnement des pages web.

Des tests ont été effectués en mars 2014 sur des contenus similaires pour tenter d'apporter une réponse (source : http://goo.gl/rPcHmn). Les résultats se sont tous montrés à l'avantage de l'HTML 5 mais si nous observons de plus près, rien ne permet de confirmer que les nouvelles balises sont la conséquence de ces classements. En effet, si Google ne fait aucune comparaison des balises, par exemple, cela signifie que les contenus équivalents ont la même valeur. Mais comme le moteur doit distinguer le positionnement des deux pages, il se base sur d'autres critères pour en placer une devant l'autre (poids de la page, date de dépôt sur le serveur...).

Si tous ces tests ont mis en avant les pages codées en HTML 5, et même si rien ne confirme l'avantage de cette version du langage, il faut reconnaître que de nombreux plus sont à noter, notamment dans la gestion des pages, des API ou même de l'Ajax. Nous ne pouvons pas dire qu'il est préférable d'utiliser une balise `<header>` plutôt qu'une simple `<div>` en termes de valeur mais nous pouvons confirmer que l'usage de l'HTML 5 multiplie considérablement les possibilités d'un site web à se développer autour d'une sémantique plus précise et complète.

Il y a fort à parier que dans quelques années, certains aspects du code sémantique de l'HTML 5 attribueront plus de poids au contenu. Il est difficilement pensable qu'un texte placé en pied de page sera aussi valorisé qu'un contenu placé dans l'en-tête. Aussi, nous pouvons légitimement penser que Google et Bing ne seront pas les premiers à nous dévoiler que des balises ont plus de poids que d'autres dans leur algorithme de pertinence, cela aurait beaucoup trop d'incidence sur le spam et sur le monde du référencement.

Seuls des batteries de tests nous permettront d'en savoir davantage à l'avenir sur le poids des balises, mais considérons qu'il est préférable d'opter pour HTML 5 sur bien des facteurs qui peuvent impacter le référencement et le positionnement (sémantique, accessibilité, API, design adaptatif pour les mobiles...).

3

Facteurs bloquants et pénalités Google

Nous avons vu dans les chapitres précédents comment obtenir un socle technique suffisant pour aborder quelques freins et problèmes liés au référencement. Cependant, nous n'avons pas encore étudié en détail les facteurs bloquants pour les robots d'indexation.

Pour commencer, nous allons vous présenter les principales mises à jour algorithmiques et les filtres qui engendrent des pénalités parfois très importantes pour les pages web voire les sites complets. Si nous maîtrisons pleinement ce sujet, nous pourrons anticiper, prévenir et guérir les problèmes relatifs aux freins et aux sanctions du référencement naturel.

Nous allons entrer dans le doux monde des blocages et du Black Hat SEO avec une revue complète des risques encourus avec des exemples de codes et de techniques qui peuvent être mis en place.

Principales mises à jour des moteurs de recherche

Dans un souci d'améliorer les SERP en les rendant de plus en plus qualitatives, Google et Bing mettent régulièrement leur algorithme de pertinence à jour. On estime le nombre d'évolutions à environ 500 par an. Beaucoup de mises à jour sont méconnues et sans grandes conséquences sur l'affichage des résultats de recherche, mais d'autres sont majeures et effrayent le petit monde du SEO.

Bing ne communique pas énormément à propos de ses mises à jour tandis que Google le fait par divers intermédiaires. Les mises à jour du moteur les plus connues (dans l'ordre chronologique) sont : Boston (mars 2003), Esmerelda (juin 2003), Florida (novembre 2003), Brandy (février 2004), Allegra

(février 2005), Jagger (octobre 2005), Big Daddy (février 2006), Caffeine (août 2009 mais officialisée en juin 2010), Mayday (mai 2010), Panda (février 2011), Penguin (avril 2012), PayDay Loan (mai 2013), Hummingbird (septembre 2013) ou encore Google Phantom (Quality update de mai 2015) et RankBrain (novembre 2015).

Toutes n'ont pas connu un développement retentissant mais elles ont permis de changer la donne en matière de recherche sur le Web. Citons l'architecture Google Caffeine qui a permis de restructurer le système d'indexation des pages ou encore Google Mayday qui a servi à mieux positionner les requêtes issues de la longue traîne.

Google Panda

Panda est le nom d'une mise à jour majeure de l'algorithme de Google qui tire son nom des deux ingénieurs ayant travaillé sur le concept, Navneet Panda et Biswanath Panda.

Le filtre est mis en place en février 2011 aux États-Unis avant d'arriver en France officiellement en août 2011, même si certains webmasters affirment avoir vu des changements dès le 15 juin. Plus qu'une simple mise à jour, Google Panda est une opération qui s'applique de façon progressive afin de « nettoyer » la Toile.

Le filtre Panda est appliqué manuellement lorsque Google estime nécessaire de vérifier et nettoyer ses résultats organiques. La firme a lancé plusieurs fois le filtre (au moins vingt-huit fois) jusqu'en juillet 2013 avant de déployer Panda 4.1 le 21 septembre 2014. Depuis le 18 juillet 2015, le trentième lancement a été officialisé, ce qui correspond à Panda 4.2 (source : http://goo.gl/b5UrtE). D'autres mises à jour sont encore à prévoir pour 2016…

Découpage des versions de Google Panda

À ce jour, Google a sorti quatre versions majeures de Panda ainsi que des versions secondaires, ce qui explique les 30 versions différentes existantes.

L'objectif annoncé par Google pour Panda est de sanctionner les sites impropres ayant un contenu de faible qualité et de favoriser ceux qui proposent aux internautes des contenus originaux et uniques. Google garde à l'esprit son but ultime : offrir aux internautes la meilleure expérience possible en tant que moteur pertinent.

D'une manière générale, Google cherche à favoriser les sites dont la valeur ajoutée est plus forte que celle de sa concurrence. Avec Panda, il vise à lutter contre le spam, les sites de mauvaise qualité, les contenus de faible pertinence mais aussi le contenu dupliqué. Le filtre sanctionne principalement les sites sans contenu pertinent ou ayant majoritairement du contenu dupliqué comme :

- les fermes de contenu *(content farms)* ou *scrappers* (sites qui recopient ou volent les contenus de tiers à leur insu) ;
- les sites dont les contenus sont de très faible qualité (mauvaise rédaction, spam, contenus dupliqués…) ;
- les comparateurs de prix ;

- les agrégateurs de contenus ;
- les sites proposant des « codes de réduction ».

Google Panda classe les pages web selon l'importance des points négatifs qui les concernent et applique différentes sanctions :

- pénalité appliquée à l'ensemble du site qui va avoir un impact négatif sur le classement des pages d'un même domaine ;
- pénalité ciblée sur le contenu en cherchant à bannir au maximum les contenus dupliqués dans les SERP ;
- pénalités touchant la publicité abusive, surtout si celle-ci est concentrée au-dessus de la ligne de flottaison (premier écran visible sans défilement) ;
- pénalités pour les sites utilisant Flash, même s'ils proposent une version HTML alternative ;
- pénalités sur les mots-clés insérés dans les noms de domaines car Panda privilégie surtout les domaines originaux ou standards.

Bien que Google Panda ait généré un sentiment de peur et de frustration dans la sphère SEO, il est plutôt facile de ne pas tomber dans ses griffes. Voici comment (liste non exhaustive) :

- rédiger des contenus de qualité, c'est-à-dire des textes qui reprennent certains des points fondamentaux suivants :
  - taille correcte de contenu (il n'existe pas de limite connue mais sachez que 300 mots environ par page est correct) ;
  - structuration des contenus avec les balises `<h1>` à `<h6>` et des balises structurelles (`<div>`, `<nav>`, `<header>`...) ;
  - traitement d'un seul thème par page pour plus de clarté ;
  - apport d'une réelle information au lecteur, contrairement aux textes paraphrasés ou sans aucun intérêt précis ;
- proposer des contenus uniques, c'est-à-dire des textes qui ne résultent pas de pages ou URL dupliquées. Google compare la pertinence et la date de publication pour gérer ses sanctions, mais il arrive fréquemment que le moteur se trompe car le voleur est plus malin que la victime, etc. ;
- faire varier les titres, les ancres de liens et les textes afin que de mêmes phrases ne soient pas toujours répétées, ce qui peut être assimilé à du spam ou de la triche pour Google ;
- limiter l'accès aux pages qui peuvent éventuellement causer des soucis d'indexation ou de compréhension pour les robots. Un bon usage d'un fichier `robots.txt` ou des balises meta `robots` peut s'avérer parfois suffisant pour contrer ce type de phénomène ;
- vérifier le taux de rebond et le temps de visite pour savoir si le comportement des usagers ne va pas nuire au référencement. En effet, Bing et Google analysent de plus en plus en profondeur les données statistiques pour jauger la qualité des pages indexées, cela peut aussi entrer en compte dans le cas d'une sanction, etc. ;

- faire attention aux noms de domaines qui contiennent trop de mots-clés car Google Panda est censé ignorer les termes s'il ressent de l'abus. Désormais, ce phénomène est géré par un autre système anti-EMD *(Exact Match Domain)*.

Malheureusement, il n'existe aucune solution miracle pour savoir si un site a été touché ou non par le filtre Panda. Néanmoins, un bon suivi des statistiques du site (trafic, pages désindexées, pertes de classement...) ainsi qu'une analyse des facteurs mentionnés précédemment sont à surveiller et peuvent nous donner des indices. Enfin, une fréquence de crawl moins importante peut être un indicateur sur la baisse d'intérêt du robot et du moteur pour un site, ce qui signifie généralement que le site n'est pas assez pertinent ou qualitatif...

Les campagnes de nettoyage ponctuelles et répétées de Google Panda ont fait de nombreux dégâts dans le monde du Web. À ce jour, il s'agit certainement de l'une des mises à jour les plus importantes en termes de recherche web.

Pour éviter de tomber dans les griffes du Panda, privilégiez le naturel, l'éthique, la pertinence et la qualité de l'écrit, il s'agit du meilleur moyen d'être serein face à une possible pénalité...

Google Phantom (Quality update)

Le 5 mai 2015 a été marqué par une mise à jour discrète et de nombreuses secousses dans les résultats de recherche, que Google n'avait pas confirmé directement (source : http://goo.gl/SrIfZm). D'abord appelée Google Phantom à cause de l'absence de nom de cette mise à jour, il aura fallu attendre son officialisation par Google le 19 mai 2015 (source : http://goo.gl/vcpVuG) pour que Barry Schwartz du site SearchEngineLand l'intitule « Quality update ».

Si ce nom peut paraître étrange, il est en réalité en adéquation avec l'objectif de la mise à jour. En effet, Google Phantom « Quality update » vise à dévaluer les contenus de faible qualité, un peu à l'image de son précédesseur Google Panda. Il s'agit ici pour sûr d'une modification interne de l'algorithme, qui est appliquée en continu lors du crawl. L'objectif est une nouvelle fois de mettre l'accent sur la qualité des contenus, non pas en pénalisant les pages web sans plus-value, mais en abaissant leur valeur initiale.

Google n'a pas précisé quels types de contenu étaient considérés comme de mauvaise qualité, mais les divers retours post-Phantom donnent des pistes intéressantes. Voici les ressources susceptibles d'être touchées par la dévaluation de la mise à jour :

- pages avec trop peu de contenu (pas assez de texte lisible) ;
- pages avec du contenu créé uniquement pour générer des clics ;
- pages surchargées de publicités, d'interstitiels ou de pop-ups (voir Page Layout plus loin dans ce chapitre, qui répond également à cette problèmatique) ;
- pages surchargées de vidéos ou de ressources cliquables.

Nous pouvons remarquer que la « Quality update » cible surtout les contenus trop maigres ou ceux qui sont intégrés uniquement pour générer du clic. Cela vient compléter la liste des contenus de faible qualité dans le viseur de Google Panda. Si cela a l'avantage de ne faire que dévaluer des contenus (et

non sanctionner les pages), il faut tout de même prendre garde à ne pas mettre en place des contenus trop réduits ou sans valeur ajoutée. Ce serait le meilleur moyen pour ne pas progresser en termes de positionnement…

Google Panda et « Phantom », frères de sang ?

Gary Illyes a indiqué le 15 septembre 2015 que Google Panda devrait être mis à jour moins fréquemment à l'avenir pour être intégré entièrement dans l'algorithme (source : http://goo.gl/DZ0kFo). Dans ce cas, Google Panda et Phantom pourraient se cumuler pour favoriser davantage la qualité des contenus de façon parallèle. Reste à savoir si cela sera officialisé durant l'année 2016 ou ultérieurement...

Google Penguin

Un an après la sortie de Google Panda, la firme n'a pas hésité à relancer une autre mise à jour appelée Penguin. Cette fois, le but est de s'attaquer au netlinking abusif ainsi qu'aux suroptimisations de code trop agressives ou non naturelles aux yeux du moteur.

La première version est sortie le 24 avril 2012. Il ne s'agit que d'un filtre mais qui applique une nouvelle façon de lire les contenus au sein de l'algorithme du moteur de recherche.

Le principal objectif du manchot (et non du pingouin comme la sphère SEO a tendance à l'appeler) est de détecter et pénaliser les liens de mauvaise qualité ainsi que les techniques de spam utilisées pour manipuler l'algorithme du moteur. En d'autres termes, il faut désormais faire du netlinking « propre » pour éviter des sanctions.

À ce jour, 6 principales mises à jour du filtre ont eu lieu, dont seulement une majeure en 2014 puis rien en 2015 :

- Penguin 1 le 24 avril 2012 ;
- Penguin 2 le 26 mai 2012 ;
- Penguin 3 le 5 octobre 2012 ;
- Penguin 4 le 22 mai 2013. Mise à jour appelée en interne 2.0, certainement pour surligner sa relative nouveauté et ses changements majeurs par rapport aux anciennes versions. Matt Cutts a affirmé que cette nouvelle mise à jour du filtre Penguin « analyse plus profondément les pages et les liens » et a « plus d'impact » (source : http://goo.gl/t4m3KH) sur les résultats (au moins 2,3 % des requêtes ont été affectées) ;
- Penguin 5 le 22 octobre 2013. Cette mise à jour est appelée Penguin 2.1 à l'échelle mondiale et 1 % des requêtes ont été affectées ;
- Penguin 6 a été déployée le 18 octobre 2014 et affecte aussi 1 % des requêtes environ (source : https://goo.gl/Y2R7Kl). D'autres mises à jour ont été remarquées jusqu'au 10 décembre 2014. Cette version est appelée Penguin 3.0 dans le monde.

Vers une future version automatisée ?

L'année 2015 a été marquée par l'absence de mise à jour de Google Penguin, ce qui s'explique par un travail acharné de la firme pour rendre automatique le filtre historique. Ainsi, depuis la fin janvier 2016, son application n'est plus activée avec parcimonie mais devient automatique (source : http://goo.gl/cgmBkx). En d'autres termes, Penguin 4.0 (nom officiel) est utilisé en temps réel et plus aucune ressource web ne pourra passer au travers des mailles du filet.

Dans les faits, Google Penguin correspond à une réécriture de l'algorithme afin de chasser l'ensemble des techniques abusives ou frauduleuses en matière de netlinking et de suroptimisation :

- suroptimisation de contenus comme du *keyword stuffing* (littéralement, le bourrage de mots-clés) ;
- automatisation des procédés de netlinking ;
- opérations de netlinking massif. Il s'agit souvent d'inscrire des sites sur des annuaires ou de rédiger de faux communiqués de presse (qui sont aussi des proies de Google Panda puisque les contenus sont souvent de faible qualité voire dupliqués). Le webmaster va utiliser l'ensemble des supports pour récupérer un nombre important de liens sur une période restreinte ;
- techniques de « Spam Co ». Elles consistent à laisser des commentaires sur des blogs, forums, réseaux sociaux, etc., uniquement pour obtenir un lien en retour. Dans ce cas, le commentaire n'a tout simplement aucune valeur ajoutée pour les internautes et il est alors assimilé à du spam ;
- usage abusif des mêmes ancres de liens. Cette technique consiste à obtenir un nombre important de liens en utilisant toujours la même ancre. La technique est sanctionnée par Penguin qui n'aime pas les concordances trop évidentes. Pour une URL du type www.monsite.com/assurance-vie.html, il convient par exemple d'éviter d'utiliser toujours une même ancre comme « assurance vie », cela ne semble pas assez naturel…
- liens sans rapport sémantique avec le sujet de la page. Par exemple, sur une page qui traite de la construction de maisons, un lien vers une page de formation Photoshop avec une ancre comme « Photoshop » ne va pas être tolérée par l'algorithme. Force est de constater que ce lien n'a aucun intérêt pour le lecteur, cela ne résulte pas d'un profil naturel de liens…
- achat multiple de noms de domaines satellites. Il s'agit d'acheter de nombreux noms de domaines génériques uniquement dans le but de faire pointer des liens sur des ancres précises et vers le site qui a mis en place la stratégie de netlinking. Souvent, les domaines satellites sont laissés à l'abandon ou ne proposent que des contenus de piètre qualité. Ce type de procédé est dans l'œil du cyclone avec Panda et Penguin. Google détecte facilement les relations entre les sites satellites et leur cible, il pénalise au moins celui qui recueille les liens entrants mais, généralement, c'est tout le système qui s'effondre...
- achat de liens. Ce procédé est fortement déconseillé par Google bien qu'il ne puisse pas toujours savoir si nous achetons des liens. Acheter des liens signifie que nous sortons des consignes édictées par Google. Jusqu'à nouvel ordre, nous pouvons éviter les sanctions car le moteur ne détecte pas toujours la supercherie, mais il ne laisse planer aucun doute sur une reconnaissance future. Si vous achetez des liens, l'idéal est sûrement d'ajouter l'attribut `rel="nofollow"` si nécessaire.

Pour savoir si vous avez été touchés par Google Penguin, il suffit de suivre le trafic et les statistiques car les chutes sont souvent vertigineuses en nombre de visites. Qui plus est, nous sommes rarement naïfs au point de ne pas nous douter que notre profil de liens n'est pas très « naturel » et qu'il risque donc d'être

touché. Il est également possible de vérifier la messagerie intégrée dans les outils pour webmaster afin de savoir si une sanction est tombée car Google déploie cette fonctionnalité de plus en plus fréquemment.

Et les sanctions sur Bing ?

Bien que Bing n'applique pas encore de sanctions aussi lourdes que Google, Microsoft a mis en place son propre système de messagerie interne pour prévenir les éventuels problèmes liés aux sites enregistrés dans Bing Webmaster Center.

À l'instar de Google Panda, la firme ne nous a pas expliqué en détail ce qu'il faut faire pour éviter le piège de la pénalité. Il faut dire que sa défense est simple : Google a des *guidelines* à respecter. Si nous ne sommes pas dans les cases, les sanctions tombent…

Voici une liste de suppositions et de conseils de bons sens qu'il faut mettre en place pour éviter la pénalité du manchot.

- Optez pour une stratégie de netlinking réellement naturelle.
- Variez les ancres de liens en prenant garde au ratio d'ancres similaires (si par le passé vous avez utilisé trop d'ancres identiques, il est conseillé d'en modifier et de les faire varier avec un profil plus naturel et moins optimisé).
- Diversifiez les types de backlinks (image, texte, JavaScript…) ;
- Privilégiez les sources de qualité, pertinentes et reconnues dans leur domaine pour les liens entrants que vous récupérez.
- Prenez le temps d'analyser le site sur lequel vous envisagez de placer le lien (thématique, qualité, PageRank, fréquence de publication, présence sur les réseaux sociaux…).
- Évitez tout système d'automatisation autour de l'obtention de liens entrants (certes, cela demande plus de temps mais il vaut mieux cela que d'être sanctionné lourdement).
- Envisagez une stratégie de netlinking sur le long terme.
- Maîtrisez le ratio follow/nofollow (il est admis qu'un ratio de 25 %/75 % est raisonnable).
- Surveillez fréquemment les backlinks à l'aide d'outils dédiés comme MajesticSEO, Ahrefs, Open-SiteExplorer, Ranks.fr...

Contrairement à Google Panda qui applique une sanction à la totalité du site, Penguin n'applique la pénalité qu'aux URL suspectées de triche. Nous pouvons donc avoir des pages déclassées dans les SERP sans que d'autres pages du même site soient affectées.

En définitive, Google Penguin ne pénalise que les pages contenant des backlinks artificiels, de mauvaise qualité ou ayant eu recours à des suroptimisations du code. Ceci est dû au fait qu'il s'agit d'une réécriture ou d'un complément dans l'algorithme initial. Par conséquent, chaque page dispose de son propre traitement et peut aussi bien être mise en avant que pénalisée…

D'une manière générale, essayez de garder à l'esprit ces conseils :

- le netlinking se développe sur le long terme et ne doit pas être automatisé ;

- le nombre de liens entrants doit évoluer de manière régulière et naturelle pour développer un profil sain et naturel de backlinks auprès des moteurs de recherche.

L'objectif de Google n'est pas nécessairement de sanctionner à tout va mais il tient à lutter contre les fraudes dans le but d'améliorer ses résultats de recherche en s'approchant au plus près des attentes des internautes. Tout est une affaire de bon sens et de logique. Si vous faites le travail correctement, de façon naturelle et dans une optique dédiée à l'utilisateur, vous ne courrez quasiment aucun risque et tout ira bien pour votre site.

Que ce soit pour Google Panda et Penguin ou pour les lecteurs, portez toujours attention aux contenus, à leur originalité, à leur fréquence de mise à jour, à leur valeur ajoutée et surtout à la stratégie de netlinking associée. Plus vous saurez apporter de la plus-value à vos visiteurs tout en faisant des optimisations les plus propres possibles, plus vous serez respectés par l'algorithme et les filtres de Google, le jeu en vaut la chandelle…

Les EMD (Exact Match Domain)

Toujours dans l'optique de fournir aux utilisateurs du moteur de recherche des résultats de plus en plus qualitatifs, Google a décidé de mettre en place un algorithme visant à éliminer, ou au moins à affecter, les noms de domaines de mauvaise qualité comprenant trop de mots-clés. Ces noms de domaines à mots-clés (généralement séparés par un tiret) s'appellent EMD pour *Exact Match Domain*. L'algorithme anti-EMD a été annoncé par Matt Cutts via son compte Twitter dès le 29 septembre 2012.

Nous savons tous ce qu'est un domaine, il s'agit d'une adresse textuelle qui masque une adresse IP de serveur afin de faciliter la mémorisation et d'améliorer la communication autour d'un site.

Les noms de domaines sont souvent composés de plusieurs mots accolés ou séparés par des tirets (ce qui constitue la meilleure solution pour que les robots les lisent correctement). Les domaines peuvent être précédés de plusieurs préfixes tels que www ou tout autre sous-domaine, chacun constituant un site à part entière d'un point de vue technique (seul l'URL est assimilée au domaine).

En SEO, il n'existe aucune documentation qui indique qu'un nom de domaine est plus favorable qu'un autre, il faut juste déduire que les mots-clés qui le composent ont un impact sur le positionnement à la fois dans les contenus mais aussi dans les liens et les ancres associés. Le choix de l'extension peut aussi poser question, il est fréquemment conseillé d'opter pour les TLD *(Top-Level Domain)*, c'est-à-dire pour les domaines dont les extensions sont reconnues, telles que .fr, .com, .org, .gouv, .net, .info…

Nous pouvons nous demander pourquoi Google s'est décidé à chasser les noms de domaines abusifs. Certes, il arrive parfois que les URL frôlent le ridicule tant cela se voit que le nom de domaine a été choisi pour des raisons évidentes de référencement mais dans la majorité des cas, les sites pertinents et de qualité bénéficient de noms de domaines plutôt raisonnables.

Les EMD sont des domaines qui comportent plusieurs mots-clés comme s'il s'agissait d'une requête classique. Par exemple, un nom de domaine comme www.vente-automobile-paris.fr est un EMD typique.

Figure 3-1

Exemple de EMD

L'utilisation des EMD a été longtemps privilégiée pour une raison toute simple : ils permettent de gagner facilement quelques places dans les SERP et sont plutôt incitatifs pour les visiteurs. En effet, les mots-clés placés dans les EMD renforcent la pertinence du site sur ces termes précis, il s'agit donc d'une des vieilles techniques SEO qui permettait d'améliorer le classement dans les SERP. De plus, les EMD bénéficient d'un avantage non négligeable puisqu'ils peuvent obtenir des ancres optimisées sans effort.

Google a toujours accepté l'utilisation de l'EMD à des fins de positionnement tant que cela ne devenait pas abusif mais depuis quelques mois les conditions ont changé. Avec son algorithme anti-EMD, Google s'est mis en chasse contre tous les spammeurs spécialistes qui règnent sur la Toile.

Le but est clairement de nettoyer les SERP et de supprimer les sites jugés nuisibles par Google. Il est vrai que ces noms de domaines n'ont aucune plus-value et ont même tendance à subir l'effet pervers de leur intérêt initial, à savoir de faire fuir les visiteurs qui semblent parfois étonnés de voir des noms de domaines si longs, étonnants et impossibles à mémoriser…

Google analyse une batterie de critères avant de prendre sa décision finale : sanction ou non. Il lit le contenu, calcule la fréquence de mise à jour des contenus, le nombre de pages du site et leur valeur, le nombre de visites, étudie la fidélité et la satisfaction des visiteurs, le taux de rebond, le temps moyen passé sur le site, les signaux sociaux… Si plusieurs facteurs manquent de pertinence aux yeux des robots alors le moteur peut entraîner des sanctions liées à l'EMD, il ne se limite pas uniquement à une analyse simpliste des mots-clés qui le compose.

De nombreux sites ont été touchés par cette nouvelle mise à jour et ont vu leur classement chuter, mais attention, il s'agit de sites avec un faible contenu, peu attractifs, et principalement des sites de jeux, pornographiques, de téléchargement ou encore des MFA *(Made for Adsense)*… En réalité, l'impact semble encore limité en France mais ce phénomène devrait s'accroître dans les années à venir si les EMD continuent de pulluler sur la Toile.

Conseils pour l'enregistrement de votre nom de domaine

Il est conseillé d'enregistrer votre nom de domaine (NDD) avant la mise en ligne d'un site et pour au moins deux ans si vous croyez en votre affaire. L'ancienneté du NDD est prise en compte par les moteurs de recherche et cela se ressent dans les SERP.

En revanche, oubliez le mythe de la durée de réservation d'un nom de domaine. En effet, le fait de renouveler un domaine tous les 3 ans (voire davantage) n'a pas d'incidence directe sur le positionnement d'un site web, contrairement à l'idée couramment répandue dans la sphère SEO. Les robots n'ont pas toujours accès à ces informations et ne peuvent donc pas en tirer profit.

Si vous en avez la possibilité, n'hésitez pas à enregistrer plusieurs variantes d'un nom de domaine : soit avec plusieurs extensions (.fr, .com), soit en jonglant avec les écritures au pluriel et/ou au singulier, etc. La raison est simple : cela permet d'éviter à la concurrence de se confronter à vous mais aussi d'être quasi assuré que les internautes tombent sur vous s'ils se trompent d'extension. Sachez toutefois que si vous possédez plusieurs noms de domaines, il faudra pratiquer des redirections 301 pour ne courir aucun risque de sanctions par les moteurs.

Si vous voulez acheter un nom de domaine mis en vente, cela peut être intéressant pour le référencement mais vérifiez avec soin son historique afin d'éviter les mauvaises surprises. Imaginez que celui-ci a été pénalisé par Google ou Bing dans le passé, ou encore qu'il a été blacklisté, cela pourrait s'avérer catastrophique pour le repreneur... Attention donc, n'investissez que dans des noms de domaines fiables et « propres », si possible avec un bon PageRank et de bons backlinks déjà en place !

Avec ses diverses mises à jour (Panda, Penguin, anti-EMD, etc.), Google vise toujours à améliorer ses résultats et l'expérience utilisateur en passant bien évidemment par la mise en avant des sites jugés de qualité. Google nous a habitués à ne pas tricher depuis des années, tout est une question de bon sens. D'où l'importance de rester « naturel » et de faire les choses pour l'utilisateur et non pas pour les moteurs de recherche.

Bing développe son propre algorithme anti-EMD

Bing a annoncé sur l'un de ses blogs officiels le 9 septembre 2014 l'ajout d'un filtre visant à contrer l'abus de mots-clés *(keyword stuffing)* dans les URL (http://goo.gl/hksW1T). Microsoft a indiqué que près de 130 millions d'URL ont déjà été touchées pour 5 millions de sites sanctionnés, sachant que cela devrait continuer dans les mois à venir...

Google Page Layout

En janvier 2012, Matt Cutts annonçait un nouveau filtre visant à pénaliser les pages proposant trop de publicités et obligeant les visiteurs à utiliser la barre de défilement pour atteindre le contenu réel de la page. Ce filtre intitulé Page Layout vise à livrer bataille contre les publicités trop nombreuses situées au-dessus de la ligne de flottaison (première écran visible).

Néanmoins, si pour des raisons légitimes, nous utilisons de la publicité sur notre site, cela reste possible et n'est pas interdit. En réalité, c'est la façon de faire et le positionnement des blocs de publicités qui changent. Il est désormais fortement conseillé de les intégrer dans des zones stratégiques et de les utiliser avec modération...

Ne placez que deux ou trois publicités par page en les répartissant de façon homogène au travers des contenus. Par exemple, nous pourrions avoir une publicité en haut de page, proche du logo, une deuxième située dans la colonne de droite et une dernière positionnée en bas des contenus. Ainsi, nous aurions une répartition saine et cela éviterait d'être pénalisé par Page Layout.

Figure 3-2

Exemple de l'outil BrowserSize du site www.sitepenalise.fr

L'outil Browser Size de Google Labs

Google Labs propose l'outil Browser Size pour tester l'affichage des contenus en ligne et montrer les zones les plus visibles par les visiteurs. Par habitude, nous avons tendance à positionner les publicités dans des zones chaudes et cet outil nous permet de remarquer aisément si nous abusons des espaces publicitaires au risque de prendre une sanction à cause du filtre Google Page Layout. Malheureusement, cet outil ne fonctionne plus depuis quelques mois…

Il existe deux alternatives intéressantes au feu Browser Size pour tester et mesurer l'impact des publicités sur les sites web :

- le site http://www.sitepenalise.fr/browsersize/ ressemble quasiment en tout point à l'outil de Google et affiche des zones colorées pour montrer quels espaces sont à exploiter ou non ;
- Google Analytics permet de surveiller les zones chaudes des pages web et donc de savoir comment répartir les publicités.

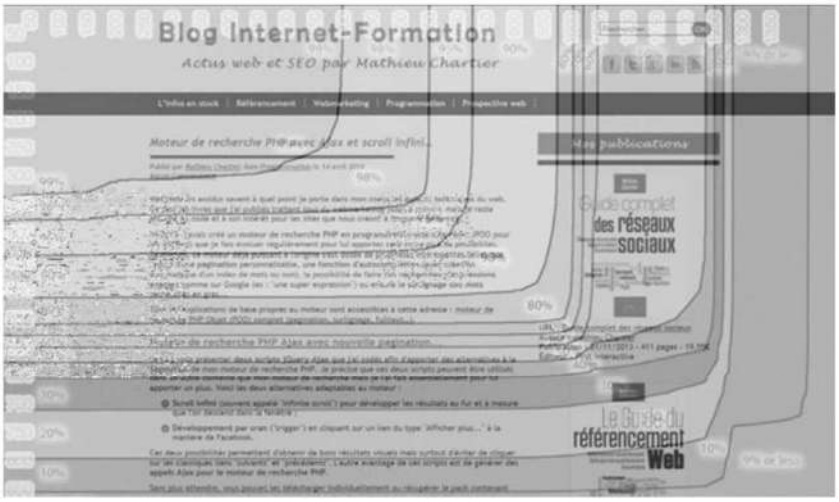
Google Analytics donne également accès à un outil d'analyse des pages web qui permet aisément de repérer les zones chaudes ou moins fiables pour inciter les internautes au clic. L'avantage de cet outil est qu'il se base également sur les données statistiques et non pas uniquement sur des concepts théoriques. Les informations sont donc assez intéressantes et qualitatives pour savoir si nos publicités sont trop nombreuses ou trop bien placées au point d'être pénalisé.

Pour utiliser cet outil, connectez-vous à votre compte sur Google Analytics, puis cliquez sur *Comportement>Analyse des pages web*. Chaque zone se voit indiquer un pourcentage de clics et donc un potentiel de visibilité. Il est conseillé d'ajouter les couleurs et l'option *Taille du navigateur* pour que l'ensemble soit plus détaillé. Ainsi, nous pouvons sans risque détecter les zones fortes ou faibles de nos

pages et donc savoir comment optimiser le positionnement des publicités tout en évitant l'éventualité d'une sanction de Google.

Figure 3-3

Analyse des pages web dans Google Analytics



Depuis le 6 février 2014, le filtre Page Layout en est à sa version 3 (les autres versions datent de janvier et octobre 2012) mais rien ne dit qu'il est appliqué.

En effet, Matt Cutts met fréquemment en garde les internautes contre les abus publicitaires mais a également répété à plusieurs reprises que le filtre n'était pas nécessairement en place. Est-ce un piège ? Est-ce une réalité ? Difficile de répondre mais si le filtre a évolué jusqu'à la version 3, nous pouvons imaginer que Google ne travaille pas dans le vide. Ce filtre peut donc à tout moment être activé ou pénaliser des sites web peu consciencieux en termes de publicités. Il faut donc rester très vigilant à ce sujet.

Attention aux interstitiels d'installation d'apps mobiles

Depuis le 2 novembre 2015, Google a officiellement mis à jour son algorithme *mobile-friendly* pour pénaliser les pages web d'installation d'applications mobiles contenant des interstitiels trop envahissants (source : http://goo.gl/Uh8LYo). En d'autres termes, les applications qui proposent des interstitiels de téléchargement d'apps mobiles prenant trop de place dans l'écran des utilisateurs peuvent être pénalisées par Google et voir les liens profonds chuter dans les SERP.

Google PayDay Loan

Les mises à jour de PayDay Loan font bien moins de bruit que celles de Google Panda ou Penguin. Elles sont néanmoins importantes et montrent la nouvelle politique mise en place par la firme.

Il s'agit d'un algorithme lancé en catimini, annoncé en prévision le 13 mai 2013 et officialisé le 11 juin 2013 dans une vidéo décryptée par Search Engine Land (source : http://goo.gl/snBke9). L'objectif des PayDay Loans est de nettoyer le moteur des requêtes qui entraînent un nombre trop important de résultats polluants.

Nous savons que des thématiques et des mots sont plus porteurs ou ont plus d'autorité pour les moteurs. Nombre de fraudeurs en profitent pour intercaler des résultats sans valeur ajoutée mais bien positionnés grâce à des optimisations souvent abusives. PayDay Loan vise donc à sanctionner ces sites et à nettoyer les requêtes les plus touchées par ce fléau, notamment dans des thématiques telles que la pornographie, le rachat de crédit ou encore des requêtes concurrentielles ou rentables. Globalement, ce sont des requêtes qui touchent de près à la notion d'économie ou à la pornographie qui sont visées par Google, mais nous pouvons imaginer que ce type de procédé sera développé à plus grande échelle à l'avenir.

Il existe actuellement trois versions de cet algorithme, la première datant de mai-juin 2013, une deuxième passée un peu inaperçue quelques jours avant le déploiement de Google Panda 4, le 20 mai 2014 (source : http://goo.gl/P6aWXy) et enfin une troisième déployée dans la foulée le 12 juin 2014. PayDay Loan touche à la fois les sites *spammy* (version 2) – qui usent de techniques de triche pour duper les moteurs – et les requêtes *spammy* (version 3). La lutte contre les mauvais contenus est donc double et ne se limite pas uniquement aux requêtes de mauvaise qualité.

Les requêtes touchées varient d'un pays à un autre. Matt Cutts a annoncé que la première version a touché 0,3 % des requêtes américaines mais aussi 4 % des recherches turques, ce qui montre des disparités importantes. Pour la deuxième version, 0,2 % des requêtes anglaises ont été affectées mais d'autres pays ont aussi subi les foudres de l'algorithme.

Il faut avouer qu'il existe encore de nombreux résultats *spammy* qui sont bien positionnés mais qui n'apportent aucune plus-value voire ne répondent pas à la recherche des internautes. Il suffit de faire quelques tests dans le milieu économique, de l'érotisme ou de la pornographie pour s'en rendre compte, comme on le voit sur la figure 3-4 montrant des pages générées par le site Mediamass en toute circonstance sur des requêtes érotiques.

Retenons également que Google a annoncé qu'il étudierait de plus en plus les sites piratés, ce qui pourrait aller en complément de ce type d'algorithme pour nettoyer complètement les SERP d'un maximum de pages polluantes.

Figure 3-4

Pages auto-générées et polluantes sur Mediamass

Redirections mobiles spammy pour faire de l'affiliation

Google est entré activement en lutte contre les redirections *spammy* mises en place, volontairement ou non, sur des sites web dans le but de générer de l'affiliation et de faire de la monétisation déguisée. La firme a officialisé ces nouvelles sanctions manuelles le 29 octobre 2015 (source : http://goo.gl/mbJs98) et prévient les fraudeurs qui tenteraient d'user encore de ces méthodes.

C'est essentiellement les contenus mobiles qui sont visés par ces méthodes de *cloaking* peu scrupuleuses. En effet, ces redirections sont réalisées parfois à l'insu des administrateurs du site, à cause de malwares ou de codes malveillants. Quand un visiteur clique dans un résultat des SERP, il est redirigé automatiquement vers une page non désirée dont l'unique but est de gagner de l'argent par de l'affiliation discrète.

Figure 3-5

Redirections douteuses et spammy sur mobile

Google a précisé que les sources de ce *cloaking* mobile pouvaient être volontaires (action délibérée du webmaster) ou non (script malveillant activé à l'insu du propriétaire d'un site). Quelle que soit la source du problème, les sanctions manuelles sont appliquées tant que les redirections indésirables sont encore en place, il convient donc de faire attention à ne pas se faire pirater…

Sites piratés

Google a annoncé le 5 octobre 2015 durcir les sanctions contre les sites piratés, que cela soit de la faute des administrateurs ou que le *hacking* provienne de l'extérieur (source : http://goo.gl/4e0BYj).

Figure 3-6

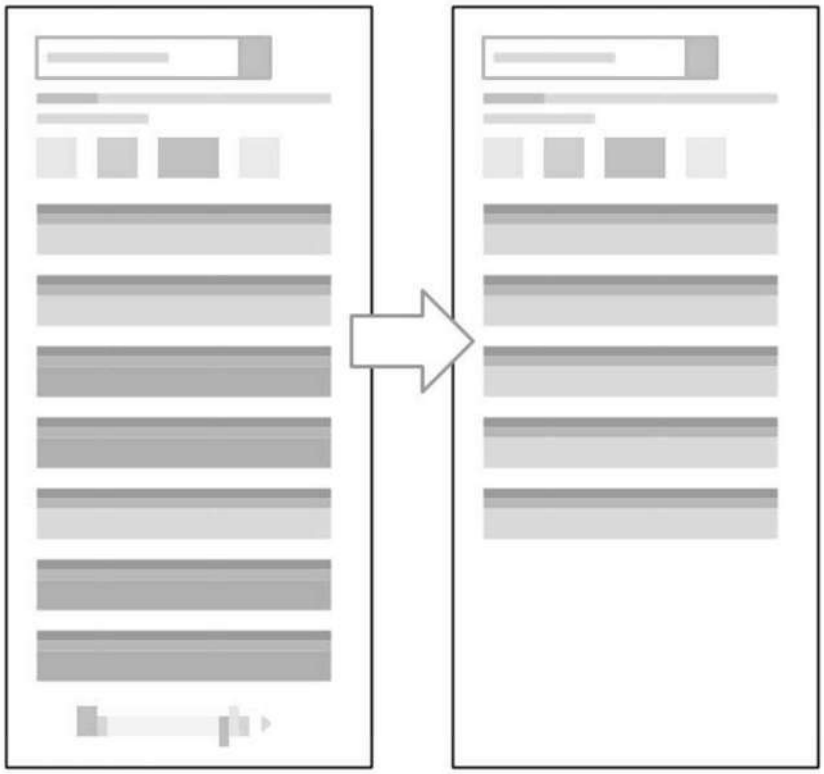
Message de la Search Console contre les pages piratées

Dans les faits, 5 % des requêtes *spammy* détectées par Google sont concernées par la mise à jour. Cela signifie que le moteur connaît les requêtes qui sont ciblées par les piratages réalisés dans les pages web. Si une page est piratée et ciblée par une requête, il la pénalise et la fait chuter dans les SERP. L'objectif est en quelque sorte d'éduquer les webmasters et de les forcer à sécuriser leur site web pour éviter tout problème de crawl et tout risque pour les visiteurs.

Sur le plan visuel, Google précise que sur certaines requêtes (selon les langues cibles), il peut arriver que seuls les résultats fiables soient mis en avant sur la première page, réduisant alors la SERP habituelle de 10 résultats à moins.

Figure 3-7

SERP réduite par le filtre visuel de suppression des pages piratées



Si vous avez peur de vous faire pirater ou de ne pas savoir si votre site est infecté, sachez que Google a mis en place un outil dédié depuis quelques mois et remis à jour le 20 octobre 2015 (source : https://goo.gl/5y6zDd), soit deux semaines après le lancement de ces nouvelles sanctions antispam. Vous pouvez le trouver à cette adresse : https://goo.gl/anLWqN.

L'outil indique le niveau de risque du site et les éventuels problèmes rencontrés. Il peut aussi bien être utile pour les piratages classiques que pour les problèmes d'affiliation *spammy* évoqués dans la sous-partie précédente.

Qualité des contenus chez Bing

Bing est supposé concurrencer Google sur la recherche sémantique comprenant les requêtes des internautes puisqu'il a racheté en juillet 2008 le moteur de recherche sémantique Powerset pour la modique somme de 100 millions de dollars. Rien ne dit que la firme a mis en place des systèmes de reconnaissance des requêtes mais cela semblerait plausible plusieurs années après ce rachat.

Toutefois, nous pouvons penser que la recherche sémantique existe puisque Duane Forrester de Bing a annoncé le 20 février 2014 que la qualité des contenus mais surtout de l'écriture pouvait affecter le classement dans les résultats (source : http://goo.gl/94kGoW).

En d'autres termes, Bing sanctionne les fautes d'orthographe et les contenus dont la qualité rédactionnelle laisse à désirer, ce qui signifie qu'il aurait une approche sémantique importante. Cette nette avancée risque de donner des boutons aux personnes qui font beaucoup de fautes de frappe et d'orthographe mais à ce jour, nous n'avons aucune preuve de son application ni de son impact dans les SERP de Bing.

Facteurs bloquants et solutions alternatives

Frames

Les jeux de cadres, ou *frames*, ont vu le jour dès les origines du Web au sein du langage HTML. Cette technique n'est certainement plus utilisée à ce jour mais elle a longtemps été en première ligne avant que les tableaux ou des `<div>` ne prennent le dessus. Les frames permettent de sectionner les pages web en plusieurs fichiers HTML distincts, ce qui offre l'avantage de faciliter la gestion des contenus en les découpant selon un ordre logique.

Les frames utilisent les balises `<frameset>` à la place de `<body>` mais aussi `<frame>` et `<noframes>` pour préciser les fichiers HTML qui correspondent aux différentes parties du site. L'exemple suivant montre un code correspondant à une page d'accueil découpée en trois fichiers distincts : le premier pour l'en-tête du site, le deuxième pour le menu latéral et le dernier pour la section destinée aux contenus.

```
<frameset rows="150px,*">
<noframes>Navigateur qui ne supporte pas les frames !</noframes>
    <frame src="entete.html" />
        <frameset cols="20%,80%">
            <frame src="menu.html" />
            <frame src="contenu.html" />
        </frameset>
</frameset>
```

Visuellement, les frames n'ont rien à envier aux autres techniques de création de pages web, mais elles posent un sérieux problème en matière de référencement et de positionnement.

Le fait d'avoir des pages découpées en plusieurs fichiers limite l'indexation car les robots peuvent rapidement se perdre et laisser des pages sur la touche. En effet, soit les robots peuvent ignorer totalement les

pages, soit ils peuvent tenter d'indexer les fichiers HTML mais dans la plupart des cas, il sera impossible de tout retenir. Si nous reprenons notre exemple, il y a de grands risques pour que l'en-tête soit ignoré car aucun lien ne mène vers cette section, sauf dans la page d'accueil du site. Qui plus est, il était frustrant de tomber sur une section de site dans les résultats de recherche, notamment lorsqu'il ne s'agissait pas du menu, car nous nous retrouvions souvent bloqués sans d'autres choix que de cliquer sur le bouton Précédent du navigateur.

L'autre inconvénient des cadres est qu'ils génèrent une page d'accueil peu valorisée et mal positionnée. En effet, nous avons l'habitude des pages d'accueil fortes avec un référencement abouti, mais dans ce cas précis, les contenus n'appartiennent pas directement à la page d'accueil, ils sont juste reliés à cette dernière grâce aux balises `<frame>`. Les moteurs auront donc du mal à valoriser les contenus.

Sachez toutefois que la balise `<noframe>` est lue par les robots, elle permet généralement d'ajouter un texte alternatif en cas d'incompatibilité avec les frames. Heureusement, l'usage de cette balise permet d'insérer des contenus qui peuvent sauver un peu la mise, mais quoi qu'il en soit, il reste fortement déconseillé d'utiliser cette technique. Insérez des liens hypertextes à l'intérieur du contenu de la balise `<noframes>` pour faciliter l'indexation et le suivi par les robots.

Si vous souhaitez toutefois obtenir un résultat similaire en matière de gestion des contenus, vous pouvez découper vos pages en plusieurs fichiers comme pour les frames mais en utilisant des inclusions. PHP fait ceci très bien, la page d'accueil de notre exemple serait alors créée de toutes pièces par inclusion de contenus, et non par séparation de contenus. Ainsi, vous pouvez profiter d'une meilleure gestion des pages sans subir les inconvénients des cadres, comme le montre le code suivant, sans le CSS associé :

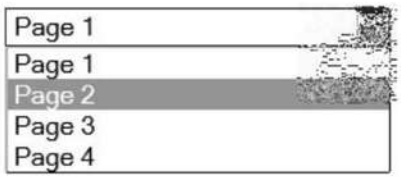
```
<body>
   <?php include_once('entete.html'); ?>
   <div class="menu">
      <?php include_once('menu.html'); ?>
   </div>
   <div class="contenu">
      <?php include_once('contenu.html'); ?>
   </div>
</body>
```

Listes déroulantes avec liens HTML

Les robots n'ont pas toujours la possibilité de suivre des liens, notamment lorsque le contexte technique agit comme un frein. Les listes déroulantes utilisées pour créer des menus discrets font partie des techniques courantes qui représentent un frein à l'indexation.

Figure 3-8

Exemple de menu déroulant en HTML



Le problème ici vient du fait que les listes sont créées à partir des balises `<select>` et `<option>` associées à du JavaScript, et non à des hyperliens que les robots peuvent suivre. La technique consiste à concevoir une liste déroulante avec un événement JavaScript déclenché au clic de la souris ou au changement de valeur qui permet de faire une redirection vers la page de destination.

Les robots n'ont pas accès à ces événements et ne voient pas de liens, ils omettent donc les pages cibles et réduisent les chances d'indexation. La conséquence est double puisque ce sont par la même occasion des liens en moins pour favoriser la transmission du PageRank de Google et du BrowseRank de Bing, ce qui peut également avoir un petit impact sur le positionnement des pages.

Voici deux exemples de codes HTML avec JavaScript associés pour créer un même menu à partir d'une liste déroulante :

```
<!-- Première technique avec script et formulaire HTML -->
<script language="JavaScript">
function changeMenu(nameForm, url) {
document.forms[nameForm].action = url;
document.forms[nameForm].submit();
}
</script>
<form name='formulaire'>
<select>
<option value="page1.html" onclick="changeMenu('formulaire', this.value)">Page 1</option>
<option value="page2.html" onclick="changeMenu('formulaire', this.value)">Page 2</option>
<option value="page3.html" onclick="changeMenu('formulaire', this.value)">Page 3</option>
</select>
</form>

<!-- Seconde technique avec JavaScript et onchange -->
<select onchange="window.location.href=this.value">
<option value="page1.html">Page 1</option>
<option value="page2.html">Page 2</option>
<option value="page3.html">Page 3</option>
</select>
```

Si votre site contient ce type de navigation, prenez garde et veillez à établir des liens vers les pages cibles par le biais d'un plan de site voire directement dans certains contenus. Rien ne remplacera un bon menu mais les robots seront ainsi dirigés vers les pages par un autre accès et le jus de liens pourra être transmis.

Vous pouvez aussi utiliser le fichier `sitemap.xml` pour indiquer l'existence des pages de destination, mais si vous n'utilisez que cette technique, vous limiterez les chances d'enregistrement des pages et vous ne transmettrez aucun PageRank.

Nous verrons un peu plus loin que le JavaScript peut être bloquant à bien des égards, les listes déroulantes n'en sont qu'un exemple. Ce type de procédé peut s'appliquer dans bien d'autres cas et donc freiner le référencement global des pages web.

Intérêt des listes déroulantes

L'usage des listes déroulantes peut en revanche s'avérer intéressant en cas de Bot Herding ou de PageRank Sculpting puisque cela permet d'orienter les robots d'indexation vers les pages qui nous intéressent sans transmettre de jus de liens aux pages ciblées par ces menus déroulants.

Formulaires et accès limités

Les formulaires tiennent un rôle majeur dans les pages web car ils constituent souvent la clé de voûte entre le site et les internautes. Nous avons vu avec les listes déroulantes que les formulaires pouvaient même être utilisés pour créer des menus déroulants, mais pas seulement. En effet, la quasi-totalité des actions effectuées par les visiteurs se fait autour de formulaires à remplir : recherche, prise de contact, accès à un compte, etc.

Nous trouvons de plus en plus fréquemment des navigations orientées par des formulaires de recherche à choix multiples. Prenons un exemple concret : une recherche de côte automobile se fera en sélectionnant la marque du véhicule, puis son modèle, puis son année (...) avant de cliquer sur un bouton de validation qui affichera la page de destination correspondante.

Figure 3-9

Exemple de formulaire de recherche avec redirection vers des pages web dissimulées

Techniquement, les formulaires sont développés autour des balises `<form>`, `<input />`, `<select>` et `<textarea>`. La particularité des formulaires est d'imposer un traitement en amont, souvent à l'aide d'un langage orienté serveur comme PHP, Java ou ASP, bien qu'il soit possible de réaliser des traitements en JavaScript pour les plus acharnés d'entre nous (déconseillé pour des raisons de sécurité, le code du traitement étant visible dans les codes sources notamment).

Un formulaire redirige les internautes vers une page tierce (parfois la même page avec un traitement effectué au rechargement) grâce à l'attribut `action`. Les robots ne peuvent pas suivre les liens générés par les boutons de formulaire ou la cible visée par l'attribut `action`, c'est là tout le nœud du problème.

Ce phénomène est courant mais totalement bloquant pour les robots d'indexation. Dans ce cas, il faut recourir à un plan de site et un fichier `sitemap.xml` pour contourner le problème. Il peut aussi être conseillé d'insérer des liens avec une ancre comme « calculer la cote du véhicule » (pour notre exemple)

quand une page en mentionne un en particulier, ou encore dans les contenus qui ont un lien avec les pages cachées. Ces connexions ne feront qu'améliorer le transfert du jus de liens mais aussi le passage des robots, et donc les chances d'indexation et de meilleur positionnement.

Il est important de se méfier des formulaires menant vers un accès restreint. Dans ce cas précis, nous souhaitons bloquer l'accès aux usagers mais aussi aux robots puisque les pages camouflées ne doivent pas être visibles dans les SERP. Croyez-le ou non, mais il arrive encore fréquemment que ce type d'accès limité ne soit qu'un leurre... Trop souvent, nous arrivons sur une page qui nous demande de nous authentifier et celle-ci nous mène vers les pages cachées, mais en réalité, beaucoup de webmasters omettent à tort de placer un code de vérification dans les pages secrètes. Il suffit donc de connaître l'URL pour les lire, mais pire encore, les robots, dont Googlebot, peuvent aussi indexer ces contenus.

L'expérience prouve que ce cas n'est pas aussi rare que nous pouvons le penser et un déréférencement de qualité ne ferait pas de mal pour éviter ce type de mésaventure. La première solution est d'ajouter un code de vérification de session dans les pages incriminées, comme dans l'exemple suivant :

```
<?php
function isSessionActive($log = "pseudo", $redirect = "index.php") {
    if(!isset($_SESSION)) {
        session_start();
    }
    if(!isset($_SESSION[$log]) || empty($_SESSION[$log])) {
        session_unset(); // Vide la session en cours
        session_destroy(); // Supprime la session
        header('Location:'.$redirect); // Redirection automatique
        exit();
    }
}
?>
```

Une deuxième méthode consiste à utiliser le fichier `robots.txt` pour bloquer l'accès à l'ensemble des fichiers et répertoires censés être masqués pour les utilisateurs. Dans ce cas, il suffit d'ajouter les directives correspondantes pour limiter l'accès. Les deux techniques combinées évitent de mauvaises surprises et auront un réel impact bloquant pour les robots.

Enfin, il existe une méthode peu répandue mais qui peut pourtant aider dans certains cas à mener le robot vers une page optimisée de notre choix sans pour autant bloquer le traitement d'un formulaire. En effet, rien ne nous oblige à utiliser les boutons classiques en HTML, il est aussi possible de créer des boutons avec les balises de liens `<a>` et de lancer le traitement grâce à la `fonction submit()` en JavaScript. Toutefois, cela ne suffit pas pour plaire aux robots car les boutons de soumission ressemblent en général à ceci :

```
<a href="#" onclick="fonctionSoumission()">Soumettre</a>
```

Dans ce cas, cela n'est pas intéressant du tout car l'attribut `href` du lien est vide et bloque le robot. Nous pouvons même imaginer que cela abaisse la part du jus de liens transmis à chaque lien de la page

(puisqu'il est divisé en fonction du nombre de liens présents). En fait, toute la technique se situe dans l'ajout d'une URL au sein de l'attribut `href` mais aussi de l'instruction `return false;` après la fonction de soumission en JavaScript. Ainsi, le moteur peut suivre le lien et l'instruction `return false;` bloque le lien actif au profit du traitement de formulaire pour l'utilisateur.

Nous allons étudier un code complet mais très simple pour vous montrer comment faire pour qu'un robot suive un formulaire. Je vous laisse imaginer tous les usages possibles que cela peut entraîner car nous pourrions indexer nombre de pages souvent délaissées voire rompre l'aspect bloquant de certains formulaires simples.

Voici tout d'abord le cas d'un formulaire sur un champ (comme un moteur de recherche interne) :

```
<form method="post" id="formulaire">
<input type="text" value="" name="champ" />
<a href="pageoptimisee.html" onclick="soumission('formulaire'); return false;"
name="bouton">Soumettre</a>
</form>
```

Voici maintenant la fonction JavaScript de soumission :

```
<script type="application/JavaScript">
function soumission(idFormulaire) {
    // Récupération des données du formulaire
    var formulaire = document.getElementById(idFormulaire);

    // Soumission du formulaire
    formulaire.submit();
}
</script>
```

Cet exemple est très simple pour que vous puissiez bien tester et comprendre comment détourner le problème des boutons de formulaire. Il existe des alternatives bien plus intéressantes en matière de référencement, que seule notre imagination peut bloquer. Mais retenons que cette méthode n'est efficace que dans le cas de formulaires qui sont utilisés en lieu et place de menus ou en cas de fenêtre de transition. Elle permet alors de ne pas voir le robot confronter à un mur mais de le rediriger vers l'accueil, par exemple, ou tout simplement vers une page optimisée créée pour l'occasion.

ActionScript et sites Full Flash

Le format Flash et son langage associé, l'ActionScript, ont toujours été problématiques pour les robots lorsqu'il s'agit de navigation et de lecture de pages. En effet, le Flash n'est pas dérangeant s'il est utilisé avec parcimonie ou pour des besoins ponctuels comme des publicités. Il devient en partie bloquant lorsque nous concevons des sites *Full Flash* ou des menus en Flash. En effet, les moteurs de recherche n'arrivent pas à suivre profondément les liens et à lire les contenus des documents .swf.

Plusieurs problèmes se posent avec ce format :

- le Flash est obsolète sur les smartphones et n'est pas lu nativement par les navigateurs, il faut ajouter un plug-in pour avoir accès aux contenus ;
- les sites Full Flash ne sont constitués que d'une seule page HTML qui renvoie vers le fichier .swf contenant le site. Par conséquent, seule une page peut être indexée et positionnée ;
- les liens internes aux fichiers SWF sont mal interprétés voire non lus par les principaux moteurs. Seul Google arrive à lire quelque peu ce format, bien que les travaux aient été abandonnés depuis l'émergence d'HTML 5.

Les sites en Flash sont de plus en plus rares et souvent relégués au fin fond des SERP, ce qui force les plus motivés à réfléchir avant de se lancer dans ce genre d'aventure. Si toutefois vous voulez réaliser un site Full Flash, il est fortement recommandé de créer une version alternative en HTML et de proposer un lien vers cette version sur la page d'accueil. Ainsi, les robots pourront lire et parcourir les pages et les indexer, le site Full Flash n'étant destiné qu'à contenter les visiteurs accédant au site par la page d'accueil.

Évitons à tout prix d'utiliser des menus en Flash codés avec de l'ActionScript, nous risquons tout simplement de ruiner tous nos efforts d'indexation, bien qu'un bon plan de site, un fichier Sitemap et des liens connexes puissent réduire ce type de problème. Il peut être intéressant de faire un rappel du menu dans le pied de page dans ce cas avec des liens classiques, c'est ergonomique en général et cela permet de limiter les risques de non-indexation.

Enfin, nous insérons souvent des contenus multimédia provenant de plates-formes vidéo telles que YouTube, Wat ou Dailymotion. En HTML, cela se traduit par l'usage du couple de balises `<object>` et `<embed>` ou plus récemment par `<iframe>`. Les deux premières balises existent de longue date mais aucune n'est compatible avec tous les navigateurs, c'est pourquoi nous devons les coupler pour résorber la faille. En revanche, `<iframe>` est une méthode courte et compatible qui peut vous ravir, c'est la raison pour laquelle cette solution est souvent proposée par défaut dans les options d'intégration.

Le problème de l'insertion du Flash au milieu des contenus classiques est le manque de valeur qui lui est attribué car les moteurs ne lisent presque pas voire pas du tout le contenu. De ce fait, ces documents sont intéressants pour les visiteurs mais totalement déréférencés et sans valeur ajoutée pour le reste des pages concernées. Il convient alors de procéder à un ou plusieurs des « pansements » suivants pour valoriser ces contenus multimédia :

- insérer du texte alternatif directement entre les balises `<object>…</object>` ou `<iframe>…</iframe>` pour ajouter de la valeur aux contenus ;
- utiliser les balises `<noembed>` à côté du bloc `<embed>` pour insérer un texte alternatif lu par les robots.

Figure 3-10

Exemple de code d'intégration proposé sur YouTube

Et les balises <noframes> ?

Il existe une balise `<noframes>` mais elle n'est pas liée à `<iframe>` et le W3C l'a rendu obsolète avec HTML 5. Ne faites pas l'erreur de l'utiliser en dehors de l'HTML 4 et dans un autre contexte que les `<frameset>`.

Le code suivant montre une alternative textuelle pour donner de la valeur aux contenus multimédia et à la page web pour les moteurs.

```
<iframe width="640" height="360" src="//www.youtube.com/embed/bc-1CVc3GRg" frameborder="0"
allowfullscreen>
Texte de remplacement lu par les moteurs de recherche.
</iframe>
```

Ajax et JavaScript non optimisés

Nous avons déjà observé dans les parties précédentes que des codes en JavaScript peu ou mal optimisés peuvent causer des problèmes de lecture pour les robots. Le langage Ajax étant fondé sur JavaScript et XML, il subit les mêmes conséquences et posent ses propres problèmes en matière de compréhension par les crawlers.

En réalité, deux principaux soucis se posent lorsque nous utilisons JavaScript ou Ajax :

- des liens sont incompris voire illisibles pour les robots et ne peuvent donc pas être suivis ;
- des contenus sont dissimulés dans les scripts Ajax et donc non pris en compte par les moteurs de recherche.

En effet, l'avantage de l'Ajax est de pouvoir charger des contenus sans forcer le rechargement des pages web, ce qui confère une grande liberté aux utilisateurs mais aussi un confort d'utilisation sans faille.

En revanche, le fait de charger tout ou partie des pages web en fonction des actions de l'internaute (clic, survol, etc.) peut causer des pertes de lisibilité ou de visibilité auprès des robots.

Prenons l'exemple le plus courant en Ajax, celui du chargement dynamique des contenus via un clic ou de façon automatique avec un scroll à la manière de Twitter. Si nous regardons de près, nous observons qu'au chargement de la page, quelques dizaines de tweets sont chargés et une fois un certain palier atteint, Twitter charge en Ajax un autre groupe de tweets, et ainsi de suite.

Figure 3-11

Chargement automatique des tweets en Ajax via une action de la barre de défilement

La conséquence de ce type de chargement est assez évidente : pour la plupart, les outils de recherche ne peuvent lire que le premier groupe de tweets lorsqu'ils parcourent la page mais ils perdent tous les autres contenus. Bien entendu, l'exemple de Twitter est particulier car il contient des milliers de tweets et sa méthode a été réfléchie pour éviter que ce problème perdure (ce que nous tenterons d'expliquer par la suite).

Dans le cas du JavaScript classique, il faut garder à l'esprit que l'idéal est de ne jamais concevoir une navigation avec des liens dans ce langage, sauf si votre souhait est justement de dissimuler des contenus ou de bloquer les robots. Nous devons imaginer qu'un site devrait être consultable même si JavaScript était désactivé dans le navigateur, bien que cela soit devenu une utopie de nos jours tant jQuery et JavaScript sont présents dans les pages web.

Ajax donne davantage de fil à retordre car les cas de figure sont nombreux et quasiment tous différents. La première idée qui nous vient à l'esprit serait de charger l'intégralité des contenus utiles au chargement de la page et un script ne ferait qu'afficher des parties supplémentaires au fur et à mesure de notre parcours.

Ceci fonctionnerait parfaitement mais si nous possédons un grand nombre de contenus, nous risquons fortement de surcharger la page mais aussi de ralentir la vitesse de chargement pour les visiteurs. Cette solution est donc peu envisageable dans une majorité de cas.

Depuis le 14 octobre 2015, Google a indiqué aux webmasters que le moteur possédait une nouvelle méthode pour lire les contenus gérés via l'Ajax (source : http://goo.gl/CDBl2s). Dorénavant, les robots sont capables de lire le CSS et les fichiers JavaScript, donc de suivre bon nombre de procédures en Ajax. Cela signifie que le problème de lecture des contenus pourrait disparaître à l'avenir.

Certes, il ne faut pas encore être aussi catégorique et croire que tout l'Ajax est parfaitement lu, mais de nombreux efforts sont à noter. Google précise que les anciennes méthodes évoquées dès 2009 (source : http://goo.gl/DanW3g) sont obsolètes mais qu'elles peuvent encore fonctionner si elles sont déjà en place. Dorénavant, il convient de ne surtout pas bloquer les ressources CSS et JS pour que Googlebot puisse lire ces fichiers et les interpréter. Ainsi, l'Ajax peut être appliqué et les contenus mieux indexés.

La meilleure prise en compte des contenus est une vraie bonne nouvelle pour les webmasters, mais sur le plan de l'indexation, cela signifie que des pages uniques à rallonge risquent de noyer leurs mots-clés dans la masse. Le positionnement des pages peut donc être affecté dans certains cas, bien que ce ne soit certainement pas un problème majeur en règle générale.

Google a fait d'énormes progrès pour crawler les pages en Ajax, mais les autres moteurs restent à la traîne. Nous allons d'ailleurs étudier certaines techniques par la suite pour améliorer l'indexation, notamment sur Bing et le moteur russe Yandex. En effet, Yandex préconise encore la technique d'échappement des URL appelée *Headless Browser* (source : https://goo.gl/Z6lAjU), abandonnée par Google en octobre 2015, tandis que Bing favorise idéalement le recours à HTML 5 et JavaScript via la méthode *pushState* (source : https://goo.gl/n7iU7W).

Ajax et URL canoniques

Une technique permet de détourner l'Ajax à l'aide d'URL canoniques et de contenus dupliqués. En effet, il est possible de créer volontairement une page en double constituée de l'ensemble des contenus qui devraient être chargés en Ajax (quand cela est possible).

La page doublon pourrait être indexée par Google dans son intégralité et rediriger vers la page initiale réalisée en Ajax. Pour ce faire, il suffit d'ajouter une balise `<link/>` canonique pointant vers la page originale pour éviter le problème des contenus dupliqués, puis d'appliquer une redirection permanente (301). L'objectif serait de donner de la valeur à la nouvelle page « orpheline », de retransmettre cette valeur vers la page réelle constituée en Ajax grâce à la redirection 301, puis d'éviter le problème du *duplicate content* avec la balise canonique.

Il ne s'agit pas d'une méthode parfaite, elle peut même s'apparenter parfois à du bricolage plutôt que de l'optimisation, mais elle permet vraiment de mieux indexer et positionner les pages de contenus dans bon nombre de moteurs. Toutefois, les améliorations majeures du crawl de l'Ajax par Google devraient nous éviter d'utiliser de telles méthodes.

La technique du headless browser

Google a proposé très tôt une solution appelée *headless browser* pour éviter que l'Ajax soit un problème majeur pour l'indexation et le positionnement des pages (source : http://goo.gl/VkS5ah). En effet, une URL

classique ressemble à http://www.monsite.com/mapage.html alors qu'en Ajax, elle diffère et prend plutôt la forme suivante : http://www.monsite.com/#mapage. Ainsi, les moteurs ne peuvent pas accéder au contenu ni indexer la page, excepté Google depuis octobre 2015 dans de nombreux cas.

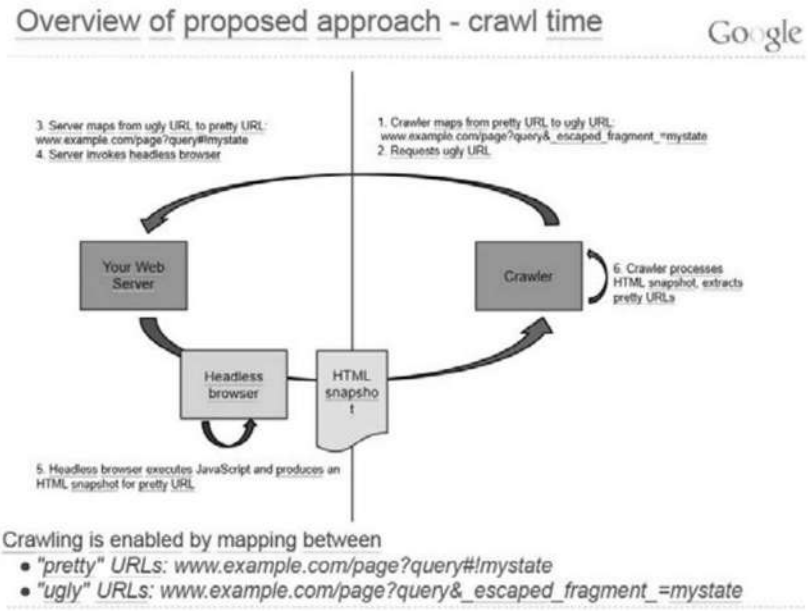
Google a trouvé une parade qui consiste à ajouter un point d'exclamation dans les URL pour les rendre indexables. Il s'agit d'une technique d'échappement et, dans ce cas, l'URL de notre exemple devient alors http://www.monsite.com/#!mapage. Cela indique au moteur qu'il va pouvoir la lire comme un utilisateur lambda.

Dans les faits, l'adresse web est modifiée et prend un paramètre intitulé `_escaped_fragment_` en lieu et place du `#!` de l'Ajax. De ce fait, l'URL est prise en compte par le moteur pour lire la page de façon classique et elle ressemble à la suivante :

```
http://www.monsite.com/_escaped_fragment_=mapage
```

Figure 3-12

Technique du headless browser schématisée par Google



En réalité, tout n'est pas si simple car la technique du headless browser demande une configuration complexe du serveur, ce qui est tout bonnement impossible sur la majorité des hébergements web (notamment les mutualisés). Nous n'entrerons pas dans les détails de l'installation de HtmlUnit ou de Jetty pour faire fonctionner cette technique, mais sachez que la méthode n'est pas toujours évidente à mettre en œuvre...

La technique viable de l'HTML 5

Aucune technique n'est parfaite pour résoudre les problèmes posés par l'Ajax, c'est pourquoi le meilleur conseil est de l'utiliser avec parcimonie et prudence, notamment si votre site est encore intégré avec d'anciennes versions des langages HTML et xHTML.

HTML 5 s'est développé en parallèle des progrès réalisés avec l'Ajax, et il n'est donc pas anodin de voir certaines fonctionnalités implémentées par défaut, notamment l'objet `pushState` qui pourrait sauver bien des référenceurs frustrés par des contenus générés en Ajax. En effet, HTML 5 a introduit une nouvelle fonction visant à générer un historique de navigation fonctionnel avec l'Ajax, ce qui permet par exemple d'utiliser les boutons *Précédent* et *Suivant* sans revenir sur la page précédente mais bien à l'état précédent de la page...

La fonction complète s'écrit sous la forme :

```
history.pushState(data, title, URL);
```

Globalement, cela signifie que les données (`data`) sont reliées à une URL et un titre donnés (`title`), ce qui permet de notifier toutes les informations dans un historique qui peut être parcouru par les usagers et les robots. En effet, les URL sont changées « en dur » dans le navigateur donc elles deviennent lisibles pour tous les robots, ce qui est bien plus performant et simple à mettre en place que la technique du headless browser. De nombreuses ressources sont disponibles sur la Toile pour mettre en place l'historique de navigation en Ajax, notamment les articles publiés sur les sites moz.com (source : http://goo.gl/olfWYi) et hypnotic.pt (source : http://goo.gl/UuBDmj).

Malheureusement, la fonction `history.pushState()` ne fonctionne pas idéalement sur tous les navigateurs classiques et mobiles. Il convient de passer par des *polyfills* (petits scripts visant à rendre compatible des fonctionnalités avec d'anciens navigateurs) pour contrecarrer le problème. Voici une courte liste de ces programmes qui permet de rendre compatible la fonctionnalité et donc de créer des sites en Ajax optimisés en SEO :

- history.js – http://goo.gl/bAv7I4
- jquery-pjax – http://goo.gl/fjSDbn
- HTML 5-History-API – http://goo.gl/YnAXNQ

Certains développeurs utilisent plutôt la technique proposée par jQuery Mobile avec la navigation Ajax qui fonctionne mieux avec la fonction `$.mobile.navigate` de la bibliothèque. Vous pourrez trouver davantage d'informations à ce sujet à l'adresse suivante : http://goo.gl/15b2l6.

Si nous faisons le point, l'Ajax est un langage vraiment intéressant mais qui pose encore de nombreux problèmes d'intégration dans certains cas. Il est possible d'utiliser à bon escient les fichiers Sitemap XML et autres techniques pour forcer l'indexation des pages bien que l'entièreté des contenus risque fortement de ne pas être lue. En définitive, la meilleure technique consiste à opter pour HTML 5 et un polyfill associé pour contrecarrer le problème.

Cookies et sessions

Nous terminons notre tour des facteurs bloquants par les sessions et cookies que nous retrouvons fréquemment dans les sites. En effet, ces deux procédés peuvent causer des soucis d'indexation et de lecture pour les robots, il faut rester mesuré quant à leur usage.

Les sessions, notamment utilisées en PHP avec la variable superglobale `$_SESSION\['…']`, permettent de mémoriser des informations cachées pendant toute la phase d'utilisation d'un site web. En effet, une fois une session ouverte dans le navigateur, les informations stockées dans les variables associées seront conservées jusqu'à la fermeture de la fenêtre ou jusqu'à ce qu'une action clôture la session (un bouton *Déconnexion* en général).

L'avantage des sessions est de permettre une navigation continue tout en conservant des paramètres en tâche de fond, ce qui peut s'avérer pratique voire obligatoire dans certains cas (par exemple, pour savoir si un utilisateur est connecté à son compte personnel).

Parmi les spécificités des sessions, il est possible d'attribuer un identifiant unique de session pour chaque utilisateur qui visite des pages. De ce fait, un long ID, souvent appelé `SID` ou `SESSION_ID`, est généré automatiquement pour chaque visiteur. Il arrive parfois qu'il se retrouve visible dans l'URL, notamment lorsqu'un script utilise la méthode `GET` pour transmettre les données. L'URL suivante montre un exemple d'adresse contenant une session avec identifiant unique :

```
http://www.site.com/page.php?id=12&SID=6bac5f8e
```

Le problème causé par les identifiants de session est double. D'une part, ces suites de caractères peuvent être longues et donc illisibles par les robots, ce qui entraîne une non-indexation des pages. D'autre part, l'URL qui comporte un paramètre pour des sessions et celle qui n'en contient pas peuvent être les mêmes, il s'agit alors de contenus dupliqués.

Plusieurs techniques permettent d'éviter des problèmes causés par les sessions, mais toutes ne sont pas toujours applicables. Il convient donc de tester au cas par cas :

- utiliser plutôt la méthode `POST` que la méthode `GET`, auquel cas les informations ne sont pas révélées dans les URL ;
- opter pour des identifiants de session à générer soi-même afin qu'ils restent courts et lisibles par les robots ;
- essayer de n'utiliser les identifiants de session qu'en cas de force majeure (système de connexion ou de vente en ligne, par exemple, seulement lorsque c'est nécessaire) ;
- faire de la réécriture d'URL pour nettoyer éventuellement les identifiants de sessions, bien que cette technique ne soit pas toujours adéquate.

Il peut être opportun également d'utiliser les URL canoniques que nous avons déjà évoquées. En effet, Google autorise l'utilisation d'une balise spécifique pour lui indiquer qu'elle est l'URL mère à indexer et donc que toutes les autres basées sur la même forme initiale soient ignorées. Dans ce cas, il faudrait indiquer à l'URL de base qu'elle est canonique, comme dans l'exemple suivant (code à placer dans la section `<head>` de la page) :

```
<link rel="canonical" href="http://www.site.com/page.php?id=12" />
```

Pour aller plus loin, nous pourrions insister en indiquant dans le fichier `robots.txt` qu'il ne faut pas indexer les pages qui contiennent le paramètre `SID` ou `SESSION_ID`, par exemple :

```
user-agent: *
# Interdire l'accès aux sessions PHP
disallow: /*SID=*
disallow: /*SESSION_ID=*
```

Enfin, nous pourrions utiliser l'adresse parente dans le fichier `sitemap.xml` pour préciser à Google quelle adresse nous souhaitons indexer. Le mélange de ces trois phénomènes devrait régler les problèmes d'indexation dans le cas d'URL contenant des identifiants de sessions uniques.

Sur le même principe, les cookies permettent de récupérer des paramètres mais cette fois-ci en les enregistrant dans des petits fichiers stockés directement sur la machine des internautes. En général, les cookies ne posent pas de problèmes dans les navigateurs car ils sont acceptés par défaut, mais pour les robots, c'est une autre histoire car ils ne peuvent pas forcer l'acceptation des cookies et donc leur lecture, ce qui peut s'avérer bloquant dans bien des cas.

La méthode est relativement simple, il faut proposer une alternative aux internautes lorsque les cookies ne sont pas acceptés. De ce fait, les moteurs ont également accès à cette page de secours et peuvent donc continuer leur parcours si elle est bien conçue.

Dans l'idéal, il convient de créer une page d'erreur personnalisée pour ce genre de cas dans laquelle nous proposons un plan de site, un lien vers le plan du site ou dans le pire des cas un lien retour vers l'accueil. L'objectif est de ne pas bloquer l'utilisateur ni le robot pour que le crawl puisse continuer normalement.

Créer un cookie est aisé en PHP comme le montre le code suivant :

```
<?php
// Fonction setcookie avec nom, texte et durée de vie (1 heure ici)
setcookie('nom_du_cookie, 'texte du cookie', (time() + 3600));
?>
```

D'autres paramètres de sécurité peuvent être ajoutés pour préciser le répertoire voire le domaine sur lequel le cookie est utilisé mais globalement, cela reste simple à mettre en place. Il est toutefois important de préciser que nombre de cookies contiennent des données secrètes qui peuvent être récupérées par des personnes malintentionnées. C'est pourquoi les développeurs ne doivent pas toujours les utiliser pour passer des données à protéger.

Pour récupérer des informations émanant de fichiers de cookies, il faut utiliser la variable `$_COOKIE['…']` en appelant le nom du cookie (et dans certains cas le nom de son paramètre). Sachant cela, nous pouvons vérifier si les cookies sont acceptés ou non en procédant à une simple vérification de lecture du fichier, comme dans le code suivant. Si les cookies ne sont pas acceptés, ils ne pourront pas être lus et nous serons redirigés vers une page d'erreur, par exemple.

```
// Si le cookie n'existe pas, redirection vers une page d'erreur
if(empty($_COOKIE["nom_du_cookie"])) {
header("location:erreur-cookies.php");
exit();
}
```

C'est dans cette page d'erreur qu'il sera conseillé de proposer un plan de site ou une autre alternative pour que les robots ne soient pas bloqués dans leur parcours d'indexation.

Typologie des pénalités

Depuis les premiers temps des moteurs de recherche, il existe des solutions pour pénaliser les sites web qui abusent des critères de lecture des robots d'indexation. En effet, les moteurs de recherche, Google en tête, n'apprécient guère d'être dupés par les webmasters qui profitent des faiblesses apparentes des robots pour être mieux classés dans les résultats de recherche.

Il existe toutes sortes de causes sur lesquelles nous reviendrons par la suite, nous avons d'ailleurs déjà évoqué Google Panda et Penguin qui représentent sûrement les exemples les plus connus d'actions anti-spam. Mais voyons surtout ce que nous risquons lorsque nous suroptimisons nos contenus et nos pages HTML.

Différencier les sanctions manuelles ou algorithmiques

Tout d'abord, il est important de distinguer les pénalités infligées manuellement par des humains de celles gérées automatiquement par les serveurs des moteurs de recherche (ou par les robots). Le fait d'être sanctionné ne révèle pas toujours d'une cause évidente trouvée lors de l'indexation. Il arrive de plus en plus fréquemment que d'autres aspects provoquent des pénalités :

- délation et plainte de la part d'autres internautes ;
- effets de bord provoqués par des connexions avec d'autres sites pénalisés ;
- erreurs humaines.

Dans la réalité, la grande majorité des pénalités sont stimulées par des erreurs grossières ou des actes de duperie causées par les développeurs ou référenceurs. Dans ces cas précis, ce sont souvent les robots qui détectent les techniques frauduleuses et qui entraînent des pénalités immédiates.

Pour le reste, les humains interviennent lorsque les robots ne découvrent pas les supercheries, c'est notamment le cas si une plainte est déposée ou si un site majeur a été pénalisé. En effet, il faut alors étudier le site en détail pour voir s'il est réellement suroptimisé ou s'il a des liens forts avec un site déjà sanctionné. Ainsi, les humains peuvent jauger le degré de pénalité à infliger ainsi que la durée des sanctions.

Pour suivre l'actualité au sujet des pénalités, consultez régulièrement les informations divulguées par Matt Cutts, responsable de l'équipe Webspam de Google. Parallèlement, certains brevets sont déposés de temps en temps pour officiellement expliquer les mécanismes mis en œuvre pour lutter contre le spam, notamment le brevet déposé le 5 mars 2013 (mais antérieur en réalité) et intitulé « Systems and Methods for Detecting Hidden Text and Hidden Links » (source : http://goo.gl/ddQlyf). Il explique comment Google lutte fermement contre le spamdexing (suroptimisation des pages).

Attention au spam à répétition

Google fait la chasse au spam depuis des années, mais compte durcir encore davantage les sanctions pour les récidivistes (source : http://goo.gl/lCfduK). En effet, il arrive encore trop souvent que des webmasters arrivent à sortir d'une pénalité manuelle ou algorithmique et qu'ils refassent les mêmes erreurs dans les mois suivant la levée de sanction (volontairement ou non). Dans ce cas, Google peut durcir les demandes de réexamen et surtout les pénalités appliquées. Il faut donc être prudent et ne pas trop jouer avec le feu…

Sandbox

La notion de « sandbox » a été très employée il y a plus d'une décennie pour parler des sites mis en quarantaine temporairement par Google. Des webmasters avaient remarqué qu'il arrivait parfois que des sites ayant beaucoup de backlinks dès leur création pouvaient être détectés par Google comme frauduleux. Aussi, la triche n'étant pas réellement mesurable, les sites concernés étaient placés dans des « bacs à sable » *(sandbox)* durant une à plusieurs semaines.

En réalité, les sites web étaient touchés dans leur globalité dans ce cas et non uniquement certaines pages en particulier. Après une courte période de quarantaine, les sanctions étaient levées et les sites pouvaient occuper leur position méritée.

Il est très difficile de savoir si la sandbox a réellement existé ou si elle n'est pas une légende urbaine qui a semé le trouble pendant quelques années dans la sphère SEO, c'est d'ailleurs pour cette raison que nous en parlons au passé.

De nos jours, cet effet de quarantaine ne semble plus appliqué mais surtout plus applicable tant les mécanismes de crawl ont évolué. Nous pouvons aussi douter de l'intérêt d'une pénalité temporaire de ce type depuis l'implantation de Google Panda et Penguin.

Ces derniers sont automatiquement lancés lorsque les pages sont scrutées par les robots, et d'après nos connaissances, soit les pages sont suroptimisées et donc sanctionnées, soit elles ne le sont pas et elles peuvent mener leur vie virtuelle. De ce fait, quel serait l'intérêt d'une mise en quarantaine ?

Rien ne nous permet aujourd'hui d'affirmer que la sandbox a existé ou qu'elle demeure active, mais force est de constater que les témoignages concernant des sites mis en quarantaine deviennent quasi inexistants depuis quelques années. Nous pouvons donc sûrement en déduire que la sandbox a rendu l'âme auprès des référenceurs et des moteurs…

Baisse de PageRank

Google a trouvé une parade intelligente pour lutter contre les campagnes de netlinking abusives et les ventes de liens *(paid linking)* en abaissant plus ou moins le PageRank des pages web jugées comme frauduleuses.

Cette pénalité n'est pas la plus sévère qui soit puisqu'elle n'engendre pas vraiment de chutes massives et irrécupérables dans les SERP. Elle est plutôt à prendre comme un avertissement avant une sanction plus lourde de conséquences.

Pour les sites qui abusent du netlinking, ce type de sanction peut être un coup de massue car il appuie sur le seul curseur valorisant pour les référenceurs. Une fois la baisse effective, il devient bien plus difficile de convaincre quelqu'un pour des échanges ou des ventes de liens. Mais en général, retenons qu'il s'agit d'une pénalité visant à interdire la vente de liens et qui n'a pas d'incidence majeure sur le positionnement des pages dans les SERP.

Déclassement

Il arrive parfois que des pages web soient déclassées dans les SERP sur des requêtes précises. Dans ce cas, seules les pages suroptimisées ou frauduleuses sont touchées et non le site au complet. Il s'agit certainement du type de pénalités le plus fréquent.

Nous connaissons ces pénalités sous l'appellation « minus 30 » ou « minus 60 » qui correspondent à des pertes de positionnement qui ont pour conséquence de ramener des pages à la 31e ou 61e place des résultats de recherche, autrement dit de les rendre quasi invisibles pour les internautes.

Certains forums et blogs mentionnent même d'autres pénalités telles que la « minus 50 » ou encore la « position 6 penalty » qui vise à abaisser une page en 6e position juste en dessous de la ligne de flottaison afin de baisser considérablement son trafic quotidien.

Il est important de ne pas confondre les pénalités et les mouvements « naturels » des moteurs de recherche. Il peut arriver de temps à autre que des pages chutent drastiquement dans les SERP sans pour autant qu'il s'agisse d'une sanction. En effet, lorsque des mises à jour de l'algorithme se produisent, aussi infimes soient-elles, il peut arriver que des pages web « disparaissent » des moteurs temporairement.

En règle générale, il faut utiliser la commande `site:` du moteur pour suivre si les pages sont bien indexées et si tel est le cas, alors il faut tester à nouveau les requêtes phares censées faire ressortir les pages dans les résultats de recherche après quelques jours de patience. S'il s'agissait d'une mise à jour, les pages auront repris plus ou moins leur position habituelle, mais si ce n'est pas le cas, l'inquiétude peut être de mise.

Soyons honnêtes, nous savons généralement quand nous sommes pénalisés à partir du moment où nous pensons suroptimiser les contenus voire tricher délibérément. Si vous respectez au plus près les *guidelines* des moteurs et que vous disparaissez des SERP, il est fort probable que cela ne soit que temporaire…

Liste noire

Dans la liste des pénalités, la liste noire *(black list)* est sans hésiter la plus sévère de toutes puisqu'elle consiste à supprimer entièrement le site web dans sa globalité de l'index du moteur. Ce type de sanction signe souvent la mort partielle ou définitive des sites concernés, mais elle n'est appliquée que dans des cas vraiment importants. En effet, les moteurs de recherche ne s'amusent pas à sanctionner si fermement uniquement pour le plaisir…

Il est important de se méfier de la liste noire, car beaucoup de webmasters sont pris de panique lorsque des mouvements sont visibles dans les SERP. Comme pour le déclassement, il ne faut pas toujours s'affoler

lorsque nous ne trouvons plus certaines pages dans l'index, de multiples raisons peuvent entraîner ce phénomène.

Pour vérifier si un site a été durement sanctionné, il faut utiliser la fonction `site:` sur Google et Bing, par exemple, car elle permet d'afficher toutes les pages indexées. Par exemple, nous pouvons taper les commandes suivantes dans le champ de recherche pour vérifier respectivement les pages web indexées pour le site principal et le blog associé (sous-domaine) :

```
site:www.monsite.com
site:blog.monsite.com
```

Si aucun résultat n'est affiché pour le site ou le blog, alors le nom de domaine a été entièrement sanctionné. Il est également possible que seule l'une des deux parties soit touchée par la pénalité.

Comment faire pour sortir d'une pénalité Google ?

Les sites pénalisés sont souvent le résultat d'une triche délibérée ou d'une action anormale jugée négativement par les robots d'indexation. Les moteurs ne sanctionnent pas leurs « clients » pour le plaisir mais bien pour des raisons qu'ils jugent évidentes. Dans la très grande majorité des cas, les sanctions tombent et surprennent les « spécialistes-victimes », mais nombre d'entre eux reconnaissent avoir peut-être abusé grossièrement lors des optimisations et du netlinking.

Toutefois, il persiste des cas pour lesquels nous parlons plutôt d'effets de bord. Par exemple, le vendeur de liens Buzzea a été sanctionné en janvier 2014 après avoir été rattrapé par la patrouille de Google, mais aussi tous les sites qui ont eu des liens plus ou moins effectifs avec ce dernier, ce qui signifie que des effets collatéraux peuvent apparaître, même si cela semble logique ici.

Dans bien d'autres cas, nous sommes surpris de perdre des positions dans les SERP alors qu'aucune pénalité réelle ne semble avoir été appliquée. Ces cas sont plus fréquents que nous le pensons et force est de constater que nous ne pouvons pas faire grand-chose pour lutter, la meilleure solution est souvent de vérifier les pages touchées, de modifier quelque peu le contenu et de repartir de plus belle en espérant que l'effet de bord s'estompe.

Depuis quelques années, Google et Bing adressent des messages par le biais de leurs interfaces Webmaster Tools et Webmaster Center afin de prévenir les administrateurs des éventuelles sanctions qui ont été infligées. La figure 3-13 montre par exemple un message envoyé à des sites qui ont été en liaison étroite avec le réseau de liens Buzzea.

Il faut savoir qu'un message n'est pas automatiquement envoyé mais cela devient de plus en plus fréquent. La première solution se résume souvent à réfléchir à ce qui aurait pu provoquer la sanction, sauf si un message précise clairement le problème. Ensuite, il est important de vérifier avec précision le niveau d'indexation des pages avec la commande `site:` mais aussi avec les outils d'aide comme les Webmaster Tools.

Selon le type de sanctions subies, les résultats peuvent être différents. Par exemple, une mise en liste noire va nécessairement provoquer une absence totale car les pages seront introuvables dans l'index.

Cela pourra être causé par une pénalité ou par des erreurs humaines. En effet, il arrive parfois que nous fassions des erreurs et que nous incriminions directement les moteurs alors que tout est de notre faute. Par exemple, nous savons grâce à Eric Kuan (source : http://goo.gl/QT36ts) qu'un fichier `robots.txt` qui existe et qui est mal rempli bloquera totalement le crawl des robots (en d'autres termes, si le fichier retourne une erreur autre que 200 ou 404).

Figure 3-13

Message envoyé par la Google Search Console à propos de liens factices détectés sur un site

Google

http://www.monsitedetriche.com : liens sortants factices

Nous avons détecté un système de liens factices ou artificiels sur ce site. La vente de liens ou la participation à des systèmes de liens dans le but de manipuler le classement PageRank constitue une infraction aux Consignes aux webmasters de Google

Suite à la détection de liens artificiels provenant de votre site, nous avons appliqué une action manuelle pour cause de spam à monsitedetriche.com/ . D'autres actions peuvent être appliquées à l'ensemble ou à certaines parties de votre site.

Actions recommandées

- Identifiez les liens payants ou factices à l'aide de l'attribut rel="nofollow" ou en redirigeant vos liens vers une page intermédiaire bloquée par un fichier robots.txt.
- Supprimez tous les liens problématiques de votre site.
- Une fois que vous estimez que votre site est conforme aux consignes aux webmasters de Google, envoyez-nous une demande de réexamen.
- Pour obtenir une liste à jour des actions manuelles actuellement appliquées à votre site, consultez la page Actions manuelles. Si aucune action manuelle n'est répertoriée, il n'est plus nécessaire de déposer une demande de réexamen.

Si nous considérons que votre site n'enfreint plus nos consignes, nous annulerons l'action manuelle.

Pour toute question relative à la résolution de ce problème, consultez le Forum d'aide pour les webmasters.

Vous souhaitez nous faire part de vos commentaires ? Rendez-vous ici. Assurez-vous d'inclure l'identifiant du message : [WMT-92503]

Google Inc. 1600 Amphitheatre Parkway Mountain View, CA 94043, États-Unis | Se désabonner

Lorsqu'il s'agit de déclassements, cela peut être dû à une mise à jour des algorithmes, comme lors de l'arrivée de Google Panda ou Penguin, mais aussi à d'autres modifications mineures. Les SERP fluctuent régulièrement et il n'est pas rare de gagner ou perdre quelques positions, ce n'est pas toujours la conséquence d'une sanction ou d'une mauvaise action de notre part.

En revanche, il faut toujours se poser les bonnes questions lorsqu'un déclassement survient.

- Avons-nous modifié les contenus internes des pages déclassées ?
- Avons-nous trop optimisé le code HTML et les contenus ?
- Comptons-nous trop de liens factices ?
- Est-ce que certains contenus sont des copies dupliquées d'autres pages ?
- Sommes-nous touchés par une tentative de *negative SEO* (voir section éponyme en fin de chapitre) ?

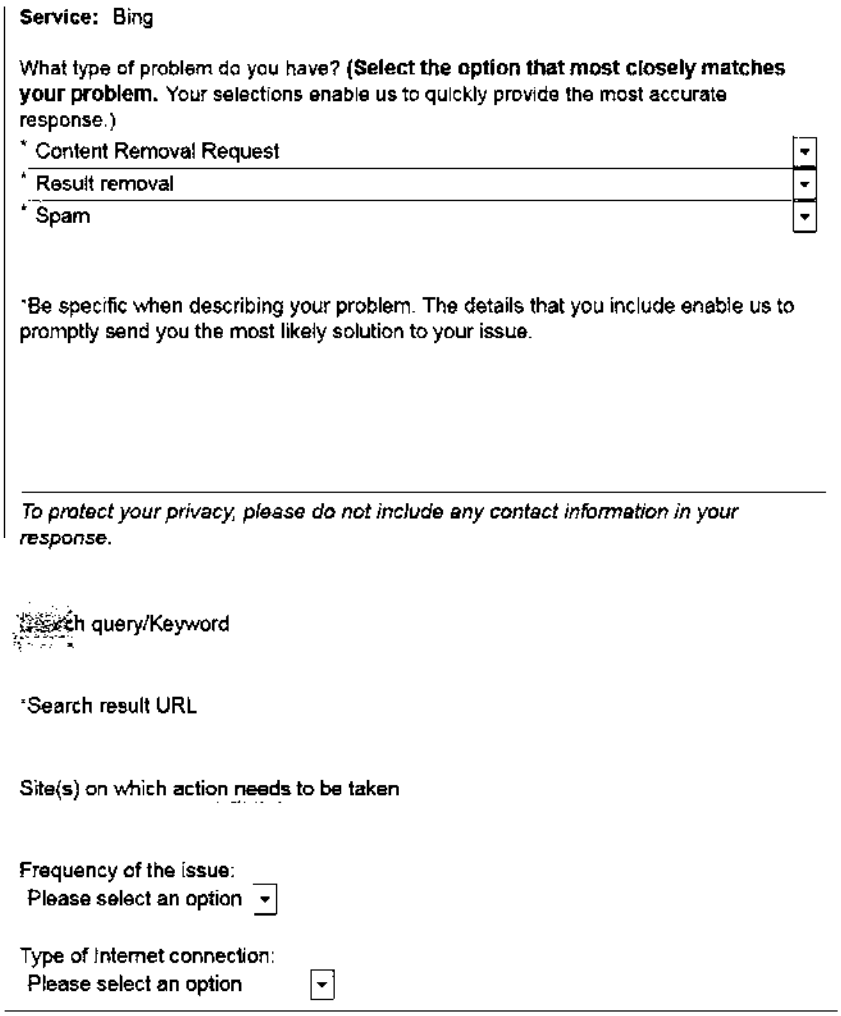
Si les réponses à ces quelques questions s'avèrent positives alors vous comprendrez aisément pourquoi des pages ont perdu leur positionnement initial.

Enfin, si le site global a totalement disparu de l'index, il est fortement probable qu'un problème technique lié au serveur soit en cause. Il faut alors se renseigner auprès de son administrateur ou de son hébergeur pour comprendre le problème. Toutefois, nous avons vu également qu'un mauvais fichier `robots.txt`, un fichier `.htaccess` erroné ou encore une succession de liens morts peuvent entraîner des conséquences identiques.

Une fois le diagnostic effectué, il convient de nettoyer les erreurs éventuelles afin que les pages web récupèrent les positions qu'elles méritent. S'il s'agit d'erreurs personnelles (problème de serveur, de fichier `robots.txt`...), il n'est pas nécessairement utile d'agir. Il faut souvent attendre quelques temps après avoir renvoyé un fichier `sitemap.xml` ou procédé à une nouvelle suggestion d'URL, par exemple. En revanche, si des sanctions sont à l'origine des chutes voire des disparitions dans les SERP, il est indispensable de supprimer toutes les suroptimisations et les liens factices le plus rapidement possible.

Figure 3-14

Formulaire de demande de réexamen de Bing



Une fois le nettoyage de fond effectué, il est recommandé d'effectuer une demande de réexamen (source : http://goo.gl/qYPXU7) auprès des moteurs. Il s'agit en fait d'un court formulaire dans lequel nous expliquons le problème rencontré et ce que nous avons fait pour résoudre le contentieux. Il faut savoir que Google laissera toujours un certain temps de pénalité avant de prendre en compte la demande de réexamen, de l'ordre d'environ deux mois selon Matt Cutts. En d'autres termes, un site sanctionné le sera au moins pour deux mois dans la grande majorité des cas...

En général, les demandes de réexamen se font via la Google Search Console ou le Webmaster Center de Bing, souvent en réponse à un message reçu. Elles peuvent aussi s'effectuer par demande directe de la part des webmasters. Il est important de s'exprimer clairement lors de la soumission d'une vérification, mais aussi d'être le plus honnête et transparent possible. Il ne faut pas hésiter à avouer d'éventuelles suroptimisations si vous avez été pris la main dans le sac, ou même à être totalement transparent sur la méthode employée par erreur ou non.

Ensuite, il convient d'expliquer toutes les actions qui ont été mises en œuvre pour nettoyer les actes répréhensibles et les liens factices. Une fois la demande examinée et traitée par les services de Google ou Bing, le site peut espérer reprendre des positions confortables après un laps de temps. Néanmoins, sachez que certains sites sanctionnés par Google Penguin, par exemple, n'ont jamais réussi à récupérer les positions qui étaient les leurs auparavant. Réparer ses erreurs n'est pas toujours synonyme « d'excuser » pour Google, et parfois, il faut des mois pour pouvoir reprendre de bonnes positions dans les SERP, alors gare aux tricheries...

Quelques causes de pénalités

Les moteurs de recherche ne pénalisent jamais au hasard, il faut toujours une raison logique ou être dénoncé pour être pris dans la tourmente des sanctions. Beaucoup de référenceurs ont tendance à voir des pénalités à tout bout de champ mais dans la réalité, ce sont surtout les sites les plus « spammeurs » qui se font toucher rapidement. Cela ne veut pas dire que les sites qui abusent peu passent au travers des mailles du filet, les robots sont de plus en plus efficaces et certains abus sont détectés lors de l'indexation, donc nous ne pouvons pas les éviter...

Spamdexing

Principe général et brevet

Le référencement abusif *(spamdexing)* correspond à un ensemble de techniques qui permettent de dissimuler des textes et des liens optimisés uniquement pour être mieux positionné. De multiples méthodes en HTML, CSS voire JavaScript sont à notre disposition pour duper les robots d'indexation en affichant des zones optimisées spécifiquement pour eux que nous rendons invisibles aux internautes qui visitent le site.

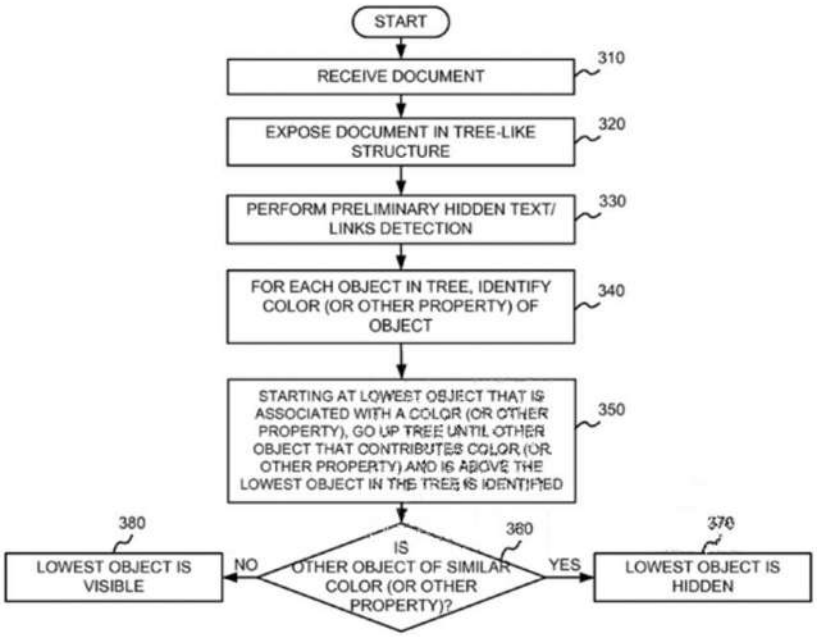
Le brevet US8392823 B1 intitulé « Systems and Methods for Detecting Hidden Text and Hidden Links » résume parfaitement comment Google lutte contre le référencement abusif (source : http://goo.gl/ddQlyf).

En effet, nous savons avec certitude que la détection des liens cachés et des textes dissimulés se fait dès l'indexation lors du parcours des robots.

Dans les faits, les robots décortiquent l'arbre HTML (le « DOM ») des pages pour étudier leur structure sémantique et hiérarchique. Si des anomalies flagrantes sont repérées lors de l'analyse, le spamdexing est sanctionné. En d'autres termes, si les référenceurs usent des CSS pour afficher du texte blanc sur fond blanc, par exemple pour le masquer aux humains, la supercherie est détectée grâce au robot et la page est sanctionnée ou non en conséquence.

Figure 3-15

Procédé de détection du spamdexing selon Google



Voici quelques cas de spamdexing courants ou énoncés dans le brevet de Google :

- répétitions abusives de balises valorisantes telles que `<h1>`, `<h2>` ou `<strong>` ;
- multiplication de liens internes et externes en bas de page ;
- usage abusif des balises `<noframes>` et `<noscript>` ;
- texte de même teinte que le fond (texte blanc sur fond clair…) ;
- texte placé avec z-index sur une image dont le fond est de même couleur ;
- texte écrit en minuscule (taille de police d'un seul pixel…) ;
- lien caché derrière une image de 1 × 1 px (GIF transparent…) ;
- lien (avec ou sans image) placé derrière un bloc de contenus disposé grâce à un z-index.

Ce qu'il est important de noter, c'est que les robots sont capables de lire un peu le CSS comme le suggère le brevet, ce qui n'a jamais été totalement confirmé de la part des porte-parole de Google notamment.

Il est également précisé que tout ce qui sort du cadre de l'écran du navigateur est considéré comme du spam, ce qui exclut très clairement des techniques CSS comme les suivantes :

- `text-indent: -9999px;`
- `position:relative;` avec `left:-9999px;`
- `visibility:hidden.`

Il arrive parfois que nous fassions du spamdexing sans réellement nous en rendre compte. En effet, il n'est pas exclu lors d'une refonte graphique d'un site que nous n'insistions pas assez sur les divers contrastes et que les robots détectent certaines zones comme du référencement abusif puisqu'ils sont capables de différencier les couleurs identiques mais aussi approchantes (avec des codes hexadécimaux de la même gamme ou RVB voire HSL).

En conséquence, il est écrit dans cet ancien brevet du 4 décembre 2003, mis à jour en août 2009, que la détection de spamdexing n'entraîne pas nécessairement de sanctions. Il arrive dans les cas les moins dérangeants que les zones sensibles soient uniquement ignorées. En d'autres termes, si vous faites du spamdexing à outrance, les pages touchées seront pénalisées. Pour l'anecdote, sachez que le site de BMW avait été sanctionné en 2006 pour des actions abusives de ce type…

En revanche, si vous n'avez que quelques sections détectées comme « spammy », les robots les ignoreront et indexeront le reste des pages comme habituellement. Cela n'aura aucune incidence sur le classement dans ce cas précis sauf que vous ne bénéficierez pas des suroptimisations effectuées…

Exemple de la propriété content en CSS

La propriété `content`, présente en CSS 2.1 et CSS 3, fonctionne avec les pseudo-éléments `:after` et `:before` compatibles avec bon nombre de navigateurs exceptés Internet Explorer 8 et inférieurs. L'utilité de cette propriété est d'ajouter des contenus minimes ou des éléments graphiques avant ou après un morceau de contenu dans les pages web.

Par exemple, la propriété `content` permet d'afficher des puces personnalisées dans les listes ordonnées ou non ordonnées avant les items de liste. Il est également possible d'afficher, par exemple, l'URL des liens dans les feuilles d'impression de manière dynamique afin que les lecteurs obtiennent les informations nécessaires, de la manière suivante :

```
a:after {
    content: " (" attr(href) ")";
}
```

Globalement, cette propriété accompagne les internautes et les concepteurs de sites pour améliorer le rendu graphique des textes selon les supports et le design final. Le principal défaut de `content` est d'être limité à l'ajout de textes courts, d'images (icônes en général), d'un compteur numérique ou de guillemets. Enfin, il est impossible d'intégrer du code HTML effectif et les caractères spéciaux doivent être encodés pour être fonctionnels.

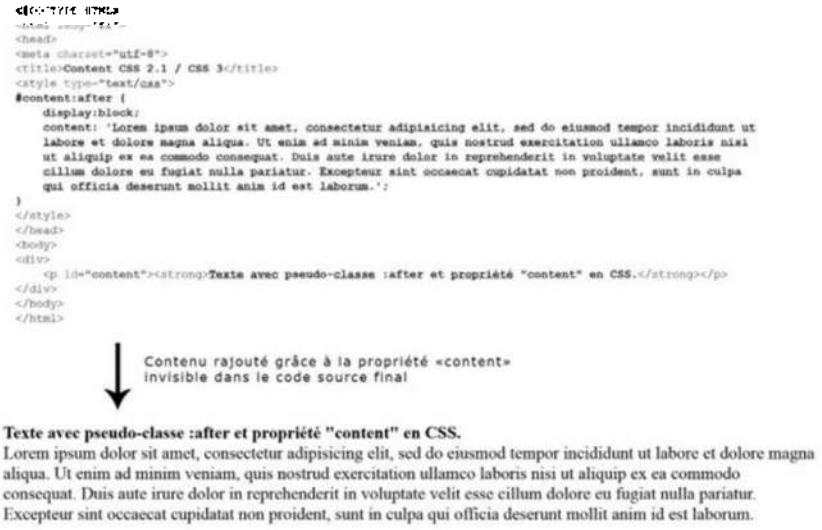
En matière de SEO, nous pouvons aisément imaginer l'intérêt d'une telle propriété, bien qu'il faille comprendre que cela soit assimiler à du spam ou du Black Hat SEO. Nous pouvons en effet ajouter des contenus que nous souhaitons afficher aux utilisateurs sans pour autant les rendre visibles aux robots d'indexation.

Si cela ne doit pas constituer la majeure partie d'un site web, il est vrai que l'astuce peut parfois aider à dissimuler des contenus qui pourraient noyer les mots-clés majeurs.

En effet, si nous prenons un texte de cent mots et qu'une trentaine d'entre eux n'ont de valeur ajoutée que pour les utilisateurs, nous pourrions très bien user de ce procédé pour laisser apparaître seulement 70 termes « forts » aux robots. Certes, cela est fastidieux et constitue un réel cas de spam, mais cette astuce doit forcément être dans les esprits. Nous devons donc la connaître au moins pour savoir la contrer si nous l'utilisons à tort !

Figure 3-16

Attention au spam avec la propriété CSS content



Particularité des textes issus de la propriété content

Les textes rédigés au sein de la propriété `content` ne peuvent pas être sélectionnés ni recopiés. Ils sont placés dans une surcouche indécelable, sauf si les moteurs de recherche arrivent à lire parfaitement les propriétés CSS (ce qui fait toujours débat de nos jours). La technique de spamdexing via cette propriété est donc facilement repérable pour les utilisateurs aguerris.

Keyword stuffing

Le *keyword stuffing*, ou bourrage de mots-clés, est chassé naturellement par les divers robots d'indexation. Les pages sont étudiées en détail et sémantiquement lors du crawl et chaque abus peut être sanctionné.

Les robots analysent la densité de chaque mot et expression au sein des pages et si des anomalies flagrantes se dégagent, des pénalités peuvent tomber. Par exemple, si une page contient cent mots mais qu'un même mot est répété dix fois, la densité est trop importante pour être naturelle et cela risque de faire tiquer les

moteurs de recherche. Qui plus est, le bourrage de mots-clés dans des zones valorisées telles que les balises `<title>`, `<strong>`, `<h1>` ou les attributs `ALT` sont facilement détectables et risquent d'être sanctionnés.

Nous ne savons pas si Google pénalise systématiquement le keyword stuffing lorsqu'il en détecte dans les pages, notamment depuis l'arrivée de Google Panda. En revanche, nous pouvons au moins imaginer que les zones suroptimisées sont ignorées à l'image du spamdexing. Dans les faits, il semble tout de même courant que des sites abusifs aient été lourdement sanctionnés après avoir appliqué des bourrages de mots-clés trop visibles pour les robots, donc restons prudents...

Cloaking

Le *cloaking* est une technique qui permet de dissimuler des contenus aux robots tout en les montrant aux visiteurs. En général, il s'agit d'utiliser des scripts, souvent en JavaScript, pour effectuer des redirections automatiques vers des contenus optimisés pour les moteurs de recherche lorsque les robots sont en phase d'indexation.

Prenons un exemple : une page peu optimisée mais graphiquement intéressante pour les clients potentiels est mise en place par un webmaster. Cette page risque fortement d'être très mal positionnée à cause d'un manque flagrant d'optimisations et de contenus textuels. Dans ce cas, il arrive que des référenceurs peu avertis préfèrent créer une page écran suroptimisée qui sera affichée pour les robots lors du crawl plutôt que la page destinée à la clientèle.

Le procédé est très simple à mettre en place, il suffit de créer une petite fonction qui distingue les robots des visiteurs classiques pour réaliser une redirection vers la page optimisée. Dans ce cas, les internautes obtiennent la page « vendeuse » tandis que les robots parcourent une page textuelle et bourrée d'optimisations idéales.

Ce type de pratiques est souvent effectué en JavaScript mais il est également possible en PHP voire dans d'autres langages. Il faut toutefois rester très prudent car les robots sont puissants et savent très bien détecter ces failles. C'est d'ailleurs pour cette raison que les redirections web sont à réaliser avec prudence depuis quelques années. Google présente dans sa documentation officielle les redirections 301 (ou redirections permanentes) pour contrer ce type de tricheries, et si nous n'optons pas pour cette méthode, il est précisé que cela risque d'être considéré comme du spam.

Le cloaking ne s'arrête pas uniquement à des redirections, d'autres techniques permettent de duper les moteurs de recherche. L'exemple du jQueryRank Sculpting (source : http://goo.gl/rwfXjv) montre comment dissimuler des liens en les transformant dynamiquement en texte par un procédé simple. En effet, le script jQuery modifie les balises HTML des liens en autre balises de notre choix (`<span>`, `<div>`, `<h2>`...) pour que les robots ne les détectent pas comme des liens lors de la lecture, mais pour que les utilisateurs puissent quant à eux profiter des vrais liens.

Le code du jQueryRank Sculpting a été créé pour montrer qu'il était encore possible de duper les moteurs de recherche avec d'autres procédés techniques. Ici, l'objectif est de développer le PageRank Sculpting sans utiliser les nofollow qui n'ont plus grand intérêt dans les liens internes, si ce n'est à faire perdre du jus de liens. Ainsi, le code permet d'afficher uniquement les liens désirés afin de transmettre le PageRank

de Google et le BrowseRank de Bing uniquement aux pages que nous souhaitons. Il est fortement déconseillé d'utiliser ce type de pratique, le code n'a été créé qu'à titre d'exemple. Le voici en détail :

```
<script type="text/JavaScript" src="jquery.min.js"></script>
<script type="text/JavaScript">
source = '.linktoggle'; // Classe (.nomClass) ou ID (#nomId)
attribut = 'title'; // Attribut qui réceptionne l'URL
newclass = 'newlink'; // Classe du lien après modification
evttag = 'hover'; // Déclencheur du script : survol (hover), clic (click)
                  // ou double-clic (dblclick)

jQuery(document).ready( function() {
    if(evttag == 'hover') {
    evenement='hover';evtJavaScript='onmouseover';evtJavaScript2='onmouseout';
    } else if(evttag == 'click') {
    evenement='click';evtJavaScript='onclick';evtJavaScript2='onmouseout';
    } else if(evttag == 'dblclick') {
    evenement='dblclick';evtJavaScript='ondblclick';evtJavaScript2='onmouseout';
    } else {
    evenement='hover';evtJavaScript='onmouseover';evtJavaScript2='onmouseout';
    }

    // Premier survol : on remplace le <span> survolé par un lien <a>
    // Cette initialisation évite d'appliquer les remplacements quand des <a> classiques
    // sont en place (on se limite aux <span> ici)
    jQuery(source).live(evenement,function() {
        // On adapte la modification en <a> en fonction des éléments d'origine (<span>, <h2>,
        // <div>...) qui portent la classe ".linktoggle"
        ElmtToggle = jQuery(this).get(0).tagName.toLowerCase();

        // On récupère les attributs existants dans la balise d'origine (rel, title, class,
        // id...)
        var arrayAttrs = [];
        for(var i=0, attrs=jQuery(this).get(0).attributes, nb=attrs.length; i < nb; i++) {
        arrayAttrs.push(attrs.item(i).nodeName+'="'+attrs.item(i).nodeValue+'"');
        }
        Attributs = arrayAttrs.join(" "); // On enregistre tous les attributs et leurs
                                          // valeurs dans une chaîne --> variable globale

        var TexteSpan = jQuery(this).text(); // On mémorise le texte contenu dans le lien
        var TexteTitle = $(this).attr(attribut); // On mémorise le texte contenu dans
                                                 // le lien

        // On utilise replaceWith() plutôt que html() car elle remplace totalement les
        // balises, elle ne les ajoute pas --> problèmes sinon !
        // On génère un appel vers une fonction qui va permettre de remettre le <span> quand
        // il n'y a plus de survol
        jQuery(this).replaceWith('<a href="'+TexteTitle+'" '+evtJavaScript2+'="jQuery(this).
        restoreSpan(ElmtToggle,Attributs);">'+TexteSpan+'<\/a>'); // On l'intègre dans des <a>
        return false;
```

```
    });

    // La fonction RestoreSpan remet le <span> initial en route (quand on quitte
    // le survol du lien)
    // On doit appeler une fonction restoreA pour autoriser à nouveau la modification du
    // <span> en <a>
    jQuery.fn.restoreSpan = function(ElmtToggle,Attributs) {
        var TexteA = jQuery(this).text(); // On enregistre le texte contenu dans le lien
        var TexteHREF = jQuery(this).attr('href'); // On enregistre le texte contenu dans
                                                   // l'attribut href --> title du <span>

        jQuery(this).replaceWith('<'+ElmtToggle+' title="'+TexteHREF+'" '+evtJavaScript+'=
        "jQuery(this).restoreA(ElmtToggle,Attributs);" '+Attributs+'>'+TexteA+'
        <\/'+ElmtToggle+'>'); // On le remet dans des <span>
    };
    // La fonction RestoreA replace le <a> désiré en cas de nouveau survol du lien
    // On appelle à nouveau la fonction restoreSpan pour retourner à l'état de départ
    jQuery.fn.restoreA = function(ElmtToggle,Attributs) {
        var TexteSpan = jQuery(this).text(); // On enregistre le texte contenu dans le lien
        var TexteTitle = jQuery(this).attr(attribut); // On enregistre le texte contenu dans
                                                      // l'attribut href --> title du <span>

        jQuery(this).replaceWith('<a href="'+TexteTitle+'" '+evtJavaScript2+'="jQuery(this).
        restoreSpan(ElmtToggle,Attributs);">'+TexteSpan+'<\/a>'); // On l'intègre dans des <a>
    };
});
</script>
```

Figure 3-17

Exemple d'utilisation du code du jQueryRank Sculpting

Le cloaking est chassé activement par les moteurs de recherche et les pénalités peuvent être très lourdes dans la plupart des cas. Il est donc conseillé d'être très vigilant et de ne pas s'amuser à tricher

systématiquement alors que des optimisations classiques et intelligentes peuvent souvent suffire pour obtenir de bons positionnements et de bonnes retombées.

Doorway ou pages satellites

Les pages satellites *(doorway pages)*, sont des pages créées de toutes pièces pour les moteurs de recherche et qui contiennent des redirections vers les pages présentées au public, souvent bien moins optimisées pour le référencement. Nous retrouvons ici une des techniques souvent réalisée avec le cloaking et qui est chassée facilement par les moteurs de recherche.

De nos jours, les doorway pages et les redirections par cloaking sont de plus en plus rares tant elles sont détectées presque automatiquement par les moteurs de recherche. Il est fortement recommandé d'oublier que cela puisse exister pour éviter de tomber dans le piège. Dans la majeure partie des cas, les pages satellites entraînent des déclassements plus ou moins importants pour les pages concernées.

Il n'existe pas de techniques propres pour rediriger vers des pages satellites mais en réalité, nous pourrions très bien utiliser une méthode avec des expressions régulières (`regex`) en PHP côté serveur, puis utiliser d'une redirection permanente tolérée par Google pour maintenir un semblant de technique de ce type. Bien entendu, cela est fortement déconseillé et ne garantit aucunement de résultats fiables, mais l'idée se tient. Voici comment nous pourrions procéder pour commencer :

```
function redirectionSatellite($urlPage) {
    // Expression régulière pour détecter les " agents " (robots)
    $regex = '#([bB]ot|[sS]pider|[yY]ahoo|[fF]eed|[gG]oogle|[sS]lurp)|[cC]rawl|[bB]ing)#iU';
    // Si ce n'est pas un robot, redirection vers la page normale
    if(!preg_match($regex, $_SERVER["HTTP_USER_AGENT"])) {
    header("Status: 301 Moved Permanently", false, 301);
    header("Location:".$urlPage);
    exit();
    }
}
```

Il suffit de lancer dès le début de la page optimisée SEO la fonction `redirectionSatellite($urlVisiteur)` avec une URL menant vers une page web pour les visiteurs. Notons qu'il ne s'agit que d'un exemple et que l'expression régulière est simple ici pour la démonstration. Cette technique reste chassée par les moteurs de recherche (Google en tête), il est donc intéressant de la connaître mais aussi de s'en méfier.

Contenus dupliqués et DUST

Les contenus dupliqués

Le contenu dupliqué *(duplicate content)* est l'un des fléaux qui pénalise le positionnement d'un grand nombre de sites web. Le fait de recopier en totalité ou presque des contenus venant d'autres pages peut entraîner des sanctions plus ou moins virulentes de la part des moteurs de recherche.

Google a le plagiat en ligne de mire car il n'apporte aucune valeur sur le Web (l'aspect juridique ne semble pas intéresser le moteur) et embête quelque peu les robots qui voient d'un mauvais œil le fait de distinguer les valeurs de pages ayant les mêmes contenus (source : http://goo.gl/5SRjxv). Il est très simple de se retrouver dans des situations de contenus dupliqués, il suffit par exemple de partager des extraits de textes sur des sites de curation ou des blogs pour que le « plagiat » soit détecté et chassé par les moteurs.

Qu'est-ce que la curation ?

Les sites web de curation sont des services en ligne dont l'objectif est d'amasser des multitudes d'informations issues de sites externes. Par exemple, des outils comme Digg ou StumbleUpon sont des sites qui agrègent des pages web et articles extérieurs pour les proposer à leur communauté. Nous pouvons parler de curation ou d'agrégation, les deux termes étant quasiment synonymes ici...

Le comble pour les contenus dupliqués est qu'une grande partie d'entre eux sont mis en place à l'insu des webmasters ou tout simplement par erreur technique, notamment dans les cas de DUST sur lesquels nous reviendrons par la suite. Certes, il arrive que des internautes n'aient pas froid aux yeux et pompent les contenus d'autres sites pour profiter de leur qualité, par exemple, mais dans la majorité des cas recensés, nous remarquons qu'il s'agit souvent d'erreurs humaines.

Les pénalités peuvent aller d'une désindexation des pages en passant par un court déclassement ou encore des sanctions plus lourdes si le nombre de contenus dupliqués est trop important. Mais ce qu'il faut retenir, c'est que les sites sources ne sont pas forcément protégés ou à l'abri des pénalités. En effet, les robots repèrent des contenus dupliqués mais ont souvent du mal à déterminer quels sont les originaux et donc les sanctions sont souvent appliquées aux deux pages.

Dans d'autres cas, c'est encore pire puisqu'il arrive que la pénalité ne soit appliquée qu'aux pages originales, sans que les pages de contenu dupliqué soient inquiétées. Ces cas se présentent en général lorsque le site frauduleux bénéficie d'un PageRank plus important ou tout du moins d'une valeur mieux estimée que le site source. Les robots ont tendance à protéger le site qui semblent être le plus populaire et honnête selon eux, mais dans bien des cas, il s'agit d'une erreur.

Si vous vous trouvez avec certitude dans ce cas, plusieurs solutions s'offrent à vous :

- contacter le webmaster frauduleux et lui demander gentiment de retirer les contenus avant une éventuelle plainte ;
- modifier vos propres contenus pour être certain de ne pas être pénalisé, mais ceci est frustrant lorsque beaucoup de pages sont touchées et que les contenus sont de qualité ;
- utiliser l'outil Spam Report pour la délation auprès de Google en relatant précisément les faits (source : http://goo.gl/IKgdeS) ;
- faire une demande de réexamen si la pénalité est tombée, en expliquant (et en prouvant) que les contenus originaux sont bien les vôtres.

La figure 3-15 montre un cas de contenu dupliqué flagrant détecté avec l'outil Plagiarism Checker (source : http://goo.gl/JulPx4) et confirmé par Copyscape (source : http://www.copyscape.com) concernant l'agence Internet-Formation dont une partie des contenus ont été repris mot pour mot par la pseudo-agence

ivoirienne OrishaCom. L'agence française n'a pas été sanctionnée mais a vécu quelques effets de bord avant de dénoncer le site copieur auprès des services de Google.

Les cas de plagiats arrivent, de nombreux témoignages sont lisibles sur le Web et montrent que les tricheurs n'ont aucun scrupule à recopier mot pour mot les contenus d'autres sites. Wikipédia est un peu dans cette logique puisque les auteurs n'ont pas réellement le droit de retirer les contenus qu'ils ont publiés sur le site, même s'ils apportent la preuve de leur existence plus ancienne et qu'ils en restent les uniques propriétaires (ce cas a été vécu par l'un des auteurs du livre et n'est pas un cas isolé, prenez garde…). Il convient donc de se méfier des contenus dupliqués car même des sites valorisés et importants peuvent se trouver dans ce cas…

Figure 3-18

Exemple de contenus dupliqués entre deux sites

Le problème du DUST et quelques solutions pour s'en sortir

Le DUST *(Different URLs with Similar Text)* est un problème récurrent provoqué et subi par les rédacteurs et animateurs de sites. Il correspond à des pages doublonnées qui reprennent tout ou en partie le contenu d'autres pages existantes. Nous pouvons parler dans ce cas d'URL dupliquées en quelque sorte… Il existe un nombre de cas incommensurable tant il est simple de dupliquer des contenus involontairement. Il est fortement recommandé d'être vigilant face à ce problème.

L'exemple le plus courant est souvent causé par les CMS tels que WordPress ou Joomla car les menus administrés en interne ne sont pas infaillibles. Si nous prenons le cas d'une page d'accueil, son URL peut être l'une des suivantes :

- http://www.monsite.com ;
- http://monsite.com ;
- http://www.monsite.com/index.php ;
- http://monsite.com/index.php...

Toutes ces URL peuvent fonctionner pour la même page d'accueil puisque les DNS du serveur redirigent souvent les domaines et leur sous-domaine en www. La page `index.php` (ou `index.html`, `home.php`...) est aussi la page d'accueil, elle ajoute donc encore d'autres doublons que les robots d'indexation distingueront naturellement. De plus, un dernier cas à vérifier est celui de la présence ou non du slash (/) après le nom de domaine qui peut parfois s'avérer fâcheux. Si vous décidez de placer un slash à la fin du nom de domaine (et des URL), il faut alors rajouter des directives dans le fichier `.htaccess` principal du site pour contrecarrer le problème et forcer la présence du slash :

```
# RewriteEngine On
RewriteCond %{REQUEST_URI} /+[^\.]+$
RewriteRule ^(.+[^/])$ %{REQUEST_URI}/ [R=301,L]
```

Le cas de la page d'accueil est relativement simple à résoudre. Nous pouvons modifier le menu géré par les sites web et remplacer le premier item par un lien en dur pointant vers le nom de domaine choisi. Sous WordPress, il suffit de sélectionner le menu *Apparence>Menus*.

Figure 3-19

Création d'un lien en dur sur WordPress pointant vers la page d'accueil pour éviter le DUST

L'autre technique consiste à utiliser un fichier `.htaccess` en effectuant une redirection permanente entre le nom de domaine et la page d'accueil. Cette technique peut également s'appliquer si vous avez un problème de DUST avec le nom de domaine avec et sans les www.

```
# Redirection permanente qui propage le jus de liens et évite le DUST
# Structure : RedirectPermanent  /ancienne-page.ext http://nouveau-site.ext
RedirectPermanent /index.php http://www.monsite.com
```

La réécriture d'URL permet parfois de régler certains cas de DUST mais la technique n'est pas toujours aisée à mettre en place, contrairement aux deux techniques précédentes qui s'appliquent parfaitement et simplement pour la page d'accueil notamment.

Attention au slash final

Il arrive également que les URL soient dupliquées uniquement à cause du dernier slash présent dans l'URL, comme dans http://www.monsite.com/ plutôt que http://www.monsite.com. Il convient alors de faire un choix entre les deux et de toujours conserver la même écriture du nom de domaine avec une règle comme celle écrite dans la page précédente.

Dans les faits, il existe de nombreux cas de DUST et une grande partie d'entre eux sont liés à un manque de précaution. Voici une courte liste d'exemples de DUST :

- pages d'accueil dupliquées ;
- cas du slash final dans les adresses mais aussi des adresses web avec ou sans extensions finales telles que .php, .html, .asp, etc., qu'il est possible de corriger grâce à un fichier `.htaccess` ;
- articles ou pages doublonnées ou multipliées à cause de modifications malencontreuses dans les backoffice (au niveau des URL ou des permaliens, par exemple) ;
- galeries multimédias (photos, vidéos, fichiers PDF...) ou catalogues avec pagination et répétition quasi complète des contenus internes ;
- URL contenant des paramètres comme des identifiants de sessions ou des ID, etc. ;
- noms de domaines et sous-domaines provisoires créés pour des tests ou avant l'installation définitive des sites web, etc.

Comme vous pouvez le constater, certains de ces cas sont relativement courants et peuvent nous concerner suite à un manque d'attention ou par méconnaissance technique. Souvent, il s'agit de contenus et de pages doublonnés lors d'une modification de contenus.

Prenons un exemple. Nous créons un article sur WordPress avec un permalien (URL directe également appelée *slug* ou alias) généré automatiquement par le CMS à partir du titre donné. Nous rédigeons nos contenus et nous publions l'article. Une fois en ligne, les robots vont pouvoir indexer la page web correspondante, et cela encore plus rapidement si nous utilisons les flux RSS couplés au protocole PubSubHubbub (aussi appelé Push).

Intérêt du protocole PubSubHubbub

Le protocole PubSubHubbub permet d'envoyer à un serveur compatible un « appel » pour le prévenir que de nouveaux contenus ont été ajoutés. Dans ce cas, Googlebot vient très rapidement indexer les contenus correspondants, après quelques minutes, alors qu'il faut parfois attendre des heures voire des jours habituellement.

Imaginons maintenant que nous souhaitions modifier le contenu ainsi que le titre et le permalien associé. Dans ce cas, les robots indexent le doublon et nous tombons dans le phénomène classique du DUST et des contenus dupliqués...

Figure 3-20

Modification du permalien d'un article sur WordPress

Les interfaces pour webmasters permettent de bloquer les paramètres d'URL qui causent des duplications mais les autres problèmes ne peuvent pas être résolus ainsi. C'est pourquoi Google a inventé le principe des URL canoniques qui permettent d'indiquer aux robots quelle est l'adresse mère à valoriser et à indexer au détriment des doublons restants.

Le principe des URL canoniques est très simple, il suffit d'ajouter une balise `<link />` spécifique dans la section `<head>…</head>` des pages concernées. La balise prend deux attributs pour être fonctionnelle :

```
<link rel="canonical" href="http://www.monsite.com" />
```

Par exemple, si vous avez les deux pages suivantes, il faudra ajouter la balise indiquant l'adresse canonique afin que les robots sachent quelle page retenir dans l'index :

- URL mère : http://www.monsite.com/categorie/article ;
- URL dupliquée avec `<link rel="canonical" … />` pointant vers l'adresse mère de l'article si nécessaire : http://www.monsite.com/categorie/resolution-du-dust.

Les CMS proposent presque tous des extensions (ou plug-ins) liées au référencement qui permettent d'ajouter et de gérer plus simplement des URL canoniques en cas de problème. Pour les sites créés manuellement, il suffit d'ajouter la balise dans les pages désirées. S'il s'agit de sites dynamiques, nous pouvons ajouter un champ pour préciser l'URL canonique et les notifier dans les bases de données. En résumé, il s'agit d'ajouter une colonne de base de données pour enregistrer l'URL canonique fournie dans un champ de formulaire correspondant à chaque page de contenu.

Si l'URL canonique présente de nombreux avantages, elle ne supprime pas réellement les URL doublonnées et il convient souvent de l'accompagner d'un bon fichier `robots.txt`, par exemple, voire de judicieuses redirections afin que le jus des liens soit propagé entièrement. En effet, il serait dommage que des internautes mettent en place des liens vers http://www.monsite.com/index.php et que les PageRank et BrowseRank obtenus soient perdus en route pour le nom de domaine officiel. Une redirection permanente présente l'avantage de renvoyer la « note » à l'URL cible donc cette technique peut souvent s'avérer utile pour certaines pages de valeur.

Enfin, un autre cas courant, présent notamment sur WordPress, est celui des pages doublons créées à cause des catégories d'articles. En effet, nous pouvons facilement nous retrouver avec des pages dupliquées si ces dernières sont reliées à plusieurs catégories. L'URL prend alors la forme http://www.site.com/nom-categorie/nom-page pour chaque page. Donc si cette dernière est placée dans deux catégories, nous nous retrouvons avec le contenu en double et deux URL différentes. Pour éviter ce problème, l'idéal est de procéder à une réécriture simple qui supprime la partie nom-categorie dans l'adresse, avec un code tel que celui-ci :

```
# RewriteEngine On
RewriteRule ^nom-categorie/(.*)$ http://www.site.com/$1 [R=301,L]
# ou RedirectMatch 301 ^/nom-categorie/(.*)$ http://www.site.com/$1
```

Attention aux redirections malencontreuses

Attention toutefois, cette méthode ne fonctionne pas toujours correctement selon votre structure de site, de base de données ou votre serveur.

Les paids links

L'obtention massive de liens entrants, ou *backlinks*, a toujours été en verve sur Google mais aussi sur d'autres moteurs de recherche plus récents tels que Bing ou Yandex. Le rôle des *rank* étant important, nombre de référenceurs souhaitent multiplier les liens vers leurs sites pour gagner en popularité et améliorer par ce biais leur positionnement.

Sur le principe, cela peut se comprendre car nous voulons tous être numéro un, et s'il faut obtenir des liens pour cela, alors la chasse est lancée… Dans les faits, tout est différent puisqu'il convient de ne pas confondre les liens de piètre qualité et ceux dont la valeur est quasi inestimable. Depuis la mise en place de Google Penguin, nous sommes certains que les liens entrants sont quantifiés par Google (c'était déjà le cas avant) mais surtout qualifiés ! En effet, la seule popularité ne se suffit plus, il faut également de la qualité et de la confiance, souvent rattachées au feu TrustRank.

Dans cette configuration, la donne a changé et il n'est plus possible d'obtenir des liens en masse de bonne qualité aussi facilement. En outre, Google a modifié le comportement de l'attribut `nofollow` en HTML afin de freiner la technique du PageRank Sculpting. Cette accumulation de mises à jour a poussé les spécialistes à chercher de plus en plus de liens par des biais différents…

La première solution, qui se pratique toujours, est de réaliser de faux communiqués de presse sous forme de blogs, par exemple, afin de rédiger des articles avec des contenus optimisés et quelques liens vers les sites associés. Ainsi, les moteurs peuvent penser qu'il s'agit de vrais textes et que tout cela est « naturel », mais dans la pratique, Google Panda repère assez facilement les faux blogs ou faux sites d'informations et les pénalisent lourdement ainsi que les petits malins qui en ont profité…

La deuxième méthode reste la recherche de liens par le biais d'annuaires, mais la qualité moyenne d'une grande majorité d'entre eux a fait déserter les référenceurs qui préfèrent se concentrer sur les meilleurs réseaux de sites.

Enfin, l'ultime méthode non naturelle est d'acheter des liens en faisant du paid linking. Elle permet de ne pas avoir à chercher longtemps et d'obtenir des liens en dur très facilement, ce qui n'est pas toujours détectable par les robots d'indexation. Cependant, la guerre a été lancée récemment par Google et de nombreux réseaux de liens payants ont été foudroyés.

Nous pouvons parler à nouveau de Buzzea, réseau français ayant succombé de ses blessures après avoir été sanctionné par Google. Matt Cutts a annoncé la sanction sur Twitter le 29 janvier 2014 mais

d'autres réseaux similaires tels que Backlink.com, Ghost Rank 2.0, AngloRank sont aussi ciblés par le moteur… Le chef de l'équipe Webspam en a profité pour mettre en garde les réseaux des autres pays, dont l'Allemagne, à propos des liens payants contre lesquels Google lutte. D'autres sanctions vont donc tomber dans les mois à venir.

Figure 3-21

Matt Cutts annonce la sanction envers le réseau de paid links français Buzzea.

Sur le principe, il est relativement logique que des liens obtenus contre de l'argent soient pénalisés car cela va à l'encontre des *guidelines* des moteurs. Aussi, les référenceurs honnêtes et respectueux se retrouvent lésés par ceux sans scrupule qui ne voient que par le PageRank. Il est certes plus facile d'acheter des lots de backlinks plutôt que de faire l'effort de construire avec intelligence son netlinking mais dans les faits, cette seconde solution est la seule qui permet de contrôler son profil de liens.

Le plus important n'est pas que les réseaux soient sanctionnés, mais surtout que les sites rattachés à ces réseaux prennent également une vague de pénalités. Google s'attache à amender tout le monde afin que cela ne se reproduise plus. En réalité, ce n'est pas la première fois que Google mène une lutte sans merci contre les techniques de spam à grand coup d'annonces fortes. Il faut donc être méfiant lorsque nous sommes dans la tourmente…

Rich snippets abusifs

Depuis l'arrivée des extraits de code enrichis *(rich snippets)*, les webmasters peuvent ajouter des attributs HTML qui permettent de qualifier certains types de contenus.

L'avantage est de donner encore plus de sens sémantique au code source pour aider les robots à mieux comprendre les structures de pages mais aussi pour améliorer l'accessibilité générale des pages, même si tous les systèmes ne sont pas en corrélation directe avec la norme WAI-ARIA élaborée à cet effet.

Avec de bons extraits de code enrichis, nous pouvons aisément ajouter des informations au sein des résultats de recherche comme le prix d'un produit ou la note d'un article, par exemple. Cela permet d'être encore plus visible dans les SERP mais aussi d'agrémenter le résultat d'informations complémentaires qui peuvent inciter les internautes à cliquer sur les liens enrichis plutôt que les autres.

Partant de ce constat, les premières fraudes aux extraits de code enrichis ont commencé et une nouvelle fois, Google a dû sévir pour remettre les choses dans l'ordre. En effet, un internaute a précisé sur un

forum avoir reçu un avertissement pour indiquer une pénalité causée par un usage abusif des rich snippets (source : http://goo.gl/T9cxpt).

Le message est très clair et indique une violation des guidelines : « Markup on some pages on this site appears to use techniques such as marking up content that is invisible to users, marking up irrelevant or misleading content, and/or other manipulative behavior that violates Google's Rich Snippet Quality guidelines. »

En d'autres termes, Google précise désormais aux webmasters qu'ils risquent une pénalité lorsqu'ils franchissent la barre de l'acceptable en matière d'utilisation des extraits de code enrichis.

Negative SEO

La technique du negative SEO (source : http://goo.gl/UedTFp) est la résultante logique de toutes les pénalités appliquées par les moteurs de recherche depuis des années puisqu'il s'agit de faire tomber des sites concurrents pour détruire le marché. Le principe est simple, il suffit de bien connaître les pénalités existantes et de tout faire pour les appliquer sur les sites concurrents afin de les faire chuter dans les SERP.

Le negative SEO (NSEO) a un effet destructeur souvent radical et il est difficile à déceler par les victimes mais aussi par les moteurs de recherche. C'est pourquoi les référenceurs les plus malintentionnés en abusent en toute impunité.

Nous venons de voir une liste de pénalités, sans oublier certaines autres comme la lutte contre les EMD, les contenus à faible valeur, les fermes de liens, etc. Par conséquent, il suffit à des spécialistes d'utiliser certaines techniques simples à mettre en place pour détruire la notoriété et la valeur d'un site auprès de Google, Bing, Yandex... Attention tout de même, ce livre n'est pas là pour vanter les mérites du negative SEO tant les agresseurs sont comparables à des casseurs, mais il est important de comprendre le principe pour mieux se protéger si possible.

Prenons un exemple simple : un référenceur vise une cible concurrente pour le faire chuter dans les SERP. Il va étudier les méthodes existantes et retenir celles qui sont le plus simples et les plus efficaces pour faire rugir les moteurs de recherche, Google en tête bien entendu. Un bon spécialiste ne mettra pas longtemps à trouver les techniques parfaites, en voici une courte liste, relativement faciles à mettre en place :

- créer des fermes de liens pointant vers des sites de la cible ;
- obtenir des liens de mauvaise qualité (paid linking ou non) par de mauvais annuaires, de mauvais blogs, etc. ;
- propager des contenus dupliqués sur divers sites ;
- réaliser de faux communiqués de presse avec des liens pointant vers la cible ;
- utiliser des failles web (injections, XSS...) pour ajouter ou modifier des scripts du site afin de faire du cloaking, des redirections malencontreuses, etc.

Une fois tout cela effectué, l'agresseur peut être encore plus vicieux et se servir du formulaire de délation Spam Report avec un compte fictif pour dénoncer le site cible qui sort des guidelines des moteurs et qui abusent des optimisations.

Figure 3-22

Formulaire de délation antispam de Google

Il ne s'agit que d'un exemple, mais si cela se fait sur plusieurs semaines naturellement et que la délation arrive à point nommé, Google aura bien du mal à savoir si le site concerné a subi une attaque de negative SEO ou s'il fraude réellement… Quant à la victime, elle sombre dans les SERP voire se retrouve en liste noire sans rien avoir demandé à personne – il faut alors qu'elle parte dans une succession de demandes auprès des moteurs pour être réintégrée et prouver son innocence…

Dans les faits, les cas de negative SEO ne semblent pas faire légion car ils sont assez longs à mettre en place. La majorité des spécialistes, aussi malintentionnés soient-ils, préfèrent tout de même travailler leur référencement plutôt que de passer du temps à détruire chaque concurrent. Certains cas ont été décelés ces derniers mois et d'autres vont suivre, mais il vaut mieux éviter de céder à la panique car toutes nos baisses de classement sont rarement liées à ce type d'attaque, rassurons-nous…

Lutter contre le negative SEO est relativement complexe car les attaques peuvent être multiples et surtout être de tous types. En effet, nous ne pouvons pas contrer des fermes de liens comme des contenus dupliqués, il faut donc trouver ce qui entraîne les chutes liées à des attaques de NSEO avant de les contrer, et c'est souvent très difficile à détecter.

Voici quelques exemples de ce qu'il est possible de faire pour limiter la casse voire de lutter contre certains types d'attaques, mais cela n'est pas efficace à 100 % :

- créer des liens vers les réseaux sociaux (et notamment Google+) à l'aide de badge, d'API... au sein des articles ;
- créer sa page Google+ Local (si nécessaire) et obtenir rapidement le contrôle des données pour éviter que d'autres personnes s'approprient le lieu ou le nom ;
- sécuriser les pages web au maximum contre les injections SQL, la modification des noms de fichiers ou encore les failles XSS ;
- créer un bon fichier `.htaccess` pour contrecarrer d'éventuels détournements et redirections frauduleuses ;

- essayer d'obtenir un PageRank plus que raisonnable pour limiter les problèmes liés aux contenus dupliqués en devenant le domaine de référence ;
- utiliser des outils pour vérifier le plagiat et le contenu dupliqué tels que Copyscape ou Plagium ;
- suivre toutes les mentions du nom, des produits, de l'activité avec des outils d'alertes comme Google Alertes, Alerti, Mention, Giga Alerts, Infoxicate, etc. ;
- signaler tous les contenus dupliqués et les faux communiqués de presse que l'on trouve sur son compte ;
- utiliser des outils de suivi de backlinks pour remarquer s'il y a des trop-pleins de liens entrants et si tel est le cas, désavouer tous les liens (source : http://goo.gl/Ej4p2p).

Si vous procédez à un suivi régulier de votre site web et que vous utilisez le désaveu des faux liens et les demandes de suppression ou la délation pour les contenus dupliqués ou autres contenus frauduleux, vous pouvez déjà prouver que vous respectez les règles des moteurs mais vous pouvez également vous prémunir contre les attaques de NSEO. Ainsi, les éventuels problèmes que vous n'auriez pas détectés seront moins efficaces contre votre site et les attaques deviendront obsolètes.

4

Le suivi du référencement

Nous savons à présent axer notre référencement autour d'une bonne stratégie de visibilité (optimisation technique, textuelle, in page et off page…), mais le travail n'est pas pour autant terminé.

Il faut désormais suivre les résultats des actions menées afin de pouvoir les ajuster en temps réel si nécessaire, ou alors profiter pleinement de leur réussite si les objectifs sont atteints. Dans tous les cas, l'analyse et le suivi permettent d'évaluer la qualité des efforts fournis et de mesurer l'impact de l'indexation et du positionnement pour chaque page.

Le suivi du référencement comporte deux grandes étapes : le suivi de l'indexation (s'assurer que le site web a bien été pris en compte par les moteurs de recherche) et le suivi du positionnement (s'assurer que le site est visible dans les pages de résultats et qu'il obtient les meilleures positions sur les requêtes travaillées).

Suivre l'indexation

Pour réaliser une bonne indexation, nous avons dû soigner quelques critères plus ou moins importants tels que les fichiers `robots.txt` et `sitemap.xml`, le code HTML (qui se doit d'être propre et optimisé), les balises meta `robots` ou encore la régulation des technologies qui freinent le référencement naturel (Flash, JavaScript, Ajax…).

Si tous les voyants sont au vert selon vous, vous avez donc tout mis en œuvre pour que votre site soit prêt à être indexé. Il ne reste qu'à vérifier la qualité de l'indexation et à suivre son évolution…

Voir le site avec l'œil du spider

Avant de vérifier et d'évaluer la pertinence de l'indexation, il peut être important de savoir ce que les spiders perçoivent lorsqu'ils crawlent votre site. Ainsi, nous pouvons rapidement déterminer si des facteurs sont encore bloquants, si le code est de piètre qualité ou si nous risquons de ne pas obtenir les résultats escomptés.

Le module Web Developer

Première chose, il est conseillé de télécharger le module gratuit Web Developer sur Firefox ou Chrome. Au-delà de ses aspects pratiques et techniques pour les développeurs web, il s'avère vraiment très utile pour les référenceurs.

Il fournit des informations sur les éléments en Flash, JavaScript ou sur les Frames, ainsi que des données concernant les images (poids, titre, texte alternatif…), la structure du contenu (balises `<h1>` à `<h6>`, par exemple) ou encore les liens externes...

Il permet également d'observer son site différemment, avec ou sans style CSS. Sur ce principe, c'est un peu comme cela que les moteurs de recherche tels que Google, Bing et Yahoo! perçoivent le site, bien qu'ils s'intéressent en réalité au code source.

Figure 4-1

Un site sans style tel qu'il est vu par les robots.

Nous pouvons voir dans quel ordre s'affichent les différents blocs et donc observer le sens de lecture des robots. Certes, cela n'est pas nouveau mais nous constatons donc que les contenus sont bien lus de haut en bas et de gauche à droite.

Historiquement, certains moteurs de recherche n'indexaient qu'une partie des contenus, il fallait donc placer le plus haut possible les informations importantes pour être assuré d'être indexé mais aussi pour être mieux positionné. De nos jours, les moteurs indexent la totalité des contenus et un doute persiste toujours sur l'importance des premiers contenus affichés.

Matt Cutts avait indiqué en mai 2010 lors d'une conférence à Paris que les contenus situés en haut de page n'avaient pas plus de valeur que ceux situés en pied de page, sauf si les textes étaient répétés sur toutes les pages. En d'autres termes, si nous prenons au pied de la lettre ses affirmations, le positionnement dans la

page ne compte pas mais les contenus répétés sont dévalués. Tous les tests effectués jusqu'à présent ont démontré le contraire, il est donc difficile de se faire un avis tranché tant la communication de Google est parfois maîtrisée pour semer le doute.

Retenons l'essentiel, l'extension Web Developer permet d'afficher les contenus sans style et d'observer rapidement comment est fondé le site web et ce que les robots peuvent voir. Mais cela ne va pas encore assez en profondeur, il faut aussi s'intéresser au code source…

Les simulateurs de robots

Les *spiders simulators* (ou simulateurs de robots) permettent de reproduire le comportement des robots d'indexation. Ces outils scrutent les sites web comme s'ils étaient des crawlers de moteurs de recherche afin de vous fournir un rendu visuel. Certes, le résultat n'est pas toujours facile à déchiffrer pour les plus débutants, voire indigeste, mais il faut surtout l'analyser pour voir si nous pouvons trouver facilement les contenus textuels, les menus (…) et comprendre la logique structurelle des pages…

Voici une liste de simulateurs en ligne de bonne facture (il arrive que certains affichent le résultat dans un mauvais encodage mais ne vous inquiétez pas, les robots lisent mieux les contenus) :

- http://tools.seochat.com/tools/search-spider-simulator/ ;
- http://www.webconfs.com/search-engine-spider-simulator.php ;
- http://totheweb.com/learning_center/tools-search-engine-simulator/ ;
- http://www.webmaster-toolkit.com/search-engine-simulator.shtml ;
- http://smallseotools.com/spider-simulator/ ;
- http://www.iwebtool.com/spider_view ;
- http://www.yatooweb.com/referencement/simuler-robot.php.

Figure 4-2

Extraction des contenus vus par les robots


SEO Tools : Search Engine Spider Simulator

Spidered Text :

Le plus précis d'entre tous est sûrement celui proposé par le site totheweb.com. Il affiche les contenus sobrement et sépare les textes inscrits dans les balises `<title>` et les métadonnées. Nous pouvons donc avoir un coup d'œil rapide du résultat et analyser les textes en profondeur.

Figure 4-3

Analyse du site de TF1 avec le simulateur du site totheweb.com.

Le principal défaut des simulateurs de robots est qu'ils ont trop tendance à se concentrer uniquement sur les contenus, c'est-à-dire qu'ils se comportent davantage comme des extracteurs de textes plutôt que comme des robots réels.

Toutefois, ils donnent une approche différente et permettent de se rendre compte rapidement de la quantité de textes insérés dans les pages et surtout des informations qui arrivent le plus tôt dans les pages. Aussi, nous pouvons améliorer notre structure de site voire notre ergonomie en nous appuyant sur ces résultats afin de proposer les contenus les plus importants aux visiteurs le plus rapidement possible, ce qui constitue un bon point pour le référencement.

Suivre les robots avec les Webmaster Tools

Les outils présentés précédemment permettent une première approche mais aucun ne remplace réellement l'œil d'un vrai robot. En revanche, pourquoi aller chercher loin quand certains moteurs tels que Google et Bing nous proposent des outils simples à utiliser pour savoir comment leur robot scrute les sites web ?

En effet, Google, Bing et même le moteur russe Yandex proposent des Webmaster Tools, c'est-à-dire des interfaces complètes dans lesquelles de nombreux outils sont disponibles, par exemple pour analyser les pages web comme les robots.

Pour accéder à cette fonctionnalité, connectez-vous à votre compte Google Search Console, puis cliquez sur *Exploration>Explorer comme Google*. Saisissez ensuite une URL, le résultat affiché correspondra à la lecture du site tel que le voit Googlebot. Si vous voulez savoir ce que le robot lit dans les pages, cliquez sur l'onglet *Récupération* qui affiche les en-têtes HTTP ainsi que le code source lu.

Notez également que cet outil est pratique pour favoriser l'indexation car une option permet d'envoyer la page à l'indexation ainsi que les pages reliées.

Depuis début novembre 2015 (source : http://goo.gl/tay7uU), Google a ajouté un bouton *Explorer et afficher* qui fournit plus de précisions sur le crawl de Googlebot. L'outil affiche la page telle que le robot la perçoit et comme un visiteur lambda la voit, tout en indiquant en dessous les ressources bloquées et les raisons du blocage. Qui plus est, le code source du simulateur de robot possède un code couleur avec parfois des notations en rouge vif pour les ressources bloquantes, comme dans la figure 4-4.

Figure 4-4

Explorer comme Googlebot avec la Google Search Console

Dans Bing Webmaster Center, le procédé est quasi similaire. Cliquez sur *Diagnostics et outils>Analyser comme Bingbot*, puis saisissez l'URL à analyser et cliquez sur *Terminé* pour afficher le résultat. Les codes couleurs affichés dans l'interface permettent de lire encore plus facilement le code que dans l'outil de Google.

Figure 4-5

Visualiser un site web comme Bingbot

Mise en cache et paramètres cachés

Enfin, terminons notre tour des techniques simples qui nous permettent de nous mettre à la place d'un robot d'indexation en vérifiant la mise en cache des pages web au sein même des SERP.

Il arrive que des moteurs de recherche proposent d'afficher les pages en cache, c'est notamment le cas de Yahoo!, Bing et Google. Ainsi, nous pouvons obtenir une capture à une date précise mais aussi des informations intéressantes.

La mise en cache nous permet, par exemple, de capter la dernière date d'indexation (ou de modification) des pages web dans l'index des moteurs de recherche. Si nous analysons fréquemment les dates affichées, nous pouvons trouver un intervalle moyen de crawl de la part des robots, et donc leur fréquence de passage. Cet indicateur peut se révéler primordial quand nous voulons suivre notre indexation, nous reviendrons sur ce point par la suite.

Dans un second temps, il est important d'analyser en profondeur les URL des pages mises en cache car elles révèlent parfois des indications non négligeables dans le suivi.

Prenons le cas de plusieurs URL de cache sur Google, Bing, Yandex et même le moteur chinois Baidu pour le site www.abondance.com fondé par Olivier Andrieu. Toutes ces URL de cache sont longues et complexes à déchiffrer mais tellement intéressantes :

- Sur Google :

```
http://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQIDAA&url=http%3A%2F%2F
webcache.googleusercontent.com%2Fsearch%3Fq%3Dcache%3AlCck6ZHqrVgJ%3Awww.abondance.com%2F%2B%
26cd%3D1%26hl%3Dfr%26ct%3Dclnk%26gl%3Dfr&ei=KhSgU9j4BMGrOQWlzYCACg&usg=AFQjCNEh-x2uWWzO8_RYUa
ofJjw5haqM3w&sig2=UFMYSe1X1q5GZdxXXMv2zA&cad=rja
```

- Sur Bing :

```
http://cc.bingj.com/cache.aspx?q=r%c3%a9f%c3%a9rencement+inurl%3aabondance&d=4629774216856 58
2&mkt=fr-FR&setlang=fr-FR&w=tIiVqzohmfQ5YNMb2X6JjRJJZMsDeMBk
```

- Sur Yahoo! :

```
http://212.82.99.176/search/srpcache?ei=UTF-8&p=r%C3%A9f%C3%A9rencement+inurl%3Aabondance&fr=
yfp-t-401&u=http://cc.bingj.com/cache.aspx?q=r%C3%A9f%C3%A9rencement+inurl%3aabondance&d=4629
774216856582&mkt=fr-FR&setlang=fr-FR&w=tIiVqzohmfQ5YNMb2X6JjRJJZMsDeMBk&icp=1&.intl=fr&sig=Tb
YdrveEM5WDkOngSkRLSw--
```

- Sur Yandex :

```
http://hghltd.yandex.net/yandbtm?fmode=inject&url=http%3A%2F%2Fblog.abondance.com%2F&tld=ru&l
ang=en&la=&text=r%C3%A9f%C3%A9rencement%20inurl%3Aabondance&l10n=ru&src=F&mime=html&sign=7763
47ffbf0a5d634ae0199d248f6c3a&keyno=0
```

- Sur Baidu :

```
http://cache.baiducontent.com/c?m=9f65cb4a8c8507ed4fece763105392230e54f7306c8a8c432c88c21f846
```

Comment obtenir les URL détaillées des moteurs ?

Les URL doivent s'obtenir en copiant l'adresse du lien lorsque l'on pointe sur le lien de mise en cache, et non pas en recopiant l'adresse affichée une fois dans la page en cache. En effet, certaines informations disparaissent dans ce cas…

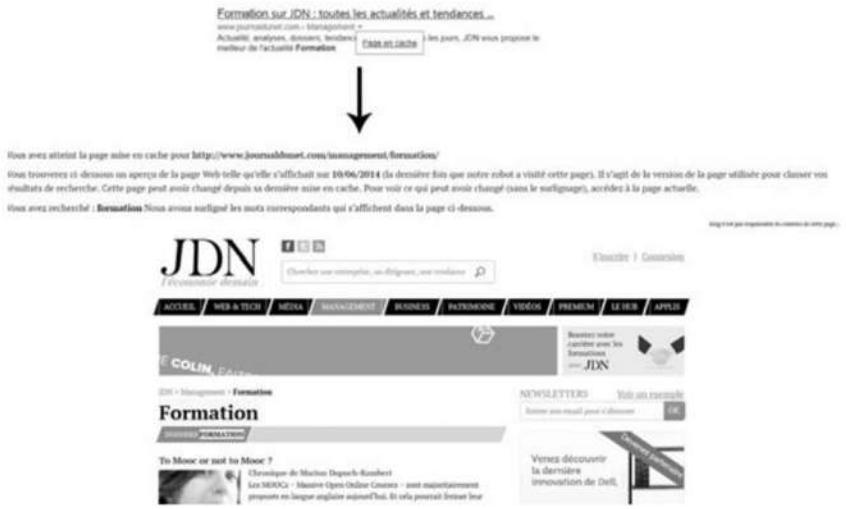
Si nous analysons ces adresses de cache, nous remarquons tout d'abord qu'aucune ne fait la même taille et qu'une multitude de paramètres sont transmis via les URL. Tous ne sont pas simples à déchiffrer mais certains sont compréhensibles et intéressants.

- Les paramètres de récupération de la requête varient d'un moteur à l'autre. Nous retrouvons `q` pour Google (vide si nous sommes en HTTPS), Yahoo! et Bing, `query` pour Baidu et `text` pour Yandex.
- Certains moteurs permettent de trouver le positionnement de la page à la date du cache. Google affiche notamment le paramètre `cd` qui indique la position dans les SERP, tandis que Baidu utilise `pl` pour cette information. Les autres moteurs masquent ces indications qui sont très importantes pour le suivi du positionnement. Nous reviendrons en détail sur ce sujet dans la suite de ce chapitre (section « Utiliser PHP pour réaliser des rapports de positionnement »).
- La langue de recherche est indiquée dans divers paramètres tels que `lang`, `mkt`, `setlang` ou encore `l10n` mais tous ne l'affichent pas. Il faut dire que cette information n'est pas nécessairement majeure pour la mise en cache et l'extension du nom de domaine peut suffire pour comprendre la langue.
- Enfin, Google et Yandex indiquent la source de la recherche. Si le mot-clé `web` chez Google est assez explicite comme source, le paramètre `src` de Yandex précise un `F` qui semble correspondre à la recherche classique.

Nous venons donc de remarquer que la mise en cache pouvait s'avérer intéressante dans l'analyse des pages web d'un point de vue visuel mais aussi pour amorcer un suivi de positionnement selon les moteurs de recherche. Retenons cependant que son rôle est avant tout d'aider à analyser l'indexation des pages et la fréquence de passage des robots.

Figure 4-6

Mise en cache proposée par Bing pour le JDN



L'avantage de certains moteurs comme Google est de proposer également une « version texte » de la mise en cache afin de voir le site sans style CSS et d'avoir un instantané à une date précise de l'ordre structurel des contenus notamment.

Figure 4-7

La même page en cache en texte seul

Scruter les contenus avec un robot en PHP

Rappelons-nous que les moteurs de recherche lisent avant tout du code source et crawlent les fichiers d'un site dans leur totalité en extrayant toutes les informations intéressantes telles que les en-têtes HTTP, les contenus textuels, les extensions de fichiers ou encore les zones chaudes…

Aucune des techniques employées précédemment ne nous permet de récupérer toutes ces informations, il faudrait plusieurs outils pour cela. Nous allons créer un simulateur de robot « maison » en PHP pour scruter nos pages de manière entièrement personnalisée. L'avantage est de pouvoir modifier le code à notre guise selon nos besoins pour obtenir toutes les informations utiles.

Nous allons devoir procéder par étape pour réaliser un simulateur totalement indépendant qui s'adapte à chaque page pour montrer ce que les robots lisent dans nos codes sources :

1. créer une fonction qui va aller chercher le code source des URL à l'aide de la bibliothèque cURL en PHP ;
2. réaliser une page avec un formulaire HTML adapté qui lancera la fonction si le simulateur est lancé ;
3. ajouter des styles CSS pour personnaliser l'interface finale et obtenir un rendu intéressant.

Qu'est-ce que cURL ?

La bibliothèque cURL *(client URL Request Library)* permet de manipuler et de récupérer des informations issues de pages ou d'autres ressources présentes sur un réseau informatique. Les avantages sont nombreux car cURL nous donne la possibilité de scruter tout ce que nous souhaitons, il faut juste veiller à ne pas outrepasser le cadre légal…

Le module cURL doit être installé sur un serveur pour que la fonctionnalité soit disponible. Il convient donc de vérifier sa présence et son activation avec la fonction PHP `phpinfo();`.

Notre fonction de simulateur de robot va pouvoir réaliser plusieurs tâches avant d'afficher le résultat :

- parcourir l'URL testée ;
- récupérer les en-têtes HTTP avec cURL ;
- récupérer le contenu de la page avec cURL et afficher le code source proprement ;
- gérer les problèmes d'encodage à la volée pour éviter un affichage de mauvaise qualité ;
- ajouter une option pour numéroter les lignes du code source ;
- ajouter une colorisation personnalisée du code pour mieux se repérer et faciliter la lecture.

Une fois toutes ces étapes franchies, nous devrions obtenir un résultat comme celui représenté à la figure 4-8.

Figure 4-8

Exemple d'un site vu via un robot d'indexation

Tout d'abord, créons le fichier `spider-simulator.php`. Ce sera ce fichier qu'il faudra lancer pour afficher le simulateur et son formulaire. Nous pouvons l'activer sur la Toile ou tout simplement passer par un serveur local comme WampServer ou EasyPHP. cURL fonctionnera à merveille dans ce cas mais il faudra veiller à ce qu'il soit activé.

Activer cURL sur un serveur local

Sur WampServer, cURL n'est pas activé par défaut. Pour l'activer, sélectionnez l'extension `php_curl` située dans PHP>Extensions PHP.

Procédons par étape pour construire notre simulateur de robot… Pour commencer, nous créons la fonction du robot :

```
<?php
// Fonction du simulateur de robot
// 1. URL à tester
// 2. Activer ou non la colorisation du code (true/false)
```

```
// 3. Numéroter ou non les lignes de code (true/false)
function spiderSimulator($page = '', $colorisation = false, $numerotation = false) {
// Ajout du protocole s'il est manquant
if(!preg_match("#^https?://#iU", $page)) {
        $page = "http://".$page;
}

// Activation de cURL
$url = curl_init($page);

// Options de cURL (retour des données et des en-têtes)
curl_setopt($url, CURLOPT_HEADER, true);
curl_setopt($url, CURLOPT_RETURNTRANSFER, true);
curl_setopt($url, CURLOPT_SSL_VERIFYPEER, false);

// Récupération du contenu
$contenu = curl_exec($url);
$code = '';

// Fermeture de cURL
curl_close($url);

// Découpage du contenu en ligne
$lignes = explode("\n", $contenu);

// Afficher ligne par ligne
foreach($lignes as $num => $ligne) {
    // Affichage optionnel du numéro de ligne
    if($numerotation == true) {
        $code .= $num.". ";
    }

    // Affichage des balises HTML
    $ligne = htmlspecialchars($ligne);

    // Si la colorisation est active
    if($colorisation == true) {
        // Colorisation des attributs et valeurs d'attributs
        $regex = "#(.*)([a-zA-Z0-9:-]+)(=)(&amp;quot;|&amp;apos;|[\'])([^\']+)
                  (&amp;quot;|&amp;apos;|[\'])([/> ]|&amp;gt;)#iU";
        $replace = "$1<span style='color:#FB5758'>$2$3</span><span style='color:#FB5758'>$4
                    </span><span style='color:#999'>$5</span>
               <span style='color:#FB5758'>$6</span>$7";
        $ligne = preg_replace($regex, $replace, $ligne);

        // Colorisation des balises
        $regex = "#(&amp;lt;/?[a-zA-Z0-9!]+[ ])#iU";
        $replace = "<span style='color:#0089E2'>$1</span>";
        $ligne = preg_replace($regex, $replace, $ligne);
        $regex = "#(&amp;lt;/?[a-zA-Z0-9]+/?&amp;gt;)#iU";
```

```
        $replace = "<span style='color:#0089E2'>$1</span>";
        $ligne = preg_replace($regex, $replace, $ligne);
        $regex = "#(/?&gt;)#iU";
        $replace = "<span style='color:#0089E2'>$1</span>";
        $ligne = preg_replace($regex, $replace, $ligne);
    }
    $code .= $ligne;
    $code .= "<br/>\n";
}

// On force l'affichage en UTF-8
preg_match("#charset=['\"]?([a-zA-Z0-9-]+)['\"]?[^a-zA-Z0-9-]#iU", $code, $result);
if(!empty($result)) {
    $encodage = strtolower($result[1]);
    if($encodage != 'utf-8') {
        $code = mb_convert_encoding($code, "UTF-8", $encodage);
    }
}

// Affiche le code source complet
echo $code;
}
?>
```

Alternatives pour afficher le code source en PHP

Il existe une méthode encore plus simple pour afficher le code source d'une page. Il suffit d'utiliser la fonction `show_source()` ou son équivalent `highlight_file()` en PHP pour donner un rendu total. Toutefois, la colorisation relative à ces fonctions n'est pas toujours activée sur nos serveurs et elle ne rend pas les en-têtes HTTP, ce qui nous éloigne d'un vrai robot d'indexation.

Maintenant, créons la page HTML avec formulaire et style CSS :

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Spider Simulator</title>
<style type="text/css"/>
* {margin:0; padding:0; font-size:100%}
#formulaire {background:#eee; padding:1%}
#formulaire h1 {font-size:1.8em; color:#004C54; margin-bottom:1em}
.bloc {margin-bottom:.8em}
.bloc label {display:block; font-weight:bold; padding:.1em;}
.bloc input, .bloc select {display:block; float:left; margin-right:1em; padding:.1em;
border:1px solid #ccc;}
.bloc input {width:250px;}
.bloc select {width:72px; text-align:center;}
#resultat {background:#fafafa; padding:1em}
```

```
#resultat h2 {font-size:1.3em; color:#198A95; margin-bottom:.8em}
#bouton input {padding:.2em .5em; font-weight:bold; border:1px solid #ccc; background:#fff;
              color:#004C54}
#bouton input:hover {background:#004C54; color:#fff}
</style>
</head>

<body>
<div id="formulaire">
<h1>Simulateur de robot d'indexation</h1>
<form method="post">
    <div class="bloc">
    <input type="text" name="url" value="<?php if(isset($_POST['url']))
    { echo $_POST['url']; } ?>"/>
    <label for="url">URL à tester</label>
    </div>

    <div class="bloc">
    <select name="col">
        <option value="0" <?php if(isset($_POST['col']) && $_POST['col'] == 0)
        { echo 'selected="selected"'; } ?>>Non</option>
        <option value="1" <?php if(isset($_POST['col']) && $_POST['col'] == 1)
        { echo 'selected="selected"'; } ?>>Oui</option>
    </select>
    <label for="col">Colorisation du code</label>
    </div>

    <div class="bloc">
    <select name="num">
        <option value="0" <?php if(isset($_POST['num']) && $_POST['num'] == 0)
        { echo 'selected="selected"'; } ?>>Non</option>
        <option value="1" <?php if(isset($_POST['num']) && $_POST['num'] == 1)
        { echo 'selected="selected"'; } ?>>Oui</option>
    </select>
    <label for="num">Numérotation des lignes</label>
    </div>

    <p id="bouton"><input type="submit" name="submit" value="Tester"/></p>
</form>
</div>

<div id="resultat">
<?php if(isset($_POST['submit']) && !empty($_POST['url'])) {      ?>
<?php // Traitement des données
    $url = htmlspecialchars($_POST['url']);
    $col = htmlspecialchars($_POST['col']);
    $num = htmlspecialchars($_POST['num']);
?>
<h2>Affichage pour l'adresse <?php echo $url; ?></h2>
<p><?php spiderSimulator($url, $col, $num); ?></p>
```

```
<?php ] ?>
</div>
</body>
</html>
```

Il suffit enfin de lancer le fichier PHP pour obtenir le formulaire. Ensuite, testons une URL et choisissons les options que nous préférons pour obtenir le résultat adéquat. Il est recommandé d'utiliser la colorisation du code source pour faciliter la lecture. Ainsi, nous pouvons rapidement constater que quelque chose ne pas va et si le code peut être bloquant…

Suivre les pages indexées

Les moteurs de recherche enregistrent 24h/24 des pages web grâce à des systèmes puissants et avancés de crawl. Nous devons absolument suivre les pages indexées afin de mesurer la qualité des sites web ainsi que leurs performances en termes d'indexation.

Nous allons voir plusieurs techniques pour obtenir des résultats intéressants mais nous remarquerons rapidement qu'aucune n'est idéale. Qui plus est, les moteurs n'indexent que les pages qu'ils estiment pertinentes, même lorsque nous mettons tout en œuvre pour enregistrer la totalité de nos pages. De ce fait, il sera difficile d'obtenir des résultats complets et réalistes du suivi de l'indexation.

La commande site:

La solution la plus simple pour savoir si une page a bien été indexée par les moteurs de recherche est de taper la commande `site:` dans le champ de recherche des moteurs. Cette commande ne fonctionne pas sur tous les moteurs du marché mais elle permet de rapidement scruter les pages enregistrées dans les index.

En d'autres termes, si un moteur de recherche ne fait ressortir aucune voire très peu de pages web en utilisant cette commande, vous pouvez estimer qu'il existe un problème d'indexation à résoudre. Ils peuvent être nombreux et souvent résolus en menant des actions précises :

- relancer un fichier `sitemap.xml` ;
- proposer des URL à l'indexation dans les moteurs ou les interfaces pour webmasters ;
- améliorer la structure du site et le Bot Herding, à savoir le parcours des robots.

Voici un exemple d'utilisation de la commande `site:` avec le blog www.miss-seo-girl.com. Google a indexé 739 pages, ce qui prouve que le moteur n'a rencontré aucun souci majeur pour indexer les pages. Cependant, n'oublions pas que la totalité des pages ne sera pas nécessairement indexée, ce n'est donc pas parce que toutes les pages ne sont pas retenues qu'il existe des problèmes majeurs d'indexation. Il suffit juste de veiller aux liens morts ou aux pages dupliquées, par exemple, pour contrer les cas problématiques.

Figure 4-9

Suivi de l'indexation dans Google

Il faut également noter que la commande `site:` prend en compte le préfixe qui précède le nom de domaine. En effet, un sous-domaine est différencié du domaine principal en `www`. Si vous tapez `site:site.com`, les moteurs affichent l'ensemble des pages indexées pour le nom de domaine et ses sous-domaines. Il faut donc taper `site:www.site.com` si vous ne souhaitez afficher que les URL du site principal. Sur le même principe, tapez une commande telle que `blog.site.com`, par exemple, si vous voulez voir seulement les pages indexées d'un sous-domaine.

La commande `site:` est pratique mais il arrive que le total des pages indexées soit inexact. En effet, il est fréquent que les moteurs suppriment des doublons ou masquent des pages qu'ils estiment moins pertinentes, par exemple, notamment pour des sites de grande envergure. Néanmoins, si le site est qualitatif et qu'aucun problème majeur de crawl n'est détecté, les résultats sont plutôt de qualité et donnent un aperçu intéressant du travail d'indexation effectué. Dans le pire des cas, vous pouvez utiliser cette fonction PHP pour obtenir le nombre total de pages indexées issu de l'API de Google :

```
<?php
$domaine = "www.domaine.ext";
function pagesIndexees($domaine = '') {
if(!empty($domaine)) {
    // URL de l'API Google
    $url = "http://ajax.googleapis.com/ajax/services/search/web?v=2.0&q=";
    $url.= "site:".$domaine;
```

```
    $url.= "&filter=0";

    // Lancement de cURL pour récupérer les données de l'API
    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
    $contenuJSON = curl_exec($curl);
    curl_close($curl);

    // Récupération du résultat
    $resultat = json_decode($contenuJSON, true);

    if($resultat['responseStatus'] == 200) {
        return $resultat['responseData']['cursor']['resultCount'];
    }
}
}
echo pagesIndexees($domaine);
?>
```

Cette technique est fastidieuse à utiliser pour les grands sites web mais présente l'avantage d'afficher les URL des pages indexées au sein des SERP. Nous pouvons donc les copier manuellement si le courage nous en prend afin de lister les pages retenues et donc déduire les pages manquantes. Par conséquent, nous pouvons réagir si nous notons l'absence de pages importantes en essayant par tous les moyens de les enregistrer dans les index des moteurs.

Webmaster Tools pour suivre l'indexation

Une autre solution pour avoir une idée du nombre des pages indexées par les moteurs de recherche est d'utiliser les interfaces pour webmasters (Google, Bing, Yandex…). Ces outils en ligne offrent la possibilité de suivre l'état de l'indexation fréquemment mis à jour par les moteurs. Ainsi, des courbes ou tableaux résument rapidement le nombre de pages indexées au fil du temps.

En revanche, il faut être très prudent avec les chiffres annoncés car ils sont parfois erronés pour plusieurs raisons :

- la mise à jour des données n'a pas encore été affichée dans les Webmaster Tools ;
- les pages doublons sont comptabilisées ou ignorées ;
- des erreurs de crawl n'ont pas permis de retenir toutes les pages à la dernière date de passage.

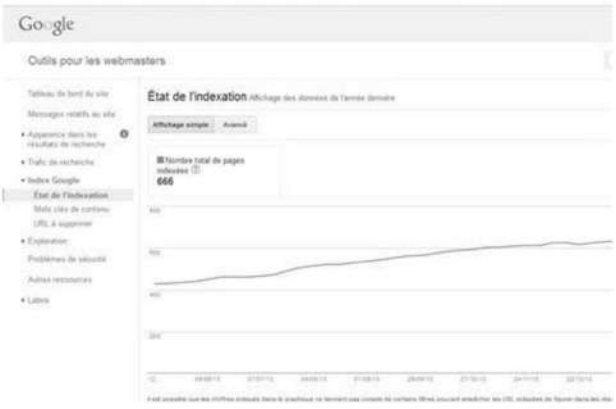
Globalement, les résultats affichés sont plus plausibles qu'avec la commande `site:` et ils sont souvent proches de la réalité sur l'état de l'indexation.

Si vous avez des doutes et que vous avez mis en place un fichier `sitemap.xml`, il peut être intéressant de comparer le nombre d'URL indexées à partir du `sitemap.xml` et celui affiché dans le suivi de l'indexation des interfaces pour webmasters. Il se peut que persistent quelques différences mais normalement, les chiffres doivent être relativement approchants.

Pour suivre la qualité de l'indexation dans Google Search Console, il suffit de cliquer sur *Index Google>État de l'indexation*. Google fournit des statistiques et une courbe évolutive sur l'indexation de vos pages (nombre de pages ajoutées ou supprimées de l'index).

Figure 4-10

Suivi de l'état de l'indexation dans la Google Search Console

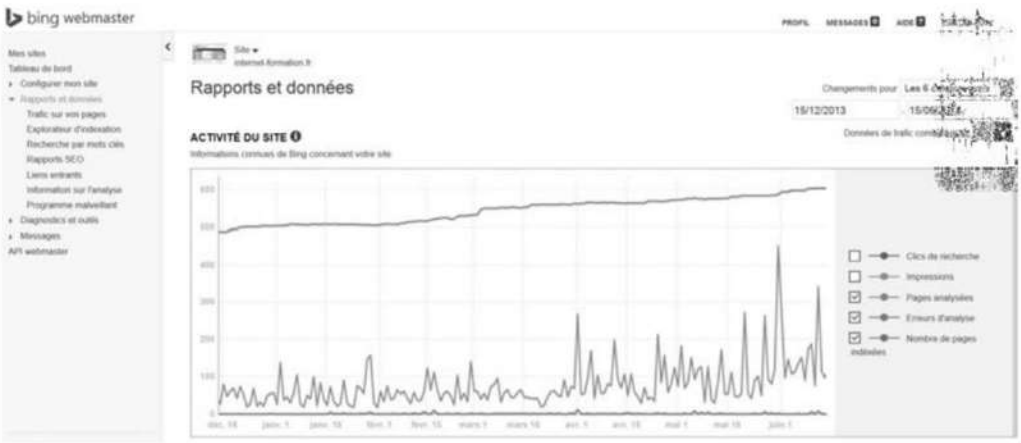


L'inconvénient majeur du service de Google est de ne pas lister les pages indexées, contrairement à la commande `site:`. Par conséquent, nous connaissons le nombre de pages retenues mais pas le nom des fichiers concernés, ce qui est pourtant primordial pour suivre l'état réel de l'indexation. Il est donc difficile de se savoir quelle page est retenue ou non, nous pouvons uniquement remarquer les éventuels problèmes d'indexation subis par le site.

D'autres moteurs comme Bing et Yandex proposent aussi ce type de service avec cette fois un listing des pages indexées. Sur Bing, il suffit d'aller dans la Toolbox pour webmasters et de cliquer sur Rapports et données. Nous pouvons alors scruter le nombre de pages indexées au fil du temps par Bingbot.

Figure 4-11

Suivi de l'état de l'indexation dans la Toolbox pour webmasters de Bing



Bing présente l'avantage de lister les pages retenues par Bingbot lorsque nous cliquons sur Explorateur d'indexation. Certes, tous les dossiers et toutes les adresses ne correspondent pas toujours à des URL réelles mais cela donne une idée rapide des pages indexées et de celles qui méritent encore des efforts de notre part pour améliorer l'enregistrement des pages.

Figure 4-12

Explorateur d'indexation dans la Toolbox de Bing

Dans les Yandex Webmaster Tools, il faut cliquer sur Indexing>Webpages available on Yandex search pour suivre la courbe du nombre de pages indexées. Nous pouvons également obtenir une liste approximative des pages indexées en cliquant sur Site structure et suivre les pages retenues par le robot russe.

Suivre les pages dans les Sitemaps XML

La technique la plus évidente pour suivre l'indexation des pages que nous voulons absolument insérer dans l'index des moteurs de recherche est d'utiliser les outils pour webmasters et le suivi des fichiers Sitemap XML.

En effet, lorsque nous ajoutons un fichier `sitemap.xml` dans les interfaces pour webmasters, ces outils affichent le nombre de pages présentes dans le fichier XML mais aussi celles comptabilisées dans l'index du moteur concerné. Il suffit alors de comparer pour déduire si l'indexation est qualitative ou non.

Cette technique ne donne qu'un chiffre total de pages indexées sur le nombre de pages recensées dans les fichiers Sitemap XML. En aucun cas nous ne savons quelles pages ont été retenues, mais cela permet toutefois d'observer rapidement l'état de l'indexation sur les moteurs qui disposent d'une interface pour webmasters.

Figure 4-13

Contrôle d'un fichier Sitemap XML sur Bing

Autres outils d'analyse

Il existe également d'autres méthodes pour suivre l'indexation d'un site. Par exemple, nous pouvons construire notre propre outil de crawl afin de scruter les moteurs de recherche mais cela est très technique et impose une maîtrise quasi totale d'un ou plusieurs langages de programmation. L'autre solution est d'utiliser des outils efficaces disponibles sur le marché.

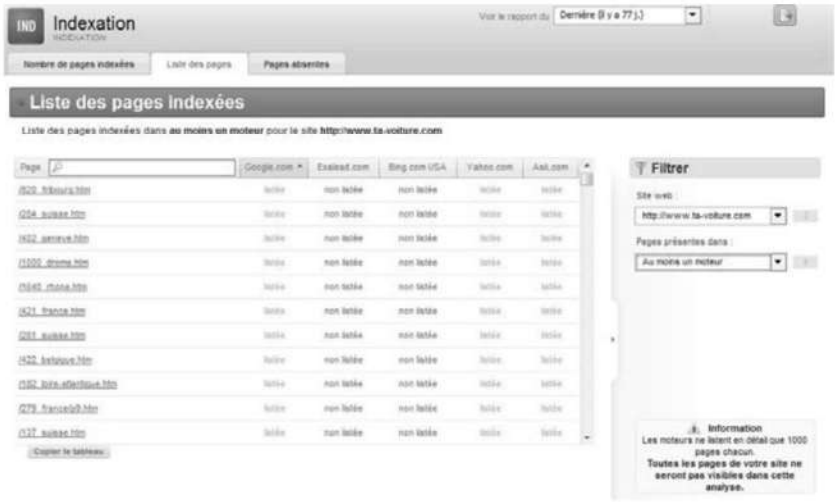
En effet, pourquoi réinventer la roue s'il existe déjà des outils pour réaliser cette tâche ingrate ? Nombreux sont les services en ligne ou logiciels pour suivre l'indexation mais les plus pertinents sont certainement les outils payants SeeUrank de Yooda (source : http://goo.gl/6gTBR), Track-Flow de Cibercité (source :

http://www.cybercite.fr/track-flow.html), le logiciel gratuit CrawlTrack (source : http://www.crawltrack.fr), l'outil d'indexation de SEO Administrator (source : http://www.seoadministrator.com/indexation-site.html) ou le module WordPress Crawl Rate Tracker de Yoast (ou sa seconde version plus récente fournie sur GitHub : https://github.com/chrisguitarguy/Crawl-Rate-Tracker-2).

Pour suivre l'indexation, SeeUrank semble avoir une longueur d'avance sur certains concurrents car il liste l'ensemble des pages indexées. Nous pouvons extraire les données et suivre l'évolution à chaque fois que nous lançons un test, ce qui facilite grandement la tâche lorsque nous gérons nombre de sites d'envergures diverses.

Figure 4-14

Suivi de l'indexation avec SeeUrank



SEO Chat met à disposition une multitude d'outils dont un qui permet de gérer rapidement les tests avec la commande `site:` vers divers moteurs (source : http://tools.seochat.com/tools/domain-indexed-pages/). Toutefois, les lacunes évoquées concernant cette commande restent vraies, et l'outil ne l'utilise pas sur tous les moteurs de recherche qui disposent de la fonctionnalité.

Notons tout de même que la plupart de ces outils présentent des inconvénients. D'une part, ils sont payants et parfois coûteux, ce qui peut être un frein si nous gérons seulement quelques sites de présentation, par exemple. D'autre part, des imprécisions demeurent entre la réalité de l'indexation et les pages notifiées dans les listings. Enfin, certains outils ne listent pas la totalité des pages pour des raisons généralement techniques, il arrive donc fréquemment que les sites de grande envergure ne puissent pas avoir une liste complète des pages enregistrées dans les index.

En définitive, nous remarquons qu'aucune technique n'est idéale pour suivre l'indexation des pages web, il faut souvent les coupler pour obtenir de meilleurs résultats…

Obtenir la fréquence de passage des robots

Dans la lignée du suivi des pages indexées (ou tout du moins de leur nombre présent dans les index des moteurs de recherche), il est important de savoir à quelle fréquence passent et repassent les robots d'indexation dans nos pages web.

Depuis que Google Caffeine a été mis en place en 2010, le monde de l'indexation a changé de visage et s'est considérablement métamorphosé. Dorénavant, les pages sont crawlées 24h/24 et les robots gèrent leur fréquence de passage de manière totalement indépendante afin d'améliorer la qualité de l'indexation selon la pertinence, la fraîcheur et la notoriété des pages. Avec les millions de pages visitées par jour, nous pouvons nous douter que notre site n'est pas toujours la priorité des robots...

Par conséquent, aucune page ne subit la même fréquence de passage selon son importance aux yeux des moteurs de recherche, et cela est aussi le cas pour d'autres moteurs que Google tels que Bing ou Yandex. Il convient donc de s'intéresser à la fréquence moyenne du crawl pour savoir quand et comment gérer son site.

Prenons un cas particulier, celui d'un site vitrine sur lequel nous publions des actualités tous les deux jours. La page générale présente trois articles dans l'ordre décroissant, lesquels sont ensuite déplacés dans les archives puis disparaissent au fond des bases de données. Nous avons donc chaque semaine une page d'actualités totalement nouvelle, il serait alors préférable que les robots indexent plusieurs fois par semaine la page pour que son rôle ne soit pas vain. En effet, si les robots crawlent uniquement une fois par quinzaine, trois pages d'actualités passeront à la trappe et les pages concernées ne seront ni indexées, ni valorisées...

Cet exemple est un peu extrême mais cela montre l'importance de la fréquence de passage. Il est totalement inutile de modifier son contenu tous les jours pour optimiser la fréquence de mise à jour (FreshRank chez Google) si les robots ne passent qu'une fois par semaine, beaucoup trop d'efforts seraient réalisés en vain...

La fréquence de passage des robots peut être suivie de plusieurs manières, nous allons les étudier.

Cache et fréquence de passage

Nous avons observé précédemment que le cache pouvait être utilisé pour obtenir la fréquence de passage des robots. Bien que cela ne soit pas très précis, cette technique permet d'obtenir une moyenne relativement juste sur le temps de passage des robots.

En effet, les pages de cache affichent des instantanés visuels avec la date de capture de l'information. Pour connaître la fréquence de passage, l'objectif est de revenir plusieurs fois sur la page de cache et de noter dans un tableau la date de l'instantané. En repassant plusieurs fois, nous pourrons déterminer l'intervalle moyen de temps écoulé entre les différentes versions de l'instantané.

Cette technique n'est pas toujours idéale mais force est de constater qu'elle rend bien des services pour obtenir rapidement une idée de la fréquence de passage des robots.

Il est également possible d'afficher certaines pages de cache dans un outil construit en PHP. Sur le même principe que la fonction du simulateur de robot que nous avons créée précédemment, nous pouvons réaliser un petit programme pour afficher les pages de cache pour une même URL. L'avantage est incontestablement le gain de temps pour passer d'un moteur à un autre, bien qu'ils ne soient pas nombreux à fournir cette fonctionnalité. En effet, il suffit de modifier la valeur dans le formulaire pour changer de moteur. Pour améliorer la fonction, il serait même possible de capter la date à la volée grâce à une expression régulière, puis de l'enregistrer dans un fichier de logs en CSV, par exemple…

Quels moteurs proposent des pages de cache ?

Actuellement, seuls deux moteurs (Google et Exalead) sur les dizaines de tests effectués ont permis d'utiliser des URL adaptées à l'outil de contrôle du cache.

D'autres moteurs comme Bing, Yahoo!, Baidu et Yandex fournissent aussi des pages de cache mais elles ne sont pas exploitables avec notre outil. En effet, nous utilisons un mécanisme autour d'une URL alors que ces moteurs fonctionnent autour d'une requête de recherche, ce qui ne nous permet pas de suivre directement les pages de nos sites web.

Le fichier de vérification doit contenir au moins le code suivant, il ne tient qu'à vous de l'améliorer pour aller encore plus loin selon vos envies et vos besoins.

```
<?php
// Fonction du simulateur de robot
// 1. URL à tester
// 2. Nom du moteur cible
function cacheControl($page = '', $moteur = 'google') {
    // Suppression du protocole (car inutile)
    if(preg_match("#^https?://#iU", $page)) {
        $page = preg_replace("#^https://#iU", "", $page);
    }

    // Personnalisation de l'URL de cache
    if(strtolower($moteur) == 'google') {
        $page = "http://webcache.googleusercontent.com/search?q=cache:".$page;
    }
    if(strtolower($moteur) == 'exalead') {
        $page = "http://www.exalead.com/search/web/cached/?url=".$page."&q=cache&qwr=cache";
    }

    // Activation de cURL
    $url = curl_init($page);

    // Options de cURL (retour des données et des en-têtes)
    curl_setopt($url, CURLOPT_RETURNTRANSFER, true);

    // Récupération du contenu
    $contenu = curl_exec($url);

    // Fermeture de cURL
    curl_close($url);
```

```
    // Affiche le code source complet
    echo $contenu;
}
?>

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Contrôle des pages de cache</title>
<style type="text/css"/>
* {margin:0; padding:0; font-size:100%}
#formulaire {background:#eee; padding:1%}
#formulaire h1 {font-size:1.8em; color:#004C54; margin-bottom:1em}
.bloc {margin-bottom:.8em}
.bloc label {display:block; font-weight:bold; padding:.1em;}
.bloc input, .bloc select {display:block; float:left; margin-right:1em; padding:.1em;
border:1px solid #ccc;}
.bloc input {width:260px;}
.bloc select {width:100px; text-align:center;}
#resultat {background:#fafafa; padding:1em}
#resultat h2 {font-size:1.3em; color:#198A95; margin-bottom:.8em}
#bouton input {padding:.2em .5em; font-weight:bold; border:1px solid #ccc; background:#fff;
color:#004C54}
#bouton input:hover {background:#004C54; color:#fff}
</style>
</head>

<body>
<div id="formulaire">
<h1>Contrôle des pages de cache</h1>
<form method="post">
    <div class="bloc">
    <input type="text" name="url" value="<?php if(isset($_POST['url']))
    { echo $_POST['url']; } ?>"/>
    <label for="url">URL de cache à vérifier</label>
    </div>

    <div class="bloc">
    <select name="moteur">
        <option value="google" <?php if(isset($_POST['moteur']) && $_POST['moteur']
        == 'google') { echo 'selected="selected"'; } ?>>Google</option>
        <option value="exalead" <?php if(isset($_POST['moteur']) && $_POST['moteur']
        == 'exalead') { echo 'selected="selected"'; } ?>>Exalead</option>
    </select>
    <label for="moteur">Moteur de recherche</label>
    </div>
```

```
    <p id="bouton"><input type="submit" name="submit" value="Vérifier le cache"/></p>
</form>
</div>

<div id="resultat">
<?php if(isset($_POST['submit']) && !empty($_POST['url'])) {      ?>
<?php // Traitement des données
    $url = htmlspecialchars($_POST['url']);
    $moteur = htmlspecialchars($_POST['moteur']);
?>
<h2>Page de cache de l'adresse <?php echo $url; ?></h2>
<p><?php cacheControl($url, $moteur); ?></p>
<?php } ?>
</div>
</body>
</html>
```

Figure 4-15

Vérification du cache sur Exalead avec PHP

Statistiques de serveur

Une autre méthode pour suivre le passage des robots est de vérifier fréquemment les fichiers de logs déposés sur notre serveur. En effet, les hébergeurs mettent généralement à disposition des outils de statistiques pour donner à leurs clients des informations intéressantes sur l'activité relative à leurs sites web comme :

- les données générales et datées sur le trafic (visites, visiteurs uniques, nombre de hits, nombre de pages vues, liste des pages vues, durée des visites, bande passante consommée…) ;
- l'historique des informations (sur un mois, une semaine, un jour…) ;
- les données sur les sources des visites (pays hôte, adresses IP des visiteurs, système d'exploitation et navigateurs utilisés, liste des *referers*…) ;
- les informations sur les robots (nom, bande passante téléchargée, date du dernier passage…) ;

- la liste des mots et expressions clés qui ont permis d'aboutir sur votre site (plutôt imprécis en général) ;
- les erreurs de lecture rencontrées (codes d'erreurs).

Parmi les informations disponibles, nous pouvons utiliser celles relatives aux robots d'indexation pour extraire des données intéressantes comme la date du dernier passage pour chaque robot mais aussi la quantité de bande passante téléchargée.

Comment suivre et distinguer les robots ?

Il est également possible de suivre les robots avec Google Analytics, mais cela demande un paramétrage précis.

Il arrive fréquemment que de nombreux robots soient inconnus dans les données disponibles sur les serveurs, ce ne sont pas des moteurs de recherche en règle générale, mais uniquement des parseurs ou *sniffers* en tous genres...

Figure 4-16

Informations du serveur sur les robots

Visiteurs Robots/Spiders (Top 10) - Liste complète - Dernière visite

27 robots différents*	Hits	Bande passante	Dernière visite
Unknown robot (identified by 'crawl')	316+5	2.77 Mo	19 Juin 2014 - 16:09
Unknown robot (identified by 'bot*')	207+29	1.40 Mo	19 Juin 2014 - 17:35
Googlebot	102+34	1.28 Mo	19 Juin 2014 - 17:08
Unknown robot (identified by empty user agent string)	132	3.47 Mo	19 Juin 2014 - 17:12
Unknown robot (identified by '*bot')	69	6.74 Mo	19 Juin 2014 - 17:30
Feedfetcher-Google	40	138.36 Ko	19 Juin 2014 - 17:28
Voila	40	266.20 Ko	19 Juin 2014 - 17:08
Java (Often spam bot)	30	582.78 Ko	19 Juin 2014 - 14:48
Unknown robot (identified by 'robot')	21	1.29 Mo	19 Juin 2014 - 17:01
Unknown robot (identified by 'spider')	16+2	1.21 Mo	19 Juin 2014 - 17:20
Autres	94+21	2.96 Mo	

* Les robots présentés ici sont à l'origine de hits ou de traffic "non vus" par les visiteurs donc non représentés dans les autres tableaux. Les nombres après le + indiquent les hits avec succès sur les fichiers "robots.txt".

La fréquence de passage des robots peut être calculée en vérifiant régulièrement ces fichiers de logs déposés sur le serveur. En comparant les dates pour chaque robot, nous pouvons obtenir un intervalle de temps correspondant à la fréquence de passage, mais nous devons rester vigilants. En effet, les informations ne précisent pas quelle page a été visitée par les robots, ce qui correspond donc en réalité à leur fréquence de passage générale et non pas à la fréquence de crawl relative à chaque page. Le traitement des données est donc à prendre avec des pincettes et doit être mesuré.

La fréquence de passage obtenue permet juste de déduire combien de fois un robot vient sur notre site. En revanche, nous ne pouvons pas savoir si un robot parcourt chaque page tous les jours ou toutes les semaines, par exemple. La technique des pages de cache est bien plus appropriée pour extraire ce type d'information.

Enfin, les données disponibles par les serveurs avec AwStats, par exemple, montrent souvent la quantité de bande passante crawlée par les robots. Cette information peut s'avérer intéressante, notamment lorsque nous lançons un nouveau site. En effet, nous pouvons analyser ces données de plusieurs manières.

- Si de nombreux mégaoctets de données ont été crawlés par les robots, cela signifie que les robots ont vu un certain nombre de pages ou qu'ils ont probablement enregistré beaucoup de données dans leur index.
- Si peu de données ont été lues par les robots, les causes peuvent être multiples :
  - le robot connaît déjà les pages visitées et n'a rien chargé de nouveau (aucune mise à jour) ;

- le robot a enregistré seulement les quelques mises à jour relatives à des pages ;
- le robot a rencontré des problèmes de crawl.

Les fichiers de logs fournis par les hébergeurs peuvent vraiment fournir une aide pour suivre les statistiques relatives aux sites web. Il convient juste de veiller à ne pas traiter les informations avec la mauvaise méthode au point de faire ressortir des contre-vérités.

Utiliser PHP pour suivre le crawl

Dans les faits, la meilleure façon de suivre les robots et d'obtenir une fréquence de passage complète et réaliste est de passer par un script personnalisé exécuté dans les pages web. La solution passe par un code de détection des robots lancé dans toutes les pages pour capter la date de passage voire d'autres informations si besoin.

Nous allons créer deux fonctions distinctes dans un même fichier pour aller plus loin que les autres techniques et éviter d'avoir un listing dans lequel seraient mélangés les URL crawlées et les robots. Ce fichier doit être placé dans une zone répétée des pages (pied de page ou en-tête, par exemple) pour qu'aucune page ne soit exclue de l'analyse.

La première fonction va créer un dossier par page visitée avec un fichier CSV pour chaque robot. Ainsi, nous obtenons des dossiers spécifiques pour chaque page visitée qui reçoivent des informations relatives à chaque robot scruté par le script. Nous pouvons donc facilement lister les dates de passage par robot et donc leur fréquence de passage exacte, à la seconde près…

La fonction écrit également la page crawlée pour éviter les doutes, ainsi que le nom complet du robot qui a visité la page, ce qui permet de déduire d'autres informations intéressantes.

Figure 4-17

Dates des différents passages de Bingbot sur une même page pour déterminer la fréquence de crawl

	A	B	C
1	mozilla/5.0 (compatible; bingbot/2.0	19/06/2014 18:22:06	/creation-blog-professionnel.php
2	mozilla/5.0 (compatible; bingbot/2.0	21/06/2014 17:37:09	/creation-blog-professionnel.php
3	mozilla/5.0 (compatible; bingbot/2.0	24/06/2014 19:03:26	/creation-blog-professionnel.php
4	mozilla/5.0 (compatible; bingbot/2.0	27/06/2014 18:38:56	/creation-blog-professionnel.php

Au début de notre fichier PHP, nous devons ajouter une liste de robots à suivre ainsi que le nom du répertoire dans lequel seront enregistrés les fichiers d'analyse.

```
// Liste des robots et nom du répertoire
$listeRobots = array("ask", "jeeves", "baiduspider", "exabot", "gigabot", "googlebot",
"googlebot-image", "inktomi slurp", "mediapartners-google", "bingbot", "slurp", "teoma",
"voila", "yandex", "yahoo");
$repertoireLogs = 'logs/';
```

Voici le code de la première fonction :

```
// Détection de la fréquence de passage par page
// 1. Liste des robots à suivre
```

```
// 2. Répertoire pour le dossier du journal (avec slash final)
function getCrawlFrequency($listeRobots,$repertoireLogs = 'logs/'){
    $crawler = strtolower($_SERVER["HTTP_USER_AGENT"]);
    // Création du journal si inexistant
    if(!is_dir($repertoireLogs)) {
        mkdir($repertoireLogs, 0705);
    }

    // On boucle le test pour tous les robots
    foreach($listeRobots as $robot) {
        if(preg_match('#'.$robot.'#iU', $crawler)) {
        // Récupération dynamique de la date
        $dateActuelle = date('d/m/Y H:i:s');

        // Récupération dynamique de l'URL crawlée
        $pageCrawlee = $_SERVER['REQUEST_URI'];

        // Nom du répertoire pour chaque page crawlée
        $nomRep = str_replace("/", "-", $_SERVER['REQUEST_URI']);

        // Création d'un répertoire pour la page d'accueil
        if($_SERVER['REQUEST_URI'] == "/") {
               $nomRep = 'accueil';
        }

        // Suppression du premier caractère inutile
        if(substr($nomRep, 0, 1) == "-") {
            $nomRep = substr($nomRep, 1);
        }
        $donnees = array($crawler, $dateActuelle, $pageCrawlee);

        // Création du répertoire si inexistant
        $rep = $repertoireLogs.$nomRep.'/';
        if(!is_dir($rep)) {
               mkdir($rep, 0705);
        }

        // Noms des fichiers à créer
        $nomFichier = $rep.$robot."-log.csv";

        // Création et remplissage du fichier
        $fichier = fopen($nomFichier, 'a');
        fputcsv($fichier, $donnees, ";");
        fclose($fichier);
        }
    }
}
// Lancement de la fonction de surveillance
getCrawlFrequency($listeRobots, $repertoireLogs);
```

La seconde fonction ajoutée dans le même fichier d'analyse apporte d'autres informations sur le crawl. En effet, cette fonction ne distingue pas les pages mais uniquement les robots, ce qui permet de savoir quelles pages ont été crawlées avec leur date de passage. Par conséquent, nous pouvons déduire et calculer un intervalle de crawl global pour chaque robot ainsi que la qualité d'indexation pour chacun d'entre eux.

La seconde fonction reprend globalement le même principe que la première que nous avons créée. Voici son code complet :

```
// Suivi complet du passage des robots
// 1. Liste des robots à suivre
// 2. Répertoire pour le dossier du journal (avec slash final)
function getCrawlPages($listeRobots, $repertoireLogs = 'logs/') {
$crawler = strtolower($_SERVER["HTTP_USER_AGENT"]);
    // Création du répertoire si inexistant
    if(!is_dir($repertoireLogs)) {
        mkdir($repertoireLogs, 0705);
    }

    // On boucle le test pour tous les robots
    foreach($listeRobots as $robot) {
        if(preg_match('#'.$robot.'#iU', $crawler)) {
        // Variables utiles
        $dateActuelle = date('d/m/Y H:i:s');
        $pageCrawlee = $_SERVER['REQUEST_URI'];
        $donnees = array($crawler, $dateActuelle, $pageCrawlee);

        // Noms des fichiers à créer
        $nomFichier = $repertoireLogs.$robot."-log.csv";

        // Création et remplissage du fichier
        $fichier = fopen($nomFichier, 'a');
        $fichier = fopen($nomFichier, 'a');
        fputcsv($fichier, $donnees, ";");
        fclose($fichier);
        }
    }
}
// Lancement de la fonction de surveillance
getCrawlPages($listeRobots, $repertoireLogs);
```

Figure 4-18

Dates et fréquence des passages de Voilabot

	A	B	C
1	mozilla/5.0 (windows nt 5.1; u; win64; fr; rv:1.8.1) voilabot beta 1.2	19/06/14 17:08:04	/agence-internet-formation.php
2	mozilla/5.0 (windows nt 5.1; u; win64; fr; rv:1.8.1) voilabot beta 1.2	19/06/14 17:41:47	/programme-formation-newsletter.php
3	mozilla/5.0 (windows nt 5.1; u; win64; fr; rv:1.8.1) voilabot beta 1.2	19/06/14 18:02:06	/formation-web-internet.php

Pour rendre le script fonctionnel, il faut d'inclure le fichier dans une zone répétée des sites avec une fonction PHP :

```
<?php include_once('frequence-crawl.php');?>
```

Enfin, il suffit de créer une page d'accès aux fichiers CSV ou tout simplement récupérer les données sur le serveur avec un client FTP comme FileZilla ou Cyberduck. Il ne nous reste plus qu'à analyser les données et calculer la fréquence précise de crawl par page ou l'intervalle général de crawl en fonction des robots.

Suivre le positionnement

Il existe également pléthore d'outils pour analyser les positions des sites web sur diverses requêtes de recherche. Seulement une infime partie d'entre eux est gratuite et ne rend pas toujours les résultats escomptés. Il faut généralement se tourner vers des variantes payantes bien plus efficaces et professionnelles si vous voulez suivre le positionnement avec plus de précision.

Le suivi du positionnement passe par plusieurs étapes et aucune technique n'est idéale pour mesurer le succès de notre travail. Nous allons étudier plusieurs méthodes, de la plus fastidieuse à la plus précise pour obtenir un rapport de positionnement qualitatif. Nous verrons que ce n'est pas si simple que cela…

Du mouvement dans les SERP ?

Les spécialistes rappellent souvent que le positionnement ne veut plus tout dire et n'est pas toujours la garantie d'un retour sur investissement en matière de SEO. Dans l'idée, c'est entièrement vrai mais dans les faits, nous remarquons qu'être bien positionné a encore un impact majeur sur le trafic obtenu par les sites web.

En réalité, ce sont nos objectifs qui ne sont plus les mêmes qu'avant, et c'est ce qui explique le rôle plus mesuré du classement dans les SERP en comparaison au référencement du passé. De nos jours, les canaux d'entrée sont multiples et il n'est pas toujours primordial d'être le premier sur une multitude de requêtes…

Il n'est pas toujours nécessaire d'être premier pour obtenir de bons résultats mais il ne faut pas non plus être en troisième page des SERP sur toutes les requêtes. D'autres canaux comme les moteurs spécifiques (images, actualités…), le développement d'outils et d'applications (mobiles ou non), les médias sociaux ou les forums spécialisés apportent une visibilité accrue que nous mésestimons souvent mais qui peut représenter une bonne part du gâteau.

Trois problèmes majeurs expliquent que le positionnement des pages ne peut pas toujours être suivi comme il se doit.

- Les SERP subissent des fluctuations naturelles au fil du temps (selon les moteurs de recherche), et même si les pages conservent une pertinence équivalente aux yeux des moteurs.
- Les URL des moteurs de recherche affichées dans les SERP ou les referers (URL de la page par laquelle nous arrivons sur une autre page) ne donnent pas toutes des indications sur le positionnement. Par exemple, nous avons précédemment observé les URL de cache et aucune d'entre elles ne fournit les mêmes informations. Il en est de même pour les résultats naturels, seuls quelques moteurs donnent des informations que nous pouvons récolter (uniquement dans certains cas de figure).
- Plusieurs moteurs ont décidé de sécuriser leur outil en passant par SSL et https://. Ce changement est significatif car il a permis aux développeurs de masquer les requêtes de recherche lorsqu'un internaute

clique sur un lien naturel, ce qui ne permet plus de récupérer cette donnée essentielle et donc de savoir quelles expressions ont été tapées lors des recherches…

Fluctuations naturelles

Les moteurs de recherche ne fixent pas les positions des pages sur des requêtes de recherche, l'indexation incessante ainsi que les mouvements dus au positionnement évolutif de millions de pages ne permettent pas de conserver une place stable dans le classement. Les filtres et pénalités qui s'ajoutent en surcouche génèrent encore plus d'incertitudes autour du positionnement des pages.

Partant de ce constat, il est bien difficile de déterminer la position d'une page de manière certaine et précise. En effet, il n'est pas rare que du jour au lendemain, une même page bouge de quelques positions. Généralement, les pages bien fixes dans les SERP répondent à des requêtes précises qui imposent peu de mouvement ou ont une telle pertinence aux yeux des moteurs qu'elles ne peuvent pas être délogées, mais cela est de moins en moins vrai…

Des pages disparues dans la nature…

Il arrive fréquemment que des pages disparaissent des SERP sans aucune raison. Souvent, nous pensons qu'il s'agit de pénalités mais il n'en est rien, cela s'explique souvent par des modifications temporaires d'un morceau d'algorithme ou par des tests effectuées sur des requêtes précises. Cela fait partie de la vie d'un moteur de recherche, il suffit de regarder plusieurs jours durant si le problème persiste afin de savoir s'il s'agit ou non d'une sanction.

Ces mouvements continuels causés par les mécanismes profonds des moteurs de recherche ne sont pas les seuls qui expliquent à quel point il faut être mesuré lorsque nous parlons de classement. En effet, nombre de facteurs impliquent des différences parfois très importantes de positionnement selon le contexte de la recherche. Voici une liste des critères qui modifient le classement :

- présence ou non de résultats issus de la recherche universelle (actualités, photos, vidéos…) ;
- historique des recherches enregistré par les moteurs ;
- historique, cache et cookies installés par le navigateur ;
- connexion ou non à un compte utilisateur sur Google (avec Gmail), sur Bing (avec Outlook) ou autres (Yandex…) ;
- langue de la recherche ;
- géolocalisation des requêtes (pays et ville) ;
- datacenter utilisé lors des recherches ;
- nombre de pages présentées dans les résultats de recherche.

L'ensemble de ces facteurs de recherche totalement indépendants de notre volonté affecte le positionnement final et ne permet donc pas de réaliser un suivi précis et réaliste des résultats.

Par exemple, nous pouvons très bien être en première place à Brest si notre localisation est approchante et nous retrouver à la fin de la deuxième page à Nice car notre site n'est plus assez pertinent à l'autre bout de la France sur une requête précise.

Dans d'autres cas, nous pouvons être confrontés à la recherche universelle qui perturbe totalement la visibilité des liens organiques. En outre, il arrive souvent que nous sous-estimions l'impact des historiques de recherche et du navigateur, ils peuvent vraiment nuire au suivi du positionnement.

Enfin, un dernier souci empêche de mesurer l'impact du classement dans les SERP. Selon les moteurs et les requêtes, les pages de résultats n'affichent plus nécessairement dix résultats comme c'est de coutume depuis des années.

Figure 4-19

Quinze résultats dans des pages de Bing

Par exemple, il arrive sur Google de trouver des pages de quatre ou sept liens organiques tandis que Bing peut en faire apparaître sept ou encore quinze. Par conséquent, l'importance du positionnement devient relative car chaque cas est différent et ce n'est pas forcément la position qui fera le succès des pages visibles.

Problème des URL referers

Le problème majeur du suivi du positionnement est la gestion des adresses de référence *(URL referers)*. Tous les moteurs ne fournissent pas d'indications précises sur les requêtes de recherche ou sur le positionnement relatif aux pages visitées, il est donc impossible de capter les places précises pour chaque outil, ce qui en soi fausse déjà beaucoup le suivi…

Qui plus est, les quelques moteurs qui fournissent des positions dans les URL referers ne le font pas tous de la même manière. La seule chance que nous avons réside dans le fait que Google est le plus précis sur ce point, et comme il reste le leader écrasant du marché, nous pouvons obtenir des données relativement précises.

D'autres moteurs comme Ask, Yandex, Aol, Lycos, Baidu ou Bing ne fournissent pas la position exacte mais le numéro de page ou plutôt le nombre de résultats par page, ce qui permet d'obtenir un positionnement à la page près, et non sur des places précises comme sur Google. Il faut également tenir compte du fait que chaque moteur a ses spécificités, ses paramètres et que le comptage des pages n'est pas toujours identique.

Le problème du protocole SSL s'ajoute à celui des paramètres cachés d'URL. En effet, Google, Bing et Yahoo! notamment font en sorte de masquer la requête des internautes voire parfois l'URL referer en totalité (Bing s'il est en https://, par exemple). Le plus malin est le français Qwant dont la sécurité fait partie de ses spécialités, à tel point que les liens organiques passent par des redirections pour cacher les paramètres valables pour notre suivi.

Depuis de nombreux mois, Google a mis en place un système pour cacher les requêtes de recherche des internautes. Dorénavant, le moteur affiche « not provided » quand il ne peut pas déterminer les mots-clés tapés. Cette fonctionnalité a été généralisée à toutes les personnes connectées à un compte Gmail mais également aux utilisateurs du navigateur Google Chrome. Depuis début 2014, le phénomène semble s'accroître puisque Google a annoncé que le not provided allait s'étendre même pour les liens sponsorisés, nous tendons donc vers un avenir sans suivi des requêtes…

Selon les dires des officiels de Google, le moteur ne peut pas fournir les requêtes lorsqu'il est en https://. Sur le principe, ce n'est pas totalement faux puisque l'intérêt est de cacher la requête, mais lorsque nous regardons cela de plus près, nous constatons qu'il s'agit plutôt d'une supercherie maligne…

Analysons une URL provenant d'une recherche sécurisée grâce à la variable `$_SERVER['HTTP_REFERER']`; en PHP. Voici le résultat :

> http://www.google.fr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CD4QFjAA&url=http://www.site.fr/&ei=7SWkU7WyOIiW0AXqnoDYAQ&usg=AFQjCNF8e6lo57CSYF60DyxtpKYODTig-A&sig2=lKvSSDty36rtGxRy3k8FDA&bvm=bv.69411363,d.d2k

Maintenant, si nous retournons dans Google et que nous saisissons exactement la même requête en supprimant juste le « s » du protocole https://, nous obtenons la nouvelle référence suivante :

> http://www.google.fr/url?sa=t&rct=j&q=requete&source=web&cd=1&ved=0CDsQFjAA&url=http://www.site.fr/&ei=yCakU7H1AuWc0QXhxYHAAQ&usg=AFQjCNF8e6lo57CSYF60DyxtpKYOD
> Tig-A&sig2=2FjgNITzXLQrTn25kUTW7w

Voyez-vous où nous voulons en venir ? Dans le premier cas, le paramètre `q` de la requête est vide tandis que la requête apparaît dans le second test. Google nous affirme qu'il ne peut pas passer la requête de recherche alors qu'il est capable de transmettre tous les autres paramètres sans aucune difficulté. Dans les faits, il est tout à fait possible de passer l'information mais il s'agit d'un choix délibéré des responsables du moteur. L'excuse de l'impossibilité ne tient donc pas et il aurait été plus honnête de l'admettre…

Bing, Yahoo! et les URL de référence

Bing ne fournit aucune URL referer lors des recherches sécurisées, ce qui bloque toutes les informations. Yahoo! est encore plus efficace puisqu'il crypte la plupart des données donc nous ne pouvons rien dégager des adresses de référence.

En d'autres termes, trouver les mots-clés tapés ainsi que les positions relèvent presque du miracle. Nous ne pourrons plus suivre entièrement et facilement nos visiteurs et nos positions comme ce fut le cas pendant des années, il faut bien en prendre conscience. Partant de ce constat, nous allons présenter quelques outils mais retenez bien que les données affichées ne seront pas toujours exemptes de défauts…

Google sécurise ses URL referers

Depuis le mois d'octobre 2015, les liens sponsorisés issus de Google Adwords ne transmettent plus d'URL de référence (source : http://goo.gl/sCrre6). Par conséquent, nous obtenons en réponse uniquement le nom de domaine de Google et plus du tout les autres informations. Google affirme que cela est fait pour protéger davantage ses utilisateurs et mieux sécuriser le moteur.

En revanche, la firme n'a rien indiqué au sujet des résultats organiques présents dans les SERP. Dans les faits, il arrive très fréquemment que Google masque les URL referers depuis la mi-2015 dans ces liens naturels, pour les mêmes raisons. Certains des codes PHP ou des filtres Google Analytics présents dans le livre sont donc parfois amputés de certains résultats à cause de ce changement mis en place aussi soudainement que discrètement par Google.

Suivre les positions et les requêtes

Outils et rapports

Il existe une multitude d'outils gratuits ou payants pour générer des rapports en référencement et positionnement. Tous ne sont pas d'égale qualité mais ils présentent un grand nombre d'informations qui peuvent nous intéresser. De plus, ils fournissent des fichiers de logs ou de génèrent des rapports de positionnement parfois complets, ce qui s'avère bien pratique lorsque nous gérons des masses de clients ou tout simplement si nous ne voulons pas perdre de temps…

Voici une liste non exhaustive d'outils qui peuvent répondre à nos besoins, nous en présenterons quelques-uns par la suite :

- SeeUrank de Yooda : http://www.yooda.com/produits/soft/ ;
- Positeo : http://www.positeo.com/check-position/ ;
- Myposeo de G4Interactive : http://fr.myposeo.com ;
- SEO Soft de Webmaster-rank : http://goo.gl/DdSHMM ;
- Allorank : http://www.allorank.com ;
- Rank Tracker : http://www.link-assistant.com/rank-tracker/ ;
- Ranks.fr de Kiwax : http://www.ranks.fr/ ;
- SEMrush : http://www.semrush.com ;
- GammaSEOTools : http://www.gammaseotools.com ;
- SEO Toolkit de Trellian : http://www.trellian.fr/seotoolkit/ ;
- AgentWebRanking : http://www.agentwebranking.com.

Tous ces outils sont de qualité différente mais il faut avouer que certains d'entre eux s'en sortent mieux que d'autres comme Positeo, Myposeo, Ranks.fr ou encore SEO Soft.

Figure 4-20

Suivi du positionnement avec Allorank

Globalement, le suivi des positions est assez honorable même s'il peut exister quelques variantes avec la réalité. En effet, nous ne pouvons pas paramétrer les types de recherches comme bon nous semble donc les fluctuations des SERP entrent en ligne de compte.

Nous ne savons pas si l'outil procède à une recherche à Paris ou Marseille, sur quel datacenter il se connecte, s'il gère bien les historiques, etc. Par conséquent, les chiffres annoncés ont parfois quelques positions de décalage avec ce que nous pourrions trouver avec une méthode manuelle, par exemple.

En revanche, ces outils ont un net avantage sur d'autres méthodes puisque les plus efficaces d'entre eux n'utilisent pas forcément les paramètres glissés dans les URL de référence pour déterminer la position. Les meilleurs possèdent leur propre robot de crawl des pages de résultats et dès que celui-ci trouve une URL donnée, les moteurs récupèrent la position dans les SERP.

Réaliser un suivi du positionnement par la technique

Il s'agit de mécanismes assez techniques qui pourraient par exemple être réalisés à l'aide du module cURL en PHP, mais il existe d'autres variantes tout aussi efficaces en Java notamment. L'idée est de lancer une requête dans les SERP via un code qui permet de tester chaque page de résultats en boucle jusqu'à trouver l'URL recherchée. Cela fonctionne bien mais les moteurs tentent de se prémunir contre ce phénomène en bloquant les accès aux outils…

Présentons quelques outils en détails…

WebRankChecker

WebRankChecker est un outil entièrement gratuit dont l'utilisation est simple. Il suffit de mentionner une URL précise, puis une expression particulière et enfin de choisir le moteur de recherche à tester pour obtenir le positionnement.

Figure 4-21

Paramétrage de WebRankChecker

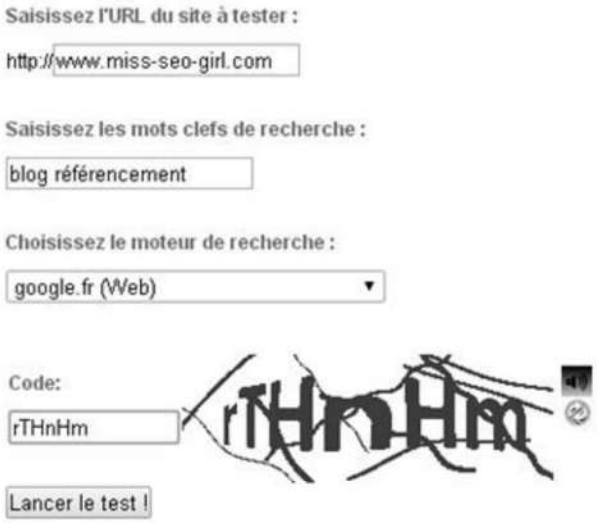
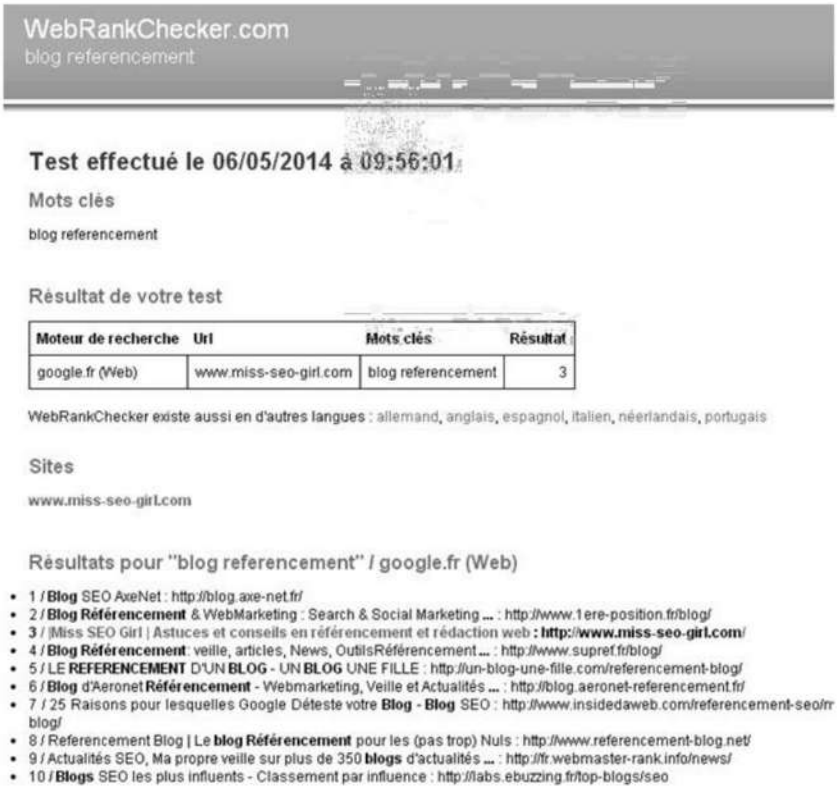


Figure 4-22

Positionnement et suivi concurrentiel



Si nous comparons les résultats avec d'autres outils du marché, nous remarquons que la position affichée dans la page du rapport est souvent identique à une ou deux positions près. Nous pouvons donc confirmer que l'outil est plutôt de bonne facture.

En plus de la position récupérée, WebRankChecker présente les dix résultats environnants autour de notre URL sur la requête donnée, ce qui permet d'avoir un rapide coup d'œil sur la concurrence.

Les limites des outils de suivi

Le principal inconvénient de cet outil mais aussi de certains de ses concurrents est d'être monotâche. De fait, il est impossible de procéder à un suivi complet très rapidement. Il faut procéder requête par requête et page par page, ce qui peut rapidement s'avérer fastidieux. Parfois, un travail manuel prend presque autant de temps pour un résultat souvent plus précis…

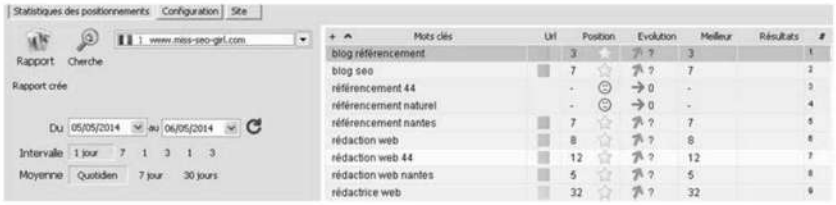
SEO Soft

SEO Soft est un logiciel gratuit qui doit être téléchargé et installé sur un ordinateur. Il est plutôt performant et fiable et permet de gagner plus de temps qu'avec les outils présentés précédemment.

Son interface est quelque peu archaïque mais la configuration permet de suivre les résultats de plusieurs manières : soit avec des courbes, soit avec des tableaux précis. Nous pouvons aisément suivre l'évolution du positionnement au cours du temps, ce qui s'avère impossible avec Positeo ou WebRankChecker, à moins de créer nos propres fichiers de comparaison.

Figure 4-23

Suivi des positions avec SEO Soft



Pour configurer l'outil, il faut procéder par étapes.

1. Cliquez sur l'onglet *Configuration*.
2. Saisissez l'URL ainsi que les mots et expressions clés à analyser.
3. Cliquez sur l'icône *Cherche* dans l'onglet *Statistiques des positionnements*.
4. Laissez l'outil procéder à sa première analyse. Ensuite, il suffit de multiplier les analyses pour obtenir des comparaisons de données sur la durée.

Figure 4-24

Paramétrage de SEO Soft

Positeo

Positeo est un autre outil gratuit qui vient concurrencer directement WebRankChecker, à la différence qu'il est paramétrable et qu'il vérifie la position sur plusieurs datacenters. De ce fait, les résultats obtenus sont plus précis et de meilleure qualité.

Positeo permet de paramétrer le pays et la langue de recherche. Pour le reste, le fonctionnement est identique à WebRankChecker, il suffit de rentrer une requête et une URL cible pour que l'outil actionne l'analyse. En revanche, Positeo ne crawle pas tous les moteurs de recherche, il ne s'intéresse qu'à des datacenters de Google. Ceci peut être un désavantage dans certains pays mais en France, la part dominante de Google est telle que le service en ligne répond parfaitement à nos besoins.

Contrer les blocages causés par les moteurs

Prenez garde à ne pas effectuer trop d'analyses consécutives car il arrive que l'outil bloque l'IP du client ou empêche que nous fassions plusieurs requêtes consécutives. Dans ce cas, il peut être intéressant de passer derrière un proxy pour éviter ce désagrément, bien que la solution ne fonctionne pas toujours… Vous pouvez trouver une liste de proxies à l'adresse suivante :

http://www.adresseip.com/liste-de-proxy.php

Figure 4-25

Suivi des positions sur plusieurs datacenters Google

SeeUrank

Enfin, terminons notre tour d'horizon des outils avec le payant mais qualitatif SeeUrank de Yooda. En réalité, cet outil va bien plus loin qu'une simple analyse du positionnement, il s'agit d'un vrai couteau suisse qui permet plusieurs suivis : suivre le positionnement des pages web sur un nombre illimité de mots-clés, analyser les backlinks (qualité, notoriété…) et les pages HTML en profondeur, réaliser une veille concurrentielle et analyser la stratégie SEO des concurrents, auditer un site pour trouver ses freins et optimiser sa structure, suivre l'indexation.

D'autres outils n'ont rien à envier à SeeUrank tels que Ranks.fr, Myposeo ou encore SEMrush, mais pour la plupart, il faut sortir le chéquier et les sommes peuvent vite grimper.

Par exemple, Allorank fonctionne avec un système de crédits pour chaque crawl donc nous pouvons rapidement dépenser des centaines d'euros si nous utilisons ces outils dans notre vie professionnelle. De ce fait, le suivi du positionnement n'est pas à la portée de toutes les bourses et il est parfois préférable de procéder par soi-même pour obtenir des rapports de référencement gratuits.

SEMrush

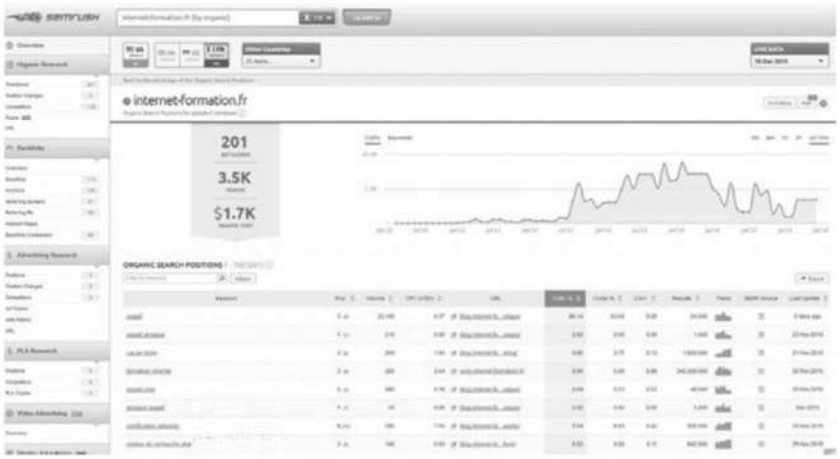
SEMrush est un outil développé aux États-Unis et en Russie pour effectuer de nombreux suivis SEO, pas uniquement le suivi du positionnement, bien que cela soit sa plus grande qualité. La version gratuite est limitée, mais il est possible d'opter pour les options Premium. Par exemple, la version Pro de l'outil offre plusieurs possibilités :

- suivi avancé du positionnement (classement des pages web en cours, mais aussi mouvements enregistrés dans les SERP), avec enregistrement de projets, exportation des rapports, envoi du suivi par e-mails (…) ;

- suivi des backlinks (bien que cela ne soit jamais vraiment réaliste en comparaison avec les Webmaster Tools des moteurs de recherche) ;
- possibilité de mener des benchmarks concurrentiels (analyse du positionnement des concurrents sur les mêmes requêtes, etc.) ;
- rapport de recherche sur des requêtes données (suggestions de mots-clés mais aussi analyse en profondeur des requêtes) ;
- outil d'évaluation de la faisabilité du positionnement sur des mots-clés donnés ;
- outil d'audit de site web.

Figure 4-26

Suivi du positionnement détaillé avec SEMrush Pro



Utiliser les Webmaster Tools pour suivre les requêtes et les positions

Les outils pour webmasters fournis par les principaux moteurs de recherche peuvent nous aider à suivre le positionnement et les requêtes tapées par les internautes. Les données sont loin d'être très précises mais permettent de contrer au moins partiellement le problème grandissant des not provided.

Dans la Google Search Console, vous pouvez suivre les requêtes de recherche tapées par les internautes via le menu *Trafic de recherche>Requêtes de recherche*. L'outil fournit aussi des informations sur le nombre d'impressions et le positionnement de la page associée à la requête tapée, ce qui peut donner des indications complémentaires pertinentes.

Figure 4-27

Suivi des requêtes de recherche du nombre de pages via la Google Search Console

La Toolbox de Bing donne aussi des données du même type (en moindre quantité à cause de la taille de l'index de Bing) via le menu *Rapports et données>Recherche par mots-clés*. L'outil affiche le nombre de clics et d'impressions, le classement des pages web ainsi que sa tendance (à la hausse ou non) mais aussi des infobulles pour afficher le CPC moyen de la requête en référencement payant.

Figure 4-28

Suivi des requêtes et du positionnement sur Bing

Procéder à un suivi manuel

S'il existe bien une méthode qui fonctionne, c'est certainement le suivi du positionnement de façon manuelle. Certes, la technique s'avère très vite fastidieuse et usante mais si vous n'avez pas des quantités de sites web à analyser ou des masses de requêtes à suivre, cette méthode reste de loin la plus précise.

Il est possible de personnaliser totalement la méthode de recherche, de changer la localisation, de jouer avec ou sans l'historique du navigateur, etc. Par conséquent, nous pouvons vraiment obtenir des positionnements précis aux dates que nous souhaitons.

Il est tout de même conseillé de suivre quelques indications pour éviter d'être trompé sur le positionnement affiché dans les SERP. Il est notamment préférable de supprimer l'historique du navigateur, de se déconnecter d'un compte Google ou Bing ou encore de passer par la navigation privée dans les navigateurs.

Pour être encore plus rassuré, il peut être intéressant de procéder au suivi avec un autre navigateur que celui utilisé fréquemment sur la machine, mais également de choisir une ville éloignée de la nôtre pour obtenir un positionnement digne d'un visiteur « neutre ».

La technique consiste à se créer un rapport de positionnement avec un tableur comme Open Office Calc ou Excel dans lequel nous insérons pour chaque requête la position ainsi que l'URL trouvées. En procédant de la sorte, nous pouvons rapidement obtenir un tableau complet et précis des données relatives au positionnement.

Utiliser PHP pour réaliser des rapports de positionnement

La programmation web peut nous venir en aide pour générer des rapports de positionnement. Nous utiliserons une nouvelle fois PHP mais d'autres langages comme Java permettent aussi d'effectuer ce type de suivi dynamique. Dans les faits, nous pourrions coder deux types de programmes pour effectuer le suivi du positionnement.

- Un crawler codé avec cURL pour accéder aux URL des moteurs et effectuer des actions (`GET`, `POST`...) afin de passer de page en page jusqu'à trouver des URL précises (donc des chaînes de caractères correspondant à des adresses web). Cette technique serait la plus efficace pour obtenir avec assurance un positionnement pour chaque page mais elle s'avère très technique et peut demander énormément de ressources au serveur pour être idéale et fonctionner parfaitement.
- Une fonction placée dans une zone répétée des pages (en-tête ou pied de page...) qui récupère de manière dynamique les informations trouvées dans les URL referers des moteurs.

Ces deux techniques ne permettent pas de réaliser les mêmes analyses. La première, gourmande en ressources, n'est utilisée que pour trouver le positionnement précis des pages. Les outils présentés précédemment conviennent tout à fait pour effectuer ce type de suivi. La seconde est moins précise sur le positionnement des pages mais permet d'obtenir pléthore d'autres informations sur le trafic web ou encore sur les requêtes de recherche.

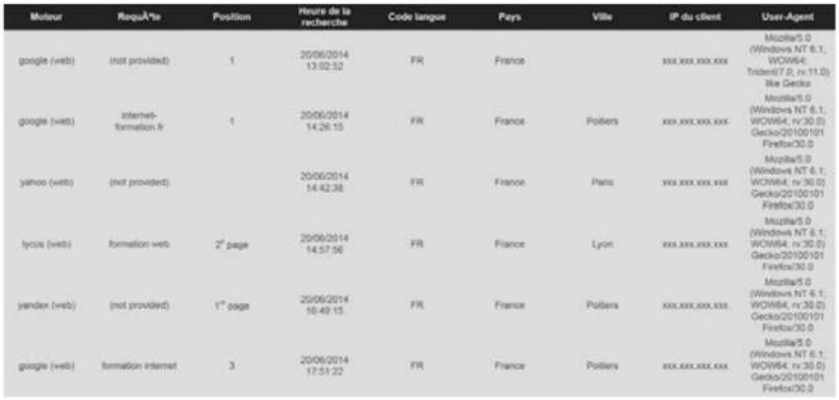
C'est la raison pour laquelle nous allons nous concentrer sur la seconde méthode pour réaliser une fonction PHP qui générera un rapport à la fois en CSV mais aussi en HTML afin d'obtenir des informations essentielles. Nous verrons néanmoins que la méthode ne permet pas de capter tous les renseignements souhaités à cause des problèmes causés par le protocole SSL et les URL de référence.

Google Analytics et le suivi du positionnement

Sachez que Google Analytics permet aussi de d'obtenir la position des résultats de Google à l'aide d'un filtre personnalisé. Nous présenterons cette fonctionnalité dans la section « Google Analytics et ses secrets », entièrement dédiée à l'outil de suivi des statistiques proposé par Google.

Figure 4-29

Rapport de positionnement généré par PHP



Moteur	RequÃªte	Position	Heure de la recherche	Code langue	Pays	Ville	IP du client	User-Agent
google (web)	(not provided)	1	20/06/2014 13:02:52	FR	France		xxx.xxx.xxx.xxx	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
google (web)	internet-formation.fr	1	20/06/2014 14:26:15	FR	France	Poitiers	xxx.xxx.xxx.xxx	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0
yahoo (web)	(not provided)		20/06/2014 14:42:38	FR	France	Paris	xxx.xxx.xxx.xxx	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0
lycos (web)	formation web	2e page	20/06/2014 14:57:56	FR	France	Lyon	xxx.xxx.xxx.xxx	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0
yandex (web)	(not provided)	1er page	20/06/2014 16:49:15	FR	France	Poitiers	xxx.xxx.xxx.xxx	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0
google (web)	formation internet	3	20/06/2014 17:51:22	FR	France	Poitiers	xxx.xxx.xxx.xxx	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0

La fonction a été intitulée `statsReferers()` et répond à plusieurs besoins afin d'être la plus pratique possible :

- récupérer dix types de données (ville d'origine de la recherche, requête du visiteur, position déterminée en fonction des moteurs de recherche, etc.). Ainsi, c'est tout un environnement statistique qui s'ouvre à nous, et encore, nous pourrions agrémenter la fonction pour avoir plus de données à exploiter ;
- s'adapter à près d'une dizaine d'infrastructures différentes pour examiner les adresses de référence et recueillir les informations pertinentes. Cette étape demande un savoir-faire accru du code mais surtout une connaissance parfaite de la construction des URL referers pour chaque moteur de recherche. Grâce à cette analyse fine, la fonction s'adapte aux différents moteurs implantés pour faire ressortir les statistiques le plus précisément possible (en espérant que ces URL ne changent pas souvent…).

Trêve de bavardage, voici la fonction complète pour effectuer notre suivi personnalisé sur plusieurs moteurs :

```
<?php
function statsReferers($logs = 'logs/') {
// Liste des robots et nom du répertoire
$moteurs = array("ask", "yahoo", "baidu", "exalead", "aol", "gigablast", "google", "bing",
                 "voila", "orange", "qwant", "yandex", "lycos", "mozbot", "seek",
                 "duckduckgo", "kelseek", "dazoo", "lemoteur");

// Récupération du referer décodé
$referer = urldecode($_SERVER['HTTP_REFERER']);

// Récupération du nom du moteur
preg_match("#https?://([a-zA-Z.]+\.)?([^\.][a-zA-Z0-9-]+).[a-zA-Z]+/#iU", $referer, $se);
$moteur = strtolower($se[2]);

if(in_array($moteur, $moteurs) && !empty($moteurs)) {
    // Récupération de l'URL de destination (page en cours)
    $url = $_SERVER['SERVER_NAME'].$_SERVER['REQUEST_URI'];

    // Récupération de l'adresse IP du client
    $ip = $_SERVER['REMOTE_ADDR'];
```

```
// Récupération de l'heure (+2h GMT pour la France)
$timestampDate = $_SERVER['REQUEST_TIME'];
$time = date("d/m/Y H:i:s", $timestampDate);

// Récupération de la ville d'où provient la recherche
$ville = $_SERVER['GEOIP_CITY'];

// Récupération du pays d'où provient la recherche
$pays = $_SERVER['GEOIP_COUNTRY_NAME'];

// Récupération de la langue de recherche
$codeLang = $_SERVER['GEOIP_COUNTRY_CODE'];

// Récupération de la requête tapée (paramètre 'q')
// Possibilité d'utiliser parse_str() aussi pour couper...
preg_match("#(q|query|text|kw)=(.*)(&|$)#iU", $referer, $query);
if(!empty($query[2]) && $query[2] != "=") {
    $q = $query[2];
} else {
    $q = "(not provided)";
}

// Récupération de la source (paramètre 'source')
preg_match("#(bvh|source)=(.*)(&|$)#iU", $referer, $source);
if(!empty($source[2])) {
    $src = " (".$source[2].")";
}
preg_match("#(images?|pictures|web|imgurl|img)[/:;=_&]#iU", $referer, $source);
if(!empty($source[1]) && $source[1] != "web") {
    $src = " (images)";
} else {
    $src = " (web)";
}

// Récupération de la position dans les SERP (paramètre 'cd')
if($moteur == "google") {
    preg_match("#(cd|page)=([0-9]+)[^0-9]#iU", $referer, $position);
    if(!empty($position[2])) {
        if(!empty($position[1]) && $position[1] == "cd") {
            $pos = $position[2];
        } else {
            $pos = $position[2]."e page";
        }
    }
}
if($moteur == "bing") {
    preg_match("#first=([0-9]+)[^0-9]#iU", $referer, $position);
    if($src == " (web)") {
        if(!empty($position[1])) {
```

```
            $pos = ceil($position[1] / 10)."e page";
        } else {
            $pos = "1re page";
        }
    }
}
if($moteur == "yandex" || $moteur == "ask" || $moteur == "aol" || $moteur == "lycos" ||
$moteur == "baidu") {
    preg_match("#p(1|n|age|os)?=([0-9]+)([^0-9]|$)#iU", $referer, $position);
    if($position[2] != '') {
        if($moteur == "yandex") {
            $position[2] = $position[2]+1;
        }
        if(!empty($position[1]) && $position[1] == "os") {
            $pos = ($position[2])."e image";
        } else {
            $pos = ($position[2])."e page";
        }
    } else {
        if($moteur != "lycos") {
            $pos = "1re page";
        }
    }
}

// Récupération du nom du navigateur
$userAgent = $_SERVER['HTTP_USER_AGENT'];

// Exportation des données
$entetes = array("Moteur", "Requete", "Position", "Date", "URL", "Code langue", "Pays",
"Ville", "IP du client", "User-Agent");
   $donnees = array($moteur.$src, $q, $pos, $time, $url, $codeLang, $pays, $ville, $ip,
   $userAgent);

// Création du journal si inexistant
if(!is_dir($logs)) {
    mkdir($logs, 0705);
}
// Création et remplissage du fichier CSV
$file = 'positionnement';
$fichier = fopen($logs.$file.".csv", 'a');
$urlCSV = urlencode($protocole.$_SERVER['HTTP_HOST']);
$content = file_get_contents("./".$logs.$file.".csv");
if(empty($content)) {
    fputcsv($fichier, $entetes, ";");
}
fputcsv($fichier, $donnees, ";");
fclose($fichier);
```

```
    // Affichage final HTML
    $result = "<table cellpadding='0' cellspacing='0'
    style='font-family:arial,tahoma,sans-serif'>\n";
    if(!file_exists($logs.$file.".html")) {
    $result.= "<tr align='center' style='background:#222; color:#eee;'>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Moteur</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Requête</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Position</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Heure de la recherche</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>URL</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Code langue</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Pays</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>Ville</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>IP du client</th>\n";
    $result.= "\t<th style='padding:.2em .5em; width:10%'>User-Agent</th>\n";
    $result.= "</tr>\n";
    }
    $result.= "<tr align='center' style='background:#ddd; color:#555;'>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$moteur.$src."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$q."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$pos."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$time."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$url."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$codeLang."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$pays."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$ville."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$ip."</td>\n";
    $result.= "\t<td style='padding:.2em .5em; width:10%'>".$userAgent."</td>\n";
    $result.= "</tr>\n";
    $result.= "</table>\n";

    // Création et remplissage d'un fichier HTML présentable
    $HTML = fopen($logs.$file.".html", 'a');
    fputs($HTML, $result);
    fclose($HTML);

    // Permet d'afficher une URL de retour si besoin
    if(isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == "on") {
        $protocole = "https://";
    } else {
        $protocole = "http://";
    }
    $urlBack = $protocole.$_SERVER['HTTP_HOST']."/".$logs.$file.".html";
    return '<a href="'.$urlBack.'" target="_blank">Suivi de positionnement</a>';
}
}
?>
```

Ensuite, il suffit de lancer la fonction dans une zone répétée d'un site web pour que la fonction génère des fichiers CSV et HTML remplis au fur et à mesure des nouvelles recherches. Il existe deux possibilités d'usage comme le montre le code commenté suivant :

```
<?php
// Lance uniquement la fonction d'analyse
statsReferers();
// Lance la fonction et affiche un lien vers le rapport HTML
echo statsReferers();
?>
```

Cette fonction utilisée en complément d'autres outils peut être très efficace pour mesurer l'impact et la réussite du référencement. Par exemple, vous pouvez grâce au fichier CSV classer les adresses IP des clients et savoir si une même personne est revenue sur votre site et si oui, après combien de temps ou à quelle fréquence. Les possibilités sont donc nombreuses pour tirer profit de ce code…

Suivre les backlinks avec des outils

Le suivi du positionnement ne se limite pas au simple suivi du classement des pages en fonction des requêtes de recherche. Il est également primordial d'examiner le nombre de liens entrants qui touchent nos pages web afin de mesurer régulièrement leurs PageRank et BrowseRank notamment.

Des outils plus ou moins efficaces…

Ici, nulle question de coder en PHP, il existe suffisamment d'outils performants pour suivre ces données. Quoi qu'il en soit, la seule méthode pour obtenir un suivi approfondi serait de créer un robot complet et autonome qui viendrait capter les informations sur un maximum de pages possibles. Nous pouvons imaginer la quantité de ressources que le serveur doit déployer pour récolter et faire fonctionner ce système, il faudrait donc une infrastructure puissante pour réaliser ce type de procédé efficacement.

Heureusement, des développeurs ont pensé à nous et ont créé des outils plutôt performants pour suivre les backlinks. Leur principal inconvénient est de fournir des nombres de backlinks irréalistes, voire souvent loin de la réalité. En effet, ce problème est logique et découle de ce qui a été dit précédemment. Il faut un robot très puissant pour récupérer ce type de données et des ressources maximales pour crawler un maximum de pages. Comme certains outils doivent être limités sur ces points précis, leur index de sites web est trop restreint pour fournir des résultats idéaux…

Voici une liste d'outils plus ou moins performants pour suivre les backlinks relatifs à nos sites web :

- Majestic SEO : http://www.majesticseo.com ;
- Ahrefs : https://ahrefs.com ;
- Open Site Explorer : http://www.opensiteexplorer.org ;
- Backlink Watch : http://www.backlinkwatch.com ;
- Analyze Backlinks : http://www.analyzebacklinks.com ;

- Link Diagnosis : http://www.linkdiagnosis.com ;
- Outil de Ranks.fr : http://www.ranks.fr/fr/outil-backlinks ;
- Advanced Link Manager : http://advancedlinkmanager.com.

Pour la plupart d'entre eux, il suffit de rentrer l'URL d'une page web et d'attendre que l'outil lance l'examen des liens entrants pour obtenir des résultats. Généralement, l'analyse peut prendre un peu de temps mais certains outils sont vraiment d'une très grande qualité. Par exemple, Ahrefs est rapide et propose d'assez bons résultats dans une interface très agréable et complète.

Figure 4-30

Rapport sur les backlinks avec Ahrefs

Majestic SEO est plutôt bien conçu également et fournit un nombre de backlinks plus proche de la réalité, bien que son interface soit moins intuitive et ergonomique que celle d'Ahrefs.

Enfin, comparons ces résultats avec l'outil Open Site Explorer pour avoir une idée de la qualité des robots et des informations fournies par les différents services en ligne.

Si nous regardons de près sur les diverses captures d'écran, nous remarquons qu'aucun des trois outils ne fournit les mêmes données, et l'écart est même parfois énorme entre les résultats. Cela montre à quel point il est difficile de suivre les liens entrants sans avoir de programmes puissants et des ressources importantes.

Globalement, l'idéal est d'utiliser les outils proposés par les moteurs de recherche équipés d'interfaces pour webmasters pour suivre les liens entrants. Leurs bases de données sont beaucoup plus complètes et pertinentes que les outils précédents. Qui plus est, il faut surtout se dire que les moteurs se fient d'abord à eux-mêmes et donc à leurs propres chiffres pour calculer le *ranking* des pages.

Figure 4-31

Suivi des backlinks avec Majestic SEO

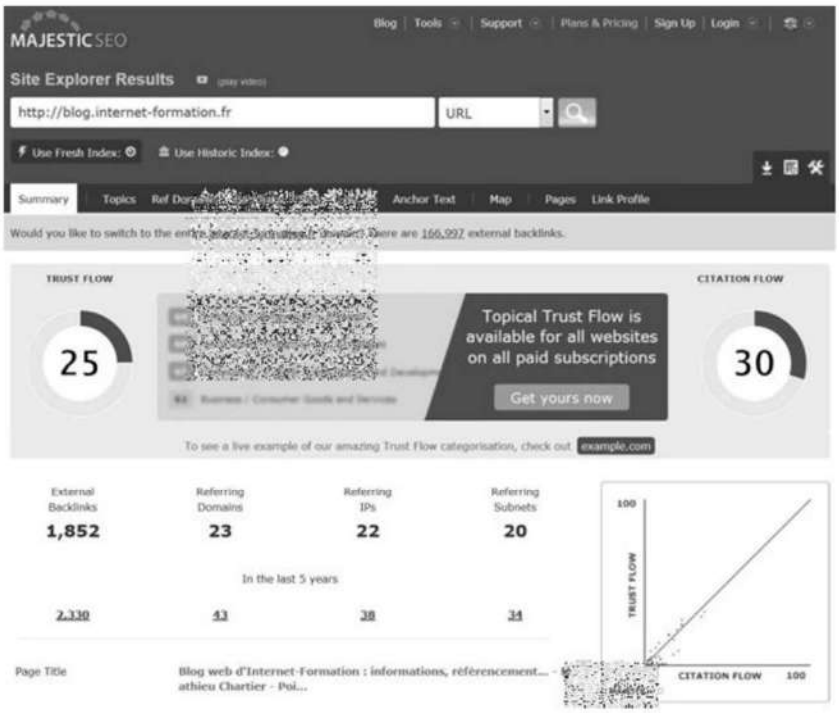
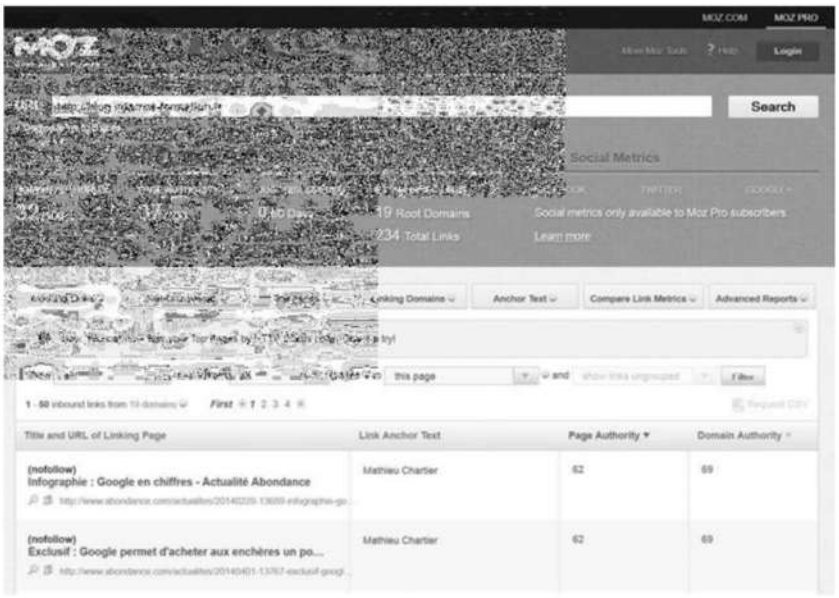


Figure 4-32

Nombre de backlinks calculé par Open Site Explorer



Suivre les liens entrants avec les outils Webmaster Tools

Ces outils proposent pour la plupart une partie consacrée au suivi des liens internes et externes afin d'obtenir les chiffres les plus pertinents possibles.

Contrairement à certains services en ligne présentés précédemment, les interfaces des moteurs de recherche ne mettent pas en avant le profil complet des liens. Il n'est donc pas possible de savoir le nombre de liens en follow ou nofollow, ou encore de savoir dans quelles zones des pages se trouvent les liens, etc.

Toutefois, les nombres de backlinks affichés sont de loin les plus crédibles de tous les outils, et même s'ils diffèrent d'un moteur à un autre, les chiffres sont souvent bien plus pertinents que ceux fournis par les logiciels gratuits en ligne.

Dans la Google Search Console, cliquez sur *Trafic de recherche>Liens vers votre site* ou *Liens internes* en fonction des besoins. Nous obtenons un rapide aperçu du nombre de liens entrants par page mais aussi des domaines qui fournissent le plus de liens retour.

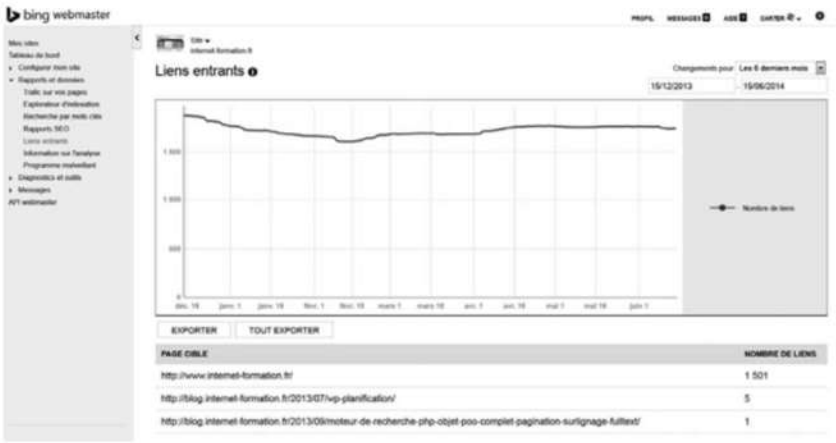
Figure 4-33

Suivi des backlinks dans la Google Search Console

Si Yandex n'est pas le plus pertinent pour des sites français, la Toolbox de Bing est en revanche de très bonne qualité et fournit également de bons résultats. Il suffit de sélectionner le menu *Rapports et données>Liens entrants* pour obtenir un suivi des backlinks. Les résultats sont moins précis que sur Google mais cela s'explique en partie par la qualité de l'index de la firme de Mountain View qui est bien plus élaborée et complète que celle de Microsoft.

Figure 4-34

Suivi des backlinks dans la Toolbox de Bing



Une fonction PHP simple pour Google

Sur le même principe que la fonction PHP que nous avons créée pour suivre les pages indexées, il est possible de construire un code pour faciliter le travail effectué par la commande `link:` de Google.

Cependant, retenez que cette commande donne des résultats très loin de la réalité en termes de backlinks. Cette fonction fait donc plus office de gadget que d'outil fiable :

```
$domaine = "www.domaine.ext";
function backlinks($domaine = '') {
    if(!empty($domaine)) {
    // URL de l'API Google
    $url = "http://ajax.googleapis.com/ajax/services/search/web?v=2.0&q=";
    $url.= "link:".$domaine&filter=0";

    // Lancement de cURL pour récupérer les données de l'API
    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
    $contenuJSON = curl_exec($curl);
    curl_close($curl);

    // Récupération du résultat
    $resultat = json_decode($contenuJSON, true);
    if($resultat['responseStatus'] == 200) {
        return $resultat['responseData']['cursor']['resultCount'];
    }
    }
}
echo backlinks($domaine);
```

Google Analytics et ses secrets

Présentation et usage de l'outil

Installer un code de suivi

Google Analytics est certainement la solution de suivi des statistiques la plus exploitée au monde. Sa gratuité et son interface intuitive en ont fait une référence absolue en matière d'analyse d'audience et de trafic sur les sites web. Certes, il existe des solutions concurrentes de qualité telles que Xiti ou Piwik notamment mais les parts de marché mondiales étant largement dominantes pour Google Analytics, nous avons logiquement porté notre attention sur ce service.

Ne perdons pas de temps à parler de l'histoire de l'outil, entrons plutôt dans le vif du sujet et voyons comment installer Google Analytics. Pour commencer, vous devez disposer d'un compte Google et enregistrer votre site. Ensuite, il suffira de copier un script fourni par l'outil dans une zone répétée du site.

La force de Google Analytics pour le SEO

Il faut savoir que Google Analytics présente des données fournies et accumulées par le moteur de recherche au cours de son crawl, ce qui confère à l'outil un avantage sur certains de ses concurrents. En effet, Google a la chance d'avoir une masse d'informations à disposition pour affiner ses statistiques et proposer un suivi relativement précis des données, ce qui n'est pas le cas de toutes les solutions, bien que les suivis soient de qualité.

Voici à quoi ressemble le code de suivi initial (les `xxx` représentent le numéro de compte et le nom de domaine) :

```
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
  })(window,document,'script','//www.google-analytics.com/analytics.js','ga');

  ga('create', 'UA-xxxxxxx-x', 'xxx.xxxxxxxx.xxx');
  ga('send', 'pageview');
</script>
```

Où placer le code de Google Analytics ?

Il est conseillé d'insérer le code avant la fermeture de la balise `</body>` mais également d'utiliser la méthode asynchrone pour favoriser le PageSpeed. Dans le même but, il est préférable de copier le code dans un fichier JavaScript externe et de ne faire qu'un appel au sein des pages web.

Dès que le code de suivi est installé sur le site, il faut attendre quelques heures pour obtenir les premières données voire plusieurs jours pour commencer à suivre des valeurs plus pertinentes. Si vous avez des doutes sur l'insertion correcte du code sur votre site, vous pouvez tester ce dernier avec l'outil d'Ebrandz (source : http://www.ebrandz.com/analyticstool/) qui vérifiera page par page si le code est présent.

L'alternative Google Tag Manager

Sachez qu'il existe une alternative au code de suivi classique de Google Analytics. En effet, le service Google Tag Manager est en place depuis octobre 2012 et se situe en quelque sorte dans une surcouche des outils disponibles via la firme de Mountain View. Il s'agit d'un gestionnaire de tags qui peut considérablement vous faire gagner du temps si vous effectuez divers suivis parallèlement.

Le principal inconvénient des solutions de Google tient dans le fait que chacune impose son propre code de suivi et ses paramètres personnalisés. Avec Google Tag Manager, ce problème est résolu puisqu'il ne reste plus qu'un code de suivi unique. Il suffit ensuite de gérer dans l'interface les tags (balises en français) pour spécifier quel service nous souhaitons utiliser et pour quelle raison.

Google Tag Manager (GTM) est un véritable couteau suisse qui permet d'économiser beaucoup de temps de développement et d'énergie, et surtout, il évite de modifier trop souvent le code de suivi au risque de le délabrer peu à peu. Désormais, il suffit d'installer un code unique et ensuite, tout se gère à partir de l'interface de Google.

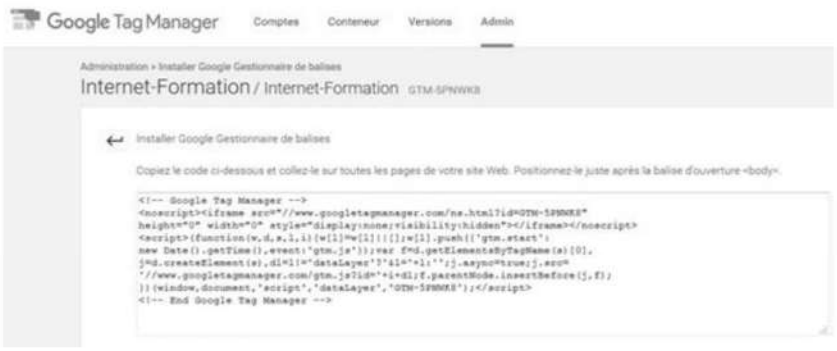
Pour installer Google Analytics via Google Tag Manager, voici comment procéder (source : https://goo.gl/WVy9fk).

1. Rendez-vous à l'adresse https://tagmanager.google.com, créez votre compte si ce n'est déjà fait. Cliquez sur *Pages web* et saisissez un ou plusieurs domaines.
2. Recopiez et installez le code de suivi universel dans le bas de vos pages web ou utilisez un plug-in comme WP Google Tag Manager pour WordPress qui pourra le faire à votre place (source : https://wordpress.org/plugins/wp-google-tag-manager/). Si vous cherchez le code de l'outil, il se trouve dans l'onglet *Admin* puis dans la sous-section *Installer Google gestionnaire de balises*.

```
<!-- Google Tag Manager -->
<noscript><iframe src="//www.googletagmanager.com/ns.html?id=GTM-5PNWK8"
height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'//www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-5PNWK8');</script>
<!-- End Google Tag Manager -->
```

Figure 4-35

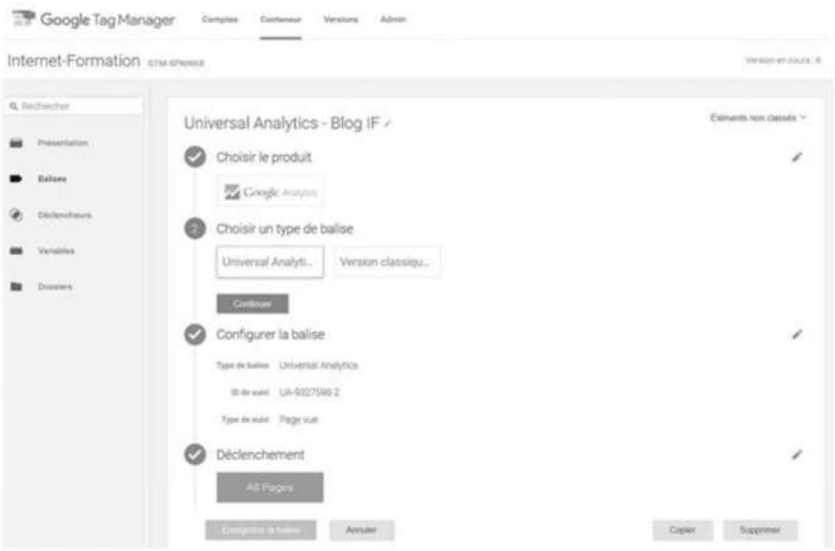
Retrouver le code de suivi de Google Tag Manager



3. Installez les balises pour les services que vous souhaitez suivre. Ici, nous installons Google Universal Analytics avec la création d'une balise simple (recommandé plutôt que Analytics classique). Il suffit de donner un nom à la balise, de choisir son type, d'indiquer l'ID du compte Google Analytics à suivre, puis de fixer les règles de déclenchement (l'option *Toutes les pages* est conseillée).

Figure 4-36

Création d'un tag Google Universal Analytics



4. Ajoutez ou non des règles, des macros ou des conditions pour personnaliser et affiner les règles de suivi.

L'autre avantage de Google Tag Manager est de proposer une liste assez conséquente de variantes de balises pour suivre tous types de données. Cela peut passer par un suivi remarketing SEA via Google AdWords, un suivi des conversions en référencement payant, un écouteur d'événements lors d'un clic sur un lien, dans une page ou pour le remplissage de formulaires. Enfin, il est même possible d'ajouter des balises HTML, à savoir des tags provenant d'outils et solutions concurrentes à Google afin de faciliter le suivi au sein d'une seule et même interface.

Nul doute que les habitués auront du mal à retourner à leur ancien système de gestion mais ici, nous conserverons la méthode classique pour ne pas se perdre entre les deux types de systèmes…

Bien comprendre le système de Google Tag Manager

Il faut prendre un peu de temps pour lire la documentation et s'imprégner de l'usage de l'outil mais une fois habitué, il s'agit d'une alternative qui ravira nombre d'entreprises et de référenceurs tant le suivi est facilité par ce biais. L'interface manque encore un peu de clarté par moment mais reste de très bonne facture et plutôt intuitive, le mieux est de l'essayer quelques jours pour se faire une idée…

Google Analytics et SEO

Pour effectuer un suivi SEO et webmarketing, Google Analytics est la solution miracle car l'outil dispose de presque toutes les données nécessaires pour obtenir des chiffres à la hauteur de nos attentes.

L'interface change fréquemment et il est possible que les captures d'écran présentées par la suite soient différentes de celles que vous pouvez voir affichées. Rien de bien alarmant car les intitulés ne sont en général pas modifiés. Par défaut, le tableau de bord initial affiche les statistiques des trente derniers jours avec les informations principales telles que le nombre de sessions (ou visites), de pages vues, d'utilisateurs (anciennement « visiteurs uniques »), de pages/session, la durée moyenne des visites et le taux de rebond moyen sur l'ensemble du site.

Un menu situé à gauche permet de parcourir rapidement les catégories d'informations que nous souhaitons suivre. Il existe donc plusieurs sections :

- Temps réel pour un suivi en direct ;
- Audience pour les données générales ;
- Acquisition pour le suivi des données d'AdWords, des réseaux sociaux mais aussi du référencement naturel, etc. ;
- Comportement pour analyser la qualité du site, les pages visitées et de référence mais aussi la vitesse du site, etc. ;
- Conversions pour les boutiques en ligne qui ont mis en place des suivis avancés ou pour les sites qui utilisent des entonnoirs de conversion.

En termes de référencement, plusieurs données sont susceptibles de nous intéresser, au-delà des données classiques sur le trafic global. Par exemple, il est intéressant de suivre les URL referers (*Acquisition>Tous les sites référents*) afin de déterminer les sources dominantes de trafic). Selon le type de site et de communication, il n'est pas rare de ne pas trouver Google en tête.

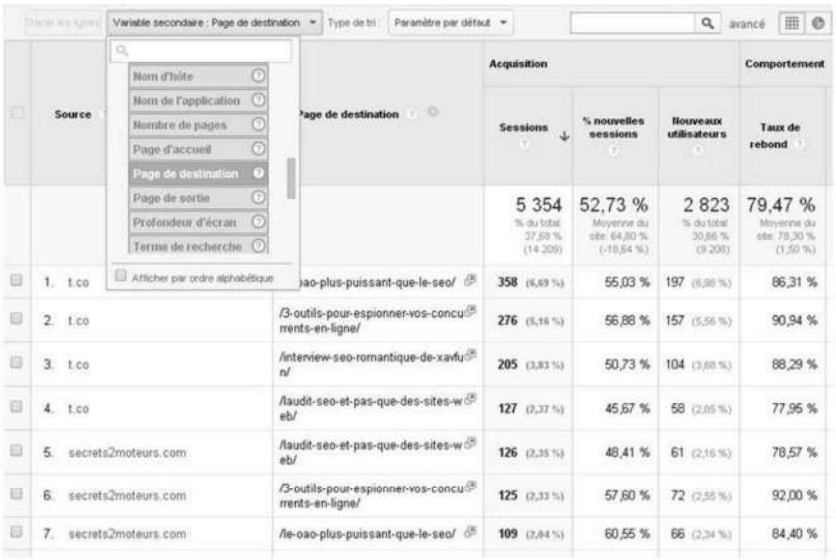
Sur la figure 4-36, nous voyons que Twitter (t.co) est en tête de liste. Nous avons également ajouté une variable secondaire pour affiner les données en ajoutant la page de destination. Ainsi, nous pouvons savoir combien de visiteurs sont venus sur notre site, d'où ils proviennent mais surtout vers quelles pages ils se sont dirigés.

Toujours dans le menu *Acquisition*, vous pouvez cliquer sur Tout le trafic pour voir quelles sont les sources les plus efficaces. Aussi, nous verrons un mélange de plusieurs moteurs de recherche et de sites référents divers selon notre mode de communication. S'il s'agit d'un site de présentation, il est fort probable que les sources soient essentiellement en provenance de Google mais si vous administrez une boutique en ligne ou un blog, elles risquent d'être multiples

D'autres données sont intéressantes pour le suivi SEO. Nous pouvons sélectionner le menu *Comportement>Contenu du site>Pages de sortie* ou *Pages de destination* par exemple afin de connaître respectivement les pages qui ont généré le plus de « fuites » et celles qui ont reçu le plus de visites en premier lieu.

Figure 4-37

Suivi des URL referers et visites par page de destination



Le menu *Acquisition>Réseaux sociaux* permet d'avoir une vue d'ensemble ou plus précise des médias communautaires qui génèrent le plus de trafic sur votre site web. Cliquez ensuite sur *Réseaux sociaux référents* afin de comparer la courbe du trafic général avec celle des apports des réseaux sociaux. Comme pour les URL referers, il peut être intéressant d'ajouter la variable secondaire *Pages de destination* pour savoir quels contenus ont généré le plus de visites selon les réseaux sociaux.

Figure 4-38

Suivi des réseaux sociaux dans Google Analytics

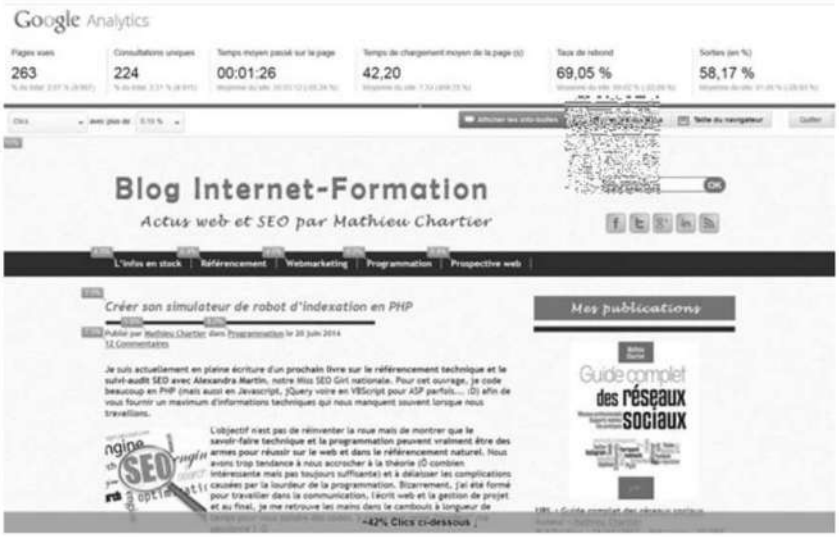
Le menu *Acquisition>Mots-clés* est l'une des fonctionnalités préférées des référenceurs car elle permet d'afficher les expressions tapées par les visiteurs dans les moteurs de recherche. Malheureusement,

il est de mise d'obtenir un résultat not provided qui fâche à juste titre nombre de marketeurs et qui ne permet plus d'obtenir de résultats probants.

Le menu *Comportement>Vitesse du site>Temps de chargement* est pertinent pour suivre l'influence de la vitesse du site sur les visites (et en quelque sorte sur le PageSpeed, même si le critère n'est pas basé sur la vitesse à proprement parler). Enfin, le menu *Comportement>Analyse des pages web* est très intéressant si vous auditez votre site car elle met en exergue les zones les plus propices au clic sur votre site web.

Figure 4-39

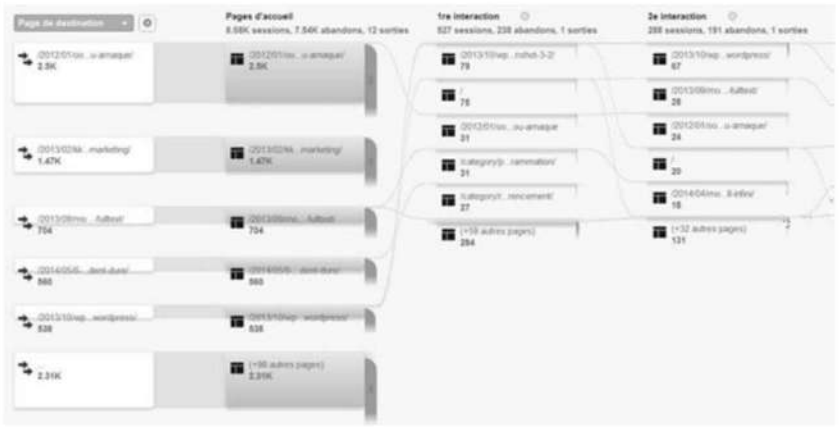
Analyse des pages avec pourcentage des zones les plus cliquées en fonction de l'écran ou des couleurs...



Terminons notre rapide tour d'horizon des possibilités de Google Analytics par le menu *Comportement>Flux de comportement* qui permet d'analyser en détail les principaux parcours de navigation réalisés par les internautes en fonction de leur page de destination, de leur referers ou d'un autre critère de départ.

Figure 4-40

Flux de comportement pour suivre les scenarii de navigation des visiteurs dans un site web



Ainsi, nous pouvons optimiser certaines parties du site si nous sentons des pertes importantes. Le blog présenté dans les captures subit une perte radicale de visiteurs une fois que la page de l'article a été lue. Cela peut s'expliquer par un manque d'incitation au clic vers un autre article, par un contenu jugé peu intéressant par les visiteurs ou simplement par une satisfaction suffisante au point que les usagers ont obtenu l'information et reparte voguer sur le Web.

Méthodes de tracking

KPI et objectifs

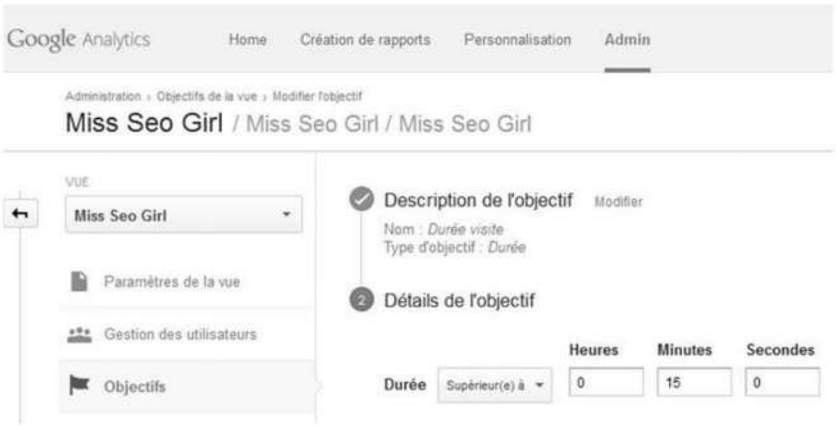
Le KPI (*Key Performance Indicator* ou Indicateur clé de performance) est un indicateur de suivi de l'efficacité des pages web. Il permet de mesurer la qualité des pages en fonction des objectifs définis au préalable.

Dans un contexte webmarketing et de suivi d'audience de sites web, les KPI peuvent être par exemple : le temps passé sur les pages, le nombre de visiteurs, de téléchargements d'un fichier (PDF ou autres), d'inscriptions à une newsletter, de partages sur les réseaux sociaux, le CA généré ou encore le taux de conversions... Les idées ne manquent pas et selon ce que vous souhaitez suivre comme données selon vos objectifs, les KPI sont faits pour cela.

Google Analytics permet de mettre en place des objectifs (donc des KPI) afin de suivre s'ils ont été atteints ou pas. Pour cela, cliquez sur le lien Admin en haut de la page, puis sélectionnez Objectifs.

Figure 4-41

Création d'un objectif de suivi dans Google Analytics



Pour visualiser les résultats, retournez dans le tableau de bord et sélectionnez le menu *Conversions>Objectifs>Vue d'ensemble.*

Figure 4-42

Suivi des KPI personnalisés avec données chiffrées

Du point de vue du référencement, certains objectifs peuvent être pertinents à mettre en place pour affiner les résultats classiques et mesurer des points précis comme le nombre de téléchargements générés par les visites ou encore le nombre de clics sur un lien. Pour ce faire, il faut se référer aux méthodes de tracking que nous allons présenter de suite car certains facteurs fonctionnent avec des « capteurs » installés dans les codes de suivi…

Variables _utm et codes personnalisés

Pour réussir à optimiser et personnaliser Google Analytics, il faut absolument maîtriser les variables `_utm` placées dans les codes de suivi, c'est-à-dire les types de cookies déposés sur l'ordinateur des visiteurs pour suivre leurs actions et récolter des informations.

Pour en savoir plus sur Google Analytics…

Google Analytics est un outil vaste et terriblement complexe lorsque nous entrons dans le vif du sujet. Il est recommandé de procéder à d'autres lectures sur le sujet pour aller plus loin que la présentation que nous ferons dans cet ouvrage, ne soyez donc pas surpris de constater des manques dans notre propos.

Il existe cinq principaux types de variables `_utm` à connaître :

- `_utma` : cookie d'une durée de vie de deux ans qui stocke entre autres les informations de domaine, d'identifiant unique de visiteur et le nombre de visites ;
- `_utmb` : cookie de visite (ou session) avec une durée de vie de 30 minutes ;
- `_utmc` : autre cookie de visite (et campagne) mais ce dernier se met à jour à chaque nouvelle visite pour récupérer les nouveaux visiteurs et d'autres informations utiles ;
- `_utmz` : cookie de conversion dont le but est de stocker les données de provenance des visites telles que les sites référents, les mots-clés utilisés (dans le cas d'une recherche), le support utilisé (naturel, CPC, etc.). Il a une durée de vie de 6 mois ;
- `_utmv` : cookie d'une durée de vie de deux ans, personnalisable, et qui permet de récupérer toutes sortes d'informations.

Le site anglophone cheatography.com a mis en place une liste des variables `_utm` très détaillée qui permet de savoir à quoi correspond chaque cookie déposé par Google Analytics (source : http://goo.gl/a6pjDh).

Il est possible de modifier la durée de vie des cookies en plaçant un code tel que le suivant en précisant la durée souhaitée :

```
_gaq.push(['_setSessionCookieTimeout', 1800000]);
```

Pour bien maîtriser l'art de Google Analytics, il faut parfois rentrer dans la documentation (non traduite) et s'imprégner des diverses formes de codes à mettre en place pour effectuer un tracking avancé et qualifié, notamment entre les méthodes classiques et les codes asynchrones.

Quand on utilise Google Analytics professionnellement, nous recourons fréquemment à des méthodes de tracking, à savoir des suivis entièrement personnalisés sur des actions, des interactions, des transactions ou des événements, par exemple. Il existe diverses techniques de tracking que nous pouvons classer en six catégories :

- suivi de l'e-commerce (transactions et produits) ;
- **suivi d'événements particuliers (clics, téléchargements...) ;**
- **suivi de la vitesse de chargement des pages ;**
- **suivi des réseaux sociaux ;**
- **suivi des moteurs de recherche et des sites référents ;**
- **suivi des navigateurs.**

Nous ne pourrons pas traiter en détail toutes ces méthodes tant les variantes sont nombreuses et car toutes n'auront pas d'intérêt direct pour un suivi de référencement.

Prenons toutefois un exemple pour commencer avec le suivi des navigateurs notamment. Google présente six variables possibles à placer dans les codes de suivi :

- `_setClientInfo()`, `_setDetectFlash()` et `_setDetectTitle()` **pour indiquer à Google Analytics d'enregistrer respectivement les données du navigateur, de Flash et des titres dans les rapports de données (valeurs** `true` **ou** `false`**).**
- `_getClientInfo()`, `_getDetectFlash()` et `_getDetectTitle()` **pour détecter respectivement si ces marqueurs sont activés ou non.**

Dans ce cas, si nous plaçons dans une ou plusieurs pages le code suivant, Google Analytics ne prendra pas en compte les informations liées à Flash Player notamment :

```
_gaq.push('_setDetectFlash', false);
```

Ceci n'est qu'un exemple mais il montre la puissance et l'étendue des possibilités en matière de tracking ou de personnalisation des codes au sein de Google Analytics.

Il existe plusieurs familles de variables propres à chaque type de suivi. Nous trouvons notamment des variables autour des méthodes JavaScript comme `_trackEvent()`, `_trackTrans()`, `_trackSocial()` ou encore `_trackTiming()`. Google Analytics propose également des collectes de données relatives

aux contenus avec `_trackPageview()` ou encore `trackView()`. Autant dire que nous ne manquons pas de choix...

Inutile de rentrer dans les détails car nous allons désormais nous tourner uniquement vers des cas pertinents pour le référencement.

Méthode _trackEvent()

La principale méthode de tracking est axée autour des variables `_trackEvent()` destinées à mener un suivi d'événements particuliers. Il est donc possible de placer ce type de code dans les pages, les liens, les formulaires, etc., qui sont susceptibles de nous intéresser.

Le format des `trackEvent()` est le suivant :

```
_trackEvent(category,action,opt_label,opt_value,opt_noninteraction)
```

- `category` est obligatoire : il s'agit du nom attribué à l'objet que vous voulez suivre (afin de pouvoir retrouver les données dans Google Analytics).
- `action` est également obligatoire : la valeur correspond au type d'action effectuée par l'internaute (Démarrer, Arrêter, Pause, Téléchargement...).
- `opt_label`, `opt_value` et `opt_noninteraction` sont tous facultatifs. Ils permettent respectivement d'attribuer une courte description à l'événement suivi, un numéro de suivi et enfin de permettre le comptage du visiteur suivi dans le taux de rebond.

Prenons un exemple tout simple, si nous voulons savoir combien de fois une vidéo de Mylène Farmer a été téléchargée, le code suivant peut faire l'affaire s'il est placé au bon endroit :

```
_gaq.push(['_trackEvent', 'Videos', 'Téléchargement', 'Mylène Farmer']);
```

Notons toutefois que le code de suivi ne peut pas suffire seul, il faut qu'il soit intégré dans une zone précise du code HTML pour être fonctionnel. Dans le cas d'un téléchargement, le plus simple est de l'appliquer directement au lien de téléchargement, comme ceci :

```
<a href="download.php?file=video-mylene-farmer.mp4" onclick="_gaq.push(['_trackEvent',
'Videos', 'Téléchargement', 'Mylène Farmer']);">Télécharger</a>
```

Ainsi, à chaque clic des internautes sur le lien de téléchargement, une information est transmise dans l'interface de Google Analytics et nous permet de récupérer l'information.

En matière de référencement, le suivi des événements peut être multiple pour nous apporter son lot d'informations sur l'impact des campagnes réalisées. En effet, il suffit dans Google Analytics d'étudier les recherches organiques qui ont abouti par un clic sur un événement de notre choix pour mesurer si les KPI ont été remplis (si notre objectif est de booster un nombre de souscriptions, de téléchargements ou encore de visionnages de médias, par exemple).

Méthode _trackTrans()

Le suivi des transactions dans les boutiques en ligne notamment est un peu plus complexe que le simple suivi d'événements que nous appliquons sur des liens ou des zones cliquables en général. Avec la méthode `_trackTrans()`, il convient de donner plus de précision dans le code de Google Analytics pour suivre la vie du site mais aussi d'un ou plusieurs produits en particulier.

L'usage de `_trackTrans()` ne correspond en fait qu'à la dernière étape, celle qui permet d'envoyer les informations sur les serveurs de Google Analytics. Nous devons préalablement composer les données avec deux fonctions simples : `_addTrans()` pour qualifier la transaction et `_addItem()` pour présenter le produit suivi.

Le code de suivi des transactions doit idéalement être placé dans la page de remerciements ou de confirmation de l'achat afin que les données ne soient pas polluées par des transactions avortées. Il faut généralement recourir à des liaisons avec les bases de données pour remplir les codes agrémentés de cette page cible.

Comment faire si plusieurs produits sont à suivre ?

La méthode `_addItem()` doit être répétée pour chaque produit présent lors de la transaction. Il faut donc prévoir un système de boucle pour répéter le code de manière dynamique.

Voici à quoi ressemble un code de suivi pour les transactions avec les commentaires associés à chaque ligne :

```
<script type="text/JavaScript">
    var _gaq = _gaq || [];
    _gaq.push(['_setAccount', 'UA-XXXXX-X']);
    _gaq.push(['_trackPageview']);

    // Gestion de la transaction
    _gaq.push(['_addTrans',
    '5189',           // Identifiant de transaction (unique)
    'Super boutique', // Nom de la boutique ou de l'affiliation
    '110.99',         // Prix total
    '22.20',          // Taxes (TVA)
    '5',              // Frais de transport
    'Paris',          // Ville
    'Ile-de-France',  // Région
    'France'          // Pays
    ]);

    // Gestion du produit, à appliquer à tous les produits du site
    _gaq.push(['_addItem',
    '5189',           // Identifiant de transaction (unique)
    'SP01',           // Code ou référence du produit
    'Super Produit',  // Nom du produit
    'Top des ventes', // Catégorie
```

```
    '110.99',          // Prix à l'unité
    '1'                // Quantité
    ]);

    // Soumission de l'information vers Google Analytics
    _gaq.push(['_trackTrans']);

(function() {
    var ga = document.createElement('script'); ga.type = 'text/JavaScript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
    '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();
</script>
```

Vérifier l'ordre d'écriture des fonctions de suivi

Il faut absolument veiller à ce que les trois fonctions soient utilisées dans cet ordre précis car le cheminement des informations est primordial.

Le principal problème du suivi des transactions concerne la gestion des prix mais aussi de l'identifiant qui doit être unique pour chaque transaction. Voyons comment procéder…

Les prix s'affichent dans un format anglophone, ce qui signifie que les virgules sont remplacées par des points, les chiffres des milliers ne sont pas séparés par des espaces et enfin, les devises sont omises dans les codes de suivi. Par conséquent, si un produit est affiché au prix de 2 257,89 €, par exemple, il faut écrire 2257.89 uniquement.

Il est très simple de faire ce type de réglages en PHP notamment avec une fonction telle que la suivante :

```
function formatPrix($prix) {
    $prix = str_replace(" ", "", $prix);
    $prix = str_replace("€", "", $prix);
    $prix = str_replace(",", ".", $prix);
    return $prix;
}
```

Le deuxième souci provient de l'identifiant de transaction unique à gérer. Par défaut, il est inexistant et doit donc être généré pour ne pas mélanger les informations. Il existe en réalité de nombreuses méthodes intéressantes dont vous serez seuls juges, en voici des exemples.

- Générer un identifiant unique à partir d'un timestamp (date) précis. L'avantage est d'être assuré d'avoir un numéro unique car deux transactions ne pourront pas avoir lieu à la même seconde techniquement parlant.

```
// Création d'un objet Date() et récupération du timestamp
var dateJour = new Date();
var timestamp = dateJour.getTime();
```

```
// Suite du code…
// Ajout du timestamp dans les variables
pageTracker._addTrans(timestamp,"x","x","x","x","x","x","x");
pageTracker._addItem(timestamp,"x","x","x","x","x");
pageTracker._trackTrans();
```

- Générer un numéro au hasard accompagné d'une information intéressante pour nous faciliter le suivi. Par exemple, il peut s'agir du nom de l'hôte (domaine), etc. Dans ce cas, le code pourrait ressembler au suivant :

```
// Création d'un numéro au hasard
var str = ""+Math.random();
var hasard = str.substr(2, 10);
// Récupération du nom d'hôte
var hote = window.location.hostname;
// Formatage de l'identifiant unique
var IdUnique = hote+'-'+hasard;

// Suite du code…
// Ajout de l'identifiant dans les variables
pageTracker._addTrans(IdUnique,"x","x","x","x","x","x","x");
pageTracker._addItem(IdUnique,"x","x","x","x","x");
pageTracker._trackTrans();
```

L'avantage de la seconde technique est de pouvoir ajouter des filtres dans Google Analytics pour récupérer les données grâce au nom d'hôte (ou d'un autre type d'information). C'est d'ailleurs à cette étape que les référenceurs et marketeurs portent de l'intérêt car nous pouvons mesurer un retour sur investissement en analysant les transactions en fonction des sources initiales ou des URL referers.

Alternative avec Google Tag Manager

Vous pourrez trouver des explications sur la méthode à appliquer pour suivre les transactions et l'e-commerce avec Google Tag Manager sur le site GA-Scripts.org à l'adresse suivante : http://goo.gl/pBHDEv.

Autres tracking avec Google Analytics

Les solutions sont multiples pour mesurer toutes sortes de facteurs avec l'outil de Google mais elles ne sont pas toujours d'une grande aide pour le référencement. Il existe par exemple la méthode `_trackTime()` pour calculer des durées passées dans des pages, des catégories de produits ou encore des types d'articles.

Nous pouvons aussi mentionner la méthode `_trackSocial()` dédiée au suivi des interactions avec les réseaux sociaux. Cette dernière permet notamment d'obtenir le nombre de partages, de Likes, d'abonnés, etc., en fonction des API et des options proposées par chacune des plates-formes communautaires. Si cela vous intéresse, n'hésitez pas à lire la documentation officielle : http://goo.gl/H6urHj.

Enfin, terminons cette succincte partie sur le tracking avec Google Analytics par le suivi du positionnement grâce à un code entièrement personnalisé. Nous verrons plus tard une méthode quasi équivalente avec les filtres classiques du service de Google. La technique est valable mais il faut avouer que la propension à obtenir des not provided dans les résultats de Google Analytics ne va pas toujours nous aider à savoir quel mot est affilié à une place précise dans les SERP.

La méthode a été expliquée et présentée sur le blog cutroni.com en janvier 2013 (source : http://goo.gl/OdwTrG). Il est vrai que le suivi précis des requêtes a quelque peu déserté l'interface depuis deux ans mais gardez en tête le principe :

- ajoutez un code de suivi avec une fonction JavaScript pour capter le positionnement et la requête (presque systématiquement transformée en « not provided ») ;
- laissez Google Analytics calculer le positionnement moyen selon la requête tapée par l'internaute.

Voici le code proposé par le blog cutroni.com, notez les paramètres `cd` et `q` que nous avions déjà évoqués auparavant qui récupèrent le classement de la page et la requête de l'utilisateur.

```
if (document.referrer.match(/google\.com/gi) && document.referrer.match(/cd/gi)) {
    // Récupération de l'URL referer
    var myString = document.referrer;
    // Récupération de la position
    var r = myString.match(/cd=(.*?)&/);
    var rank = parseInt(r[1]);
    // Récupération de la requête
    var kw = myString.match(/q=(.*?)&/);

    // Assure qu'il s'agit bien d'un not provided ou non
    if (kw[1].length > 0) {
        var keyWord  = decodeURI(kw[1]);
    } else {
        keyWord = "(not provided)";
    }
    var p = document.location.pathname;
    _gaq.push(['_trackEvent', 'RankTracker', keyWord, p, rank, true]);
}
```

Une fois ce code mis en place, le menu *RankTracker* est ajouté dans Google Analytics et permet de suivre les mots-clés ainsi que les positionnements associés. Il est même possible d'ajouter l'URL visitée si besoin pour avoir au moins une idée des mots-clés qui auraient pu être tapés (la thématique unique des pages permet d'avoir des indications et réduit le champ des possibles).

Nous disposons maintenant de plusieurs pistes pour utiliser Google Analytics dans notre suivi de référencement et positionnement. Mais grâce à des méthodes de tracking, nous allons voir qu'il existe encore d'autres possibilités de suivi avec l'outil de Google.

Filtres et rapports pour le SEO

Les filtres représentent un moyen simple de ne conserver que les informations qui nous sont utiles dans Google Analytics tout en les affichant de façon personnalisée. En général, ils ne permettent pas de récupérer des données non incluses dans l'outil de Google. Ils ne servent qu'à trier et réorganiser les tableaux de bord.

Nous allons voir quelques filtres simples mais qui peuvent nous être utiles à l'occasion. Il ne s'agit bien entendu que d'exemples qui peuvent être étayés et accompagnés d'autres techniques.

Exclure une adresse IP des statistiques

Si vous souhaitez obtenir un suivi des statistiques le plus réaliste possible, il convient d'exclure les adresses IP qui nous concernent dans les rapports de données, cela évite que nos diverses visites s'accumulent et s'entremêlent avec les données des visiteurs. De plus, cela peut avoir une forte implication sur le taux de rebond, le nombre de pages vues ou encore la durée des visites, ce filtre est donc une nécessité si nous souhaitons obtenir des résultats précis.

Pour procéder à cette exclusion, cliquez sur le lien *Admin* puis sur *Filtres*. Ensuite, il suffit de créer un nouveau filtre en choisissant les options *Exclure, trafic provenant des adresses IP* et *étant égal à*. Saisissez votre adresse IP personnelle (en tant que visiteur, donc pour chaque ordinateur lié à un routeur susceptible d'être utilisé pour se rendre sur le site web).

Différencier les profils pour un meilleur suivi

Ce type de filtre peut aussi être appliqué dans un profil différent afin de suivre à la fois les données générales mais aussi celles qui excluent le trafic des salariés d'une entreprise sur leur site, par exemple. Il peut aussi être primordial de bloquer l'accès à des URL referers ou robots non pertinents pour l'analyse des données.

Figure 4-43

Exclusion d'une adresse IP des statistiques

Sur le même principe, sachez qu'il peut être intéressant d'exclure également les visites provenant de sous-répertoires, de sous-domaines ou d'autres sources qui pourraient nous correspondre. Par exemple, exclure le dossier de l'interface d'administration d'un CMS peut être une sécurité pour s'assurer qu'aucun gestionnaire ne soit comptabilisé dans les statistiques finales (bien qu'une bonne gestion des IP permette d'éviter ce type de problème).

Bloquer le spam referer

Un des principaux problèmes que rencontre Google Analytics est la surcharge de pages référentes *spammy* qui faussent les compteurs de l'outil. De plus en plus de sites web sont touchés par ce *spam referer* polluant. Il convient donc de le filtrer ou de le supprimer.

Nous allons étudier une méthode ici, mais sachez que de nombreux tutoriels plus complets existent sur la Toile, car pour bien bloquer le *spam referer*, il faudrait modifier le fichier `.htaccess` et pas seulement Google Analytics.

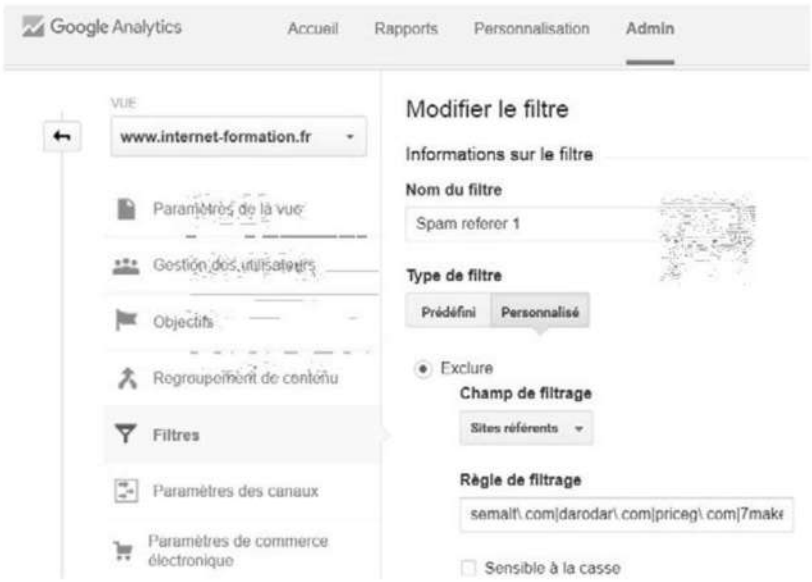
Pour créer un filtre de suppression du *spam referer*, il faut se rendre dans l'onglet *Admin*, puis cliquer sur le bouton *Filtres* dans la colonne située à droite de l'écran. Créez un nouveau filtre en respectant ces règles :

- *Nom du filtre* : indiquez le nom que vous souhaitez, mais si possible, numérotez-le car la masse de *spam referer* vous oblige souvent à créer plusieurs filtres spécifiques ;
- *Type de filtre* : personnalisé (option *Exclure*) ;
- *Champ de filtrage* : choisissez *Sites référents* ;
- *Règles de filtrages* : indiquez les règles à appliquer dans ce champ (limité à 255 signes). Il s'agit d'une expression régulière ; il faut juste échapper les points (en écrivant `\.`) et caractères spéciaux dans les URL et mettre des « ou » (en ajoutant un | entre les URL) pour cumuler les URL à filtrer.

Une fois ces étapes réalisées, validez le formulaire et recommencez l'opération autant de fois que nécessaire pour filtrer tous les *spam referers*. Il nous est impossible de dresser une liste tant ces spams sont nombreux et se multiplient ; n'hésitez pas à analyser ceux qui vous touchent particulièrement via Google Analytics (dans *Sites référents* du rapport d'acquisition).

Figure 4-44

Exclusion du spam referer via des filtres dans Google Analytics



Suivre le positionnement d'un site web

Comme nous l'avons vu au début de ce chapitre, le suivi du positionnement fait partie des éléments les plus importants pour les référenceurs. Qu'on se le dise, Google Analytics ne va pas nous permettre de révolutionner le suivi mais plutôt de capter à la volée le classement des pages dans Google et de l'afficher où bon nous semble. Il s'agit d'un filtre relativement ancien qui a été réadapté peu à peu en fonction des évolutions du service de Google.

Le principe est de récupérer le paramètre `cd` accessible dans les URL de référence des pages de résultats du moteur de recherche. Ce dernier indique le positionnement précis dans les SERP, nous allons donc l'associer aux requêtes de recherche, bien que le problème des not provided casse grandement l'intérêt du suivi.

Figure 4-45

Suivi du positionnement en fonction des requêtes

		4 918 % du total: 76,90 % (6 395)
☐	1. (not provided)	**2 857** (58,09 %)
☐	2. (not provided) (1)	**369** (7,50 %)
☐	3. (not provided) (3)	**280** (5,69 %)
☐	4. (not provided) (2)	**260** (5,29 %)
☐	5. (not provided) (6)	**187** (3,80 %)
☐	6. (not provided) (4)	**169** (3,44 %)
☐	7. (not provided) (7)	**121** (2,46 %)
☐	8. (not provided) (5)	**80** (1,63 %)
☐	9. (not provided) (9)	**69** (1,40 %)
☐	10. (not provided) (8)	**52** (1,06 %)
☐	11. (not provided) (10)	**39** (0,79 %)
☐	12. kko store	**19** (0,39 %)
☐	13. kko store (3)	**18** (0,37 %)
☐	14. oopad (6)	**17** (0,35 %)
☐	15. (not provided) (13)	**11** (0,22 %)

Nous allons devoir créer deux filtres distincts placés l'un après l'autre dans l'interface de gestion des filtres. Le premier va nous permettre d'extraire les positions à partir des URL de référence et le second va nous donner l'occasion de placer le résultat dans la zone qui nous intéresse.

Après quelques jours, le filtre aura eu le temps d'être appliqué à plusieurs requêtes et nous pourrons apercevoir dans le suivi des mots-clés la requête tapée suivie du positionnement dans les SERP inscrit entre parenthèses. Le filtre devrait donner de bons résultats après plusieurs jours ou semaines d'application. Sur la figure 4-45, vous pouvez constater le nombre quasi exclusif de not provided qui perturbe malheureusement l'intérêt de ce suivi.

Créez le premier filtre avancé selon les paramètres indiqués sur la figure 4-44 en respectant scrupuleusement la syntaxe de l'expression régulière à la mode Google Analytics. Ici, nous récupérons les données du troisième bloc entre parenthèses du *champ A*, c'est pourquoi l'indice de sortie est `$A3`, par exemple.

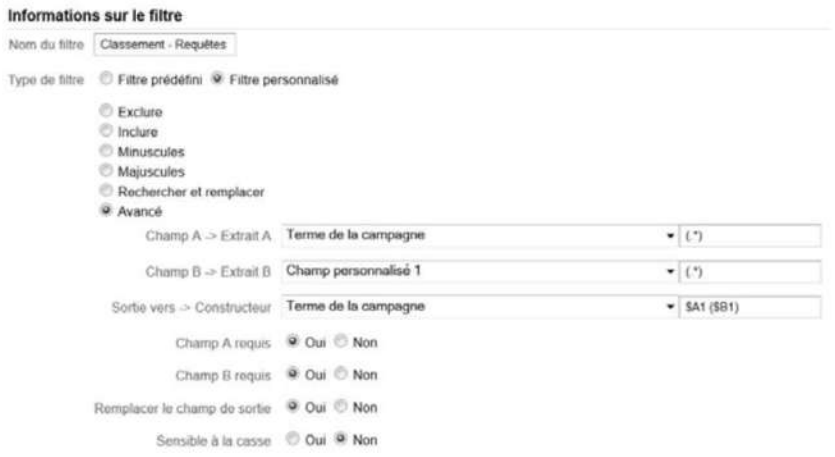
Figure 4-46

Récupération des positions avec un regex

Ensuite, créez un second filtre pour placer l'information récupérée dans la zone qui vous semble appropriée. Dans notre exemple, nous appliquons le filtre aux requêtes de recherche, donc à la catégorie *Termes de la campagne*. Nous affichons d'abord les mots-clés issus du *champ A* avec `$A1`, puis le positionnement récupéré dans un champ personnalisé avec `$B1`, tout simplement…

Figure 4-47

Ajout du positionnement à côté des requêtes



Filtrer les sites multilingues

Les sites multilingues sont généralement composés de répertoires ou de sous-domaines pour chaque langue installée. Il peut donc être intéressant d'appliquer des filtres pour les répertoires ou les sous-domaines afin de dissocier nettement les statistiques en fonction des langues.

L'avantage de ces deux filtres simples est de pouvoir mesurer les données selon les pays mais surtout l'intérêt du site à proposer diverses versions linguistiques. Parfois, les chiffres présentés peuvent donner le tournis tant certaines langues sont peu usitées sur nos sites web, mais cela fait partie du jeu… Il existe plusieurs possibilités pour créer ce type de filtre : si vous n'avez qu'un seul répertoire à suivre en particulier, créez un nouveau filtre prédéfini avec les options *Inclure* uniquement et *Trafic vers les sous-répertoires*. En revanche, si vous en avez plusieurs à suivre, vous pouvez utiliser une expression régulière en créant un filtre personnalisé sur l'URI de la demande. Le regex ressemblera au code suivant :

```
^/(repertoire1\repertoire2)/$|^/( repertoire1\repertoire2)
```

Figure 4-48

Gestion des répertoires multilingues

Alternative pour différencier les pays

Nous pouvons créer un système équivalent pour gérer les extensions relatives à chaque pays en appliquant le regex `^domaine.(ext1|ext2)|.domaine .(ext1|ext2)` au nom d'hôte en tant que filtre personnalisé à inclure.

Le principe est légèrement différent pour gérer les sous-domaines, il existe des variantes mais la plus simple à mettre en œuvre est la suivante.

1. Modifiez votre marqueur de suivi Google Analytics en ajoutant devant le nom de domaine un point qui permet de récupérer les sous-domaines.

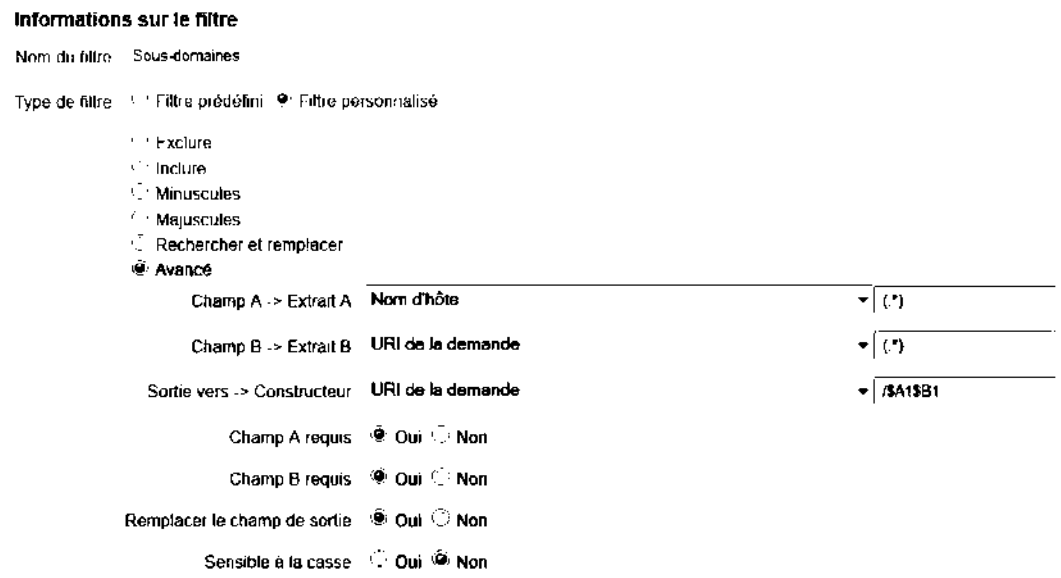
```
_gaq.push(['_setDomainName', '.nom-domaine.fr']);
```

2. Créez un filtre personnalisé *Avancé*.
3. Respectez les consignes de la figure 4-49 et appliquez le filtre.

Ainsi, les sous-domaines apparaîtront de manière différenciée sous la forme de répertoires tels que */www/*, */fr/*, */en/*...

Figure 4-49

Gestion des sous-domaines



Créer des tableaux de bord filtrés

Nous allons créer des tableaux de bord personnalisés à partir de filtres dans le but de ne conserver que les données qui nous intéressent. Suivez les étapes les unes après les autres et vous ne devriez rencontrer aucun souci particulier. Pour créer un nouveau rapport dans Google Analytics, cliquez sur le lien *Personnalisation* puis sur le bouton *Nouveau rapport personnalisé*.

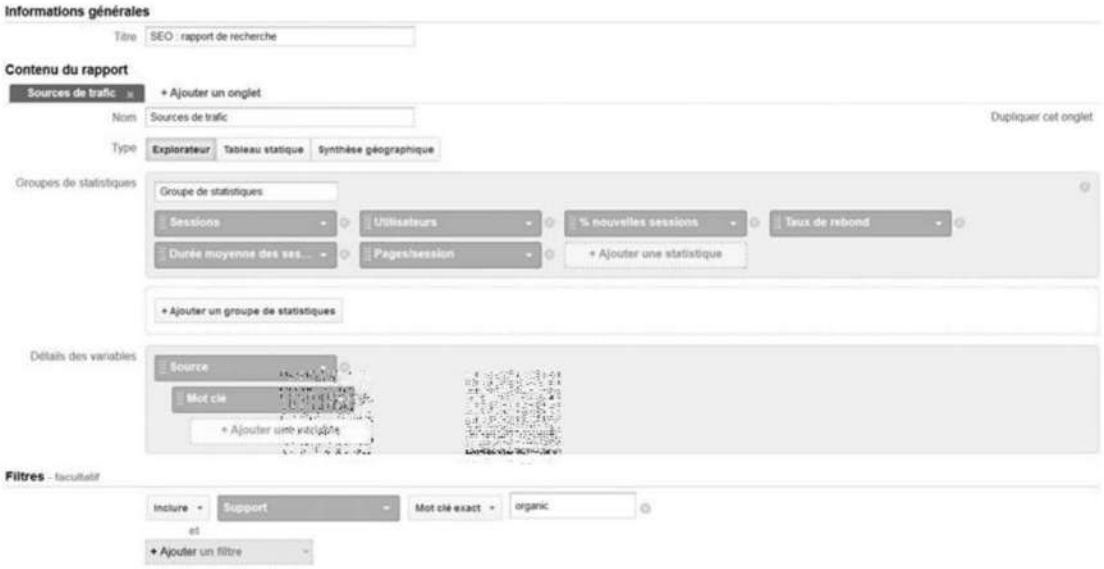
Suivre le référencement général

Ce rapport personnalisé va nous permettre de suivre les principales sources de trafic, les mots-clés ainsi que les pages de destination relatives à ces requêtes (landing pages). Il serait même possible de l'agrémenter encore davantage si nécessaire.

Nous allons créer trois onglets distincts dans ce rapport pour différencier les informations : le premier récupérera les données relatives aux adresses de référence, le deuxième les requêtes tapées et le dernier les pages de destination. Suivez les informations des figures 4-48 à 4-50 afin de créer facilement chaque onglet.

Figure 4-50

Création du premier onglet de suivi des sources de trafic par session



Le deuxième onglet est créé de la façon suivante :

Figure 4-51

Deuxième onglet créé pour suivre les requêtes tapées

Enfin, terminons par un tableau statique avec des variables simples pour composer le dernier onglet :

Figure 4-52

Troisième onglet pour les landing pages

Une fois le rapport terminé, vous devriez obtenir des courbes et des données chiffrées en fonction de plusieurs statistiques qui sont à suivre en termes de référencement.

Figure 4-53

Vue du rapport personnalisé SEO avec les adresses de référence dominantes

Étude des adresses de référence

Le deuxième tableau de bord personnalisé que nous allons créer nous permet d'afficher les pages de destination associées à leurs URL referers et à leur nombre de visites, leur taux de rebond… L'objectif est de se créer un classement des sources qui fournissent le plus de visiteurs en règle générale ou pour chaque page du site. Créez un nouveau rapport personnalisé avec un tableau statique et les variables *Sources* et *Chemin du site référent* (figure 4-52). Pour créer une variante avec les meilleures sources en général (et non par page), dupliquez l'onglet et retirez la variable *Chemin du site référent*. En enregistrant,

vous obtenez un rapport personnalisé qui présente alors les deux possibilités pour suivre les meilleures sources de vos sites web.

Figure 4-54

Création d'un rapport de suivi basé sur les URL referers

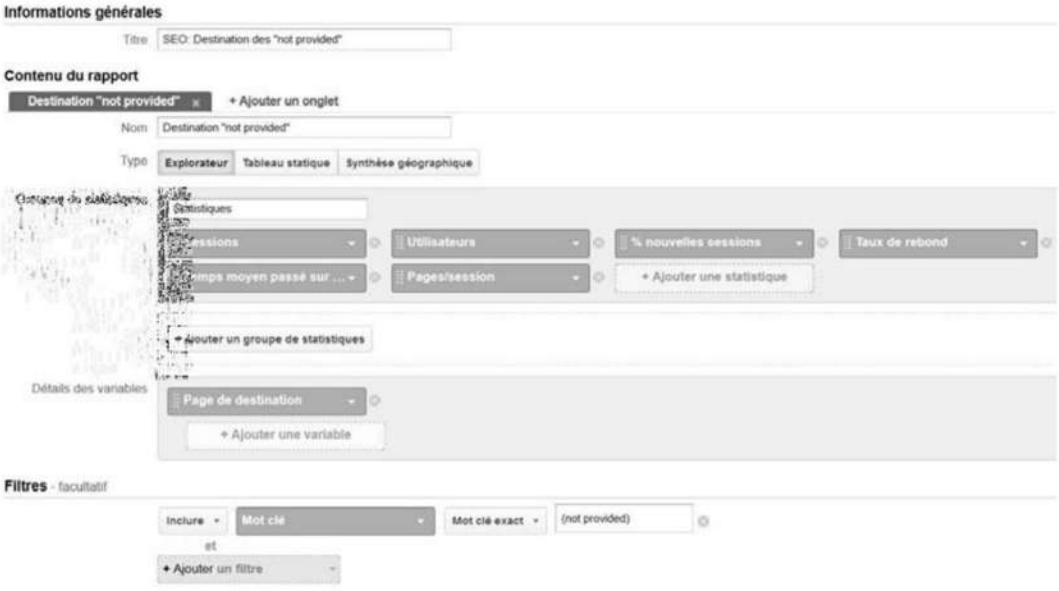
Analyse des pages de destination des not provided

Le principal ennemi des utilisateurs de Google Analytics est sans hésiter le *not provided* imposé par Google pour « protéger » les utilisateurs de son moteur de recherche. Comme nous ne pouvons pas savoir quels mots et expressions clés ont été tapés par les visiteurs, il peut être intéressant de suivre au moins les pages de destination qui sont la résultante de ces recherches.

Pour ce faire, un petit filtre ou un tableau de bord personnalisé peut faire l'affaire. Il suffit de créer un nouveau tableau avec pour variables les pages de destination et un filtre appliqué aux mots-clés not provided.

Figure 4-55

Création d'un rapport de suivi des « not provided »



Intérêt limité à cause des not provided...

Désormais, ce tableau de bord fait presque office de gadget tant les mots-clés cachés se sont multipliés sur Google et Bing notamment. En effet, l'analyse des pages de destination peut nous aider mais comme près de 95 % du trafic est marqué en not provided, il devient de plus en plus difficile de cibler les mots qui ont pu emmener les visiteurs sur les sites web.

Suivre la fréquence du crawl en direct

Il existe plusieurs méthodes pour suivre le crawl en direct dans Google Analytics mais toutes passent par l'usage d'un code en PHP ou dans un autre langage. Voici quelques ressources qui vous donneront satisfaction, nous n'en développerons qu'une seule ici :

- SEOLand : http://goo.gl/Z440Zu ;
- Watussi : http://goo.gl/wubKIJ ;
- Adrian Vender : http://goo.gl/HmscHV.

Dans ces trois articles, la méthode est toujours à peu près équivalente : créer un nouveau profil Google Analytics indépendant, récupérer le code de suivi et surtout l'identifiant du compte. Il faut ensuite faire appel à des fichiers PHP qui permettent de s'interfacer avec Google Analytics, puis d'appliquer les méthodes utiles.

Prenons le cas présenté par MrBoo et Watussi. Nous devons tout d'abord créer notre profil Google Analytics indépendant, récupérer l'identifiant, puis écrire les quelques lignes de code suivante :

```
<?php
include_once 'class/Galvanize.php';
if(strstr($_SERVER['HTTP_USER_AGENT'] ,'Googlebot')){
    $GA = new Galvanize('UA-XXXXXXXX-1');
    $GA->trackPageView();
}
?>
```

Après un certain temps, une courbe va se dessiner et montrer les variations du crawl des robots sur le site web. Cela est d'autant plus intéressant si vous analysez ces statistiques lorsqu'il s'agit d'un site récent, d'une refonte ou tout du moins d'une profonde mise à jour.

Peut-on contrer les not provided ?

Comme nous l'indiquons depuis le début de ce chapitre voire de ce livre, il est de plus en plus fréquent que les moteurs de recherche masquent ou détournent leurs URL de référence afin de sécuriser les moteurs de recherche et améliorer la confidentialité des visiteurs. Sur le principe, nous sommes plutôt tous d'accord sur le fait que ce type de pratique est plutôt sain pour les internautes mais dans les faits,

nous constatons surtout la colère des professionnels qui ne peuvent même plus jauger la qualité de leurs mots-clés et de ceux qui font la force de chaque page web.

Force est de constater qu'il va falloir s'habituer au not provided car les moteurs tels que Bing et Google vont de plus en plus propager ce type de pratique (c'est déjà le cas quasiment) afin de rassurer les internautes mais aussi pour rediriger les entreprises vers les liens sponsorisés qui permettent encore de savoir quels mots ont été requêtés (jusqu'à quand ?).

Du point de vue technique, nous avons vu lors de notre suivi avec PHP que les URL referers étaient parfois masquées ou modifiées par les moteurs de recherche. C'est ce problème qui explique le nombre de not provided si conséquent.

En toute transparence, de multiples tests ont été effectués lors de la rédaction du livre que vous avez entre les mains afin de trouver des solutions techniques à ce problème majeur. Aucune n'a abouti…

Pour que vous ne perdiez pas votre temps, nous allons dresser les quelques alternatives qui ont été testées, en espérant que nous trouverons des solutions à l'avenir autour de ces méthodes ou non pour enfin récupérer les requêtes des internautes :

- récupération des URL referers et redirection vers la même adresse avec HTTP au lieu de HTTPS. Cela ne peut pas fonctionner car nous n'avons pas la requête dès la réception de l'adresse de référence ;
- récupération et enregistrement des URL referers, puis retour sur l'URL de la page précédente avec JavaScript pour récupérer la requête dynamiquement. Sur le principe, l'idée est bonne mais nous n'avons pas accès à l'historique des navigateurs, l'objet JavaScript `history` permet uniquement de retourner à la page précédente ou d'aller vers la suivante, nous n'avons pas accès à l'URL en tant que telle ;
- génération de requêtes dynamiques avec cURL pour capter le positionnement des pages web et comparer avec les données possibles de Google Analytics. Cette méthode est la seule qui peut donner quelques éléments de satisfaction mais au fond, elle ne permet pas de récupérer la requête des internautes, c'est seulement une simulation du positionnement sur des mots-clés.

Tout au long de ce chapitre ont été présentées des méthodes plus limitées que cette dernière car aucune n'a rendu de bons résultats. Il n'est pas certain qu'il soit possible de vraiment récupérer les not provided et il faut bien avouer que Google a dû prévoir le coup face aux petits malins que nous sommes. Cependant, il ne faut pas lâcher prise et se diriger peut-être vers des solutions Black Hat pour essayer au moins de réduire la proportion de mots-clés…

Pour conclure, nous pouvons donc dire qu'il est impossible à ce jour de déterminer les expressions saisies par les utilisateurs dans le champ de recherche de Google et de Bing s'ils sont en HTTPS. Aucun test n'a donné de solution satisfaisante, il faut donc se référer aux quelques techniques présentées dans ce chapitre ou sur la Toile. En voici quelques-unes qui seront peut-être des rappels pour certains.

- Dans Google Analytics, sélectionner le menu *Acquisition>Optimisation du référencement>Requêtes* qui permet de récupérer les mots-clés, le nombre d'impressions et de clics présentés dans la Google Search Console. Même si les données sont loin d'être excellentes, c'est déjà mieux que rien…

- Installer le kit de suivi des not provided proposé par le site notprovidedkit.com. Il ajoute un tableau de bord adapté et personnalisé autour de la problématique des expressions cachées. Il est possible d'ajouter des filtres pour l'agrémenter encore davantage et avoir un suivi multiple.
- Savoir de quel type de source proviennent les not provided pour limiter leur impact et qualifier davantage les sources. La technique a été présentée par Tim Resnik du site moz.com en mai 2013 (source : http://goo.gl/ByifWG) et a pour but de récupérer et traduire à la volée les données de l'argument `ved` dans les URL de Google. L'idée est excellente et permet de savoir si la recherche provient de la partie Actualités, Images ou d'une autre source en particulier de la recherche universelle.
- Procéder à un suivi précis des requêtes en excluant les not provided et en extrapolant les données. Cette méthode est imprécise mais peut toujours aider sur de petits sites web, par exemple. Il est aussi possible d'analyser les recherches sur site avec Google Analytics ou directement par nos propres moyens afin de voir les recherches effectuées dans les sites web. Il est probable qu'elles correspondent à peu près aux requêtes formulées par les internautes dans les moteurs de recherche.
- Rassembler les données disponibles entre les diverses solutions, les comparer et essayer de récupérer des chiffres les plus pertinents possibles.

Malheureusement, nous ne pouvons pas faire mieux pour le moment ni vous fournir de solution idéale pour contrer le drame causé par les not provided sur Bing et Google notamment. Il est nécessaire de continuer de chercher et de procéder à une veille à ce sujet, en espérant qu'un miracle se produise à l'avenir…

Conclusion sur Google Analytics

Google Analytics est une solution très complète et complexe qui ne mérite pas d'être résumée en quelques lignes. Nous avons tenté de présenter les points essentiels afin de pouvoir suivre de manière générale le référencement et de montrer l'étendue des capacités de suivi mises à notre disposition.

Il existe en réalité autant de façons de faire que de spécialistes puisque chacun use de ses propres méthodes pour capter et analyser les données (cela peut même passer outre les solutions Google Analytics avec des rapports générés en C#, PHP, Java…).

Parfois, un suivi classique avec les données fournies nativement par le service suffisent amplement si les KPI restent abordables tandis qu'il faut dans d'autres cas personnaliser de nombreux codes de suivis et utiliser des filtres ou des rapports personnalisés pour obtenir satisfaction. Nous n'avons donc pas pu faire le tour de la question et nous vous recommandons vivement de consulter des ouvrages spécialisés sur le sujet pour gagner en compétences.

Analyse qualitative et ROI

Les outils proposés vous fournissent des statistiques sur différents aspects : le nombre de visiteurs, les pages consultées, la répartition géographique, les terminaux utilisés, le taux de rebond ainsi que le positionnement des pages sur des mots-clés spécifiques. Il s'agit ici d'une étude quantitative : vous avez des chiffres que vous devez interpréter. C'est ici que l'analyse qualitative intervient…

Avoir des chiffres est une chose mais les comprendre, les analyser et les exploiter pour en tirer profit en est une autre. Il n'existe aucun outil pour mener à bien ces études qualitatives, nous ne pouvons compter que sur nos compétences et notre expérience...

Vous devez absolument garder à l'esprit les objectifs que vous vous êtes fixés pour le site (KPI) et le choix du public cible tout au long de l'analyse, c'est le seul moyen de ne pas s'éparpiller ou croire que tout le travail fourni n'est pas efficace. Chaque action que nous menons a un but précis, nous ne pouvons pas demander plus que ce qu'il est possible d'obtenir...

Pour analyser les données, vous pouvez vous poser quelques questions intuitives. En voici quelques exemples.

- « Qui sont les visiteurs ? » Connaître sa cible permet de la satisfaire et de toujours lui proposer des produits/solutions en adéquation avec ses besoins et attentes.
- « Quelle langue parlent les visiteurs ? » Peut-être sont-ils plus anglophones que français alors que vous n'avez pas prévu un site multilingue initialement ?
- « Comment les internautes sont-ils arrivés sur le site ? » Peut-être est-ce directement grâce au référencement naturel ou aux liens sponsorisés ? Peut-être est-ce par bouche à oreille ? Peut-être est-ce par un média social ou par un blog ? Il faut mesurer l'impact du référencement naturel dans cette multitude de possibilités afin de calculer le pourcentage de réussite de la stratégie SEO.
- « Quel est le ratio entre les nouveaux visiteurs et le trafic connu ? » Selon l'objectif, le ratio ne doit pas être interprété de la même manière. Vous êtes peut-être plutôt dans une optique d'acquisition de nouveaux clients, auquel cas le nombre de nouveaux visiteurs doit être supérieur à celui des internautes déjà connus. Ou alors, vous préférez travailler la fidélisation, et dans ce cas, c'est plutôt le nombre de visiteurs connus qui doit être supérieur aux taux de nouvelles visites...
- « Quel terminal utilisent mes visiteurs pour consulter mon site ? » Imaginez que votre site n'est pas adapté à un usage sur support mobile, cela pourrait être très embêtant si la majorité des visites proviennent de ces terminaux.
- « Quelles pages sont les plus visitées ? » Ce sont ces pages qu'il va falloir utiliser pour insérer les messages les plus importants en fonction des objectifs, afin d'améliorer la visibilité des contenus voire la fidélité des lecteurs.
- « À quelle heure le site reçoit-il le plus de visites ? » Cette information est importante car elle vous permettra de faire vos actions de promotion dans les heures idéales pour toucher un maximum de personnes...
- « Combien de temps les visiteurs passent-ils sur le site ? » Si le temps passé sur le site est trop court ou que le taux de rebond est important, cela peut signifier que les visiteurs n'ont pas trouvé ce qu'ils cherchaient. Par conséquent, peut-être que les expressions clés que nous avons optimisées ne collent pas assez bien aux souhaits des internautes et à la réalité du marché ?

Nous n'allons pas entrer dans les méandres du webmarketing car cela est un autre métier et dépasse le cadre précis de notre propos. Néanmoins, il est intéressant au-delà du suivi de savoir répondre aux questions que nous avons présentées auparavant mais aussi de pouvoir analyser les chiffres avec intelligence et précision.

Dans un plan marketing, nous devons suivre plusieurs types de données pour maîtriser notre image de marque et la qualité de notre communication de A à Z :

- effectuer un suivi du référencement pour mesurer l'impact des efforts consentis ;
- mener un suivi le plus précis possible du positionnement pour évaluer les chances de visibilité ;
- vérifier les données internes du site ainsi que ses qualités intrinsèques (audit complet, nous en parlerons dans le prochain chapitre) en matière d'ergonomie, de code, de rédaction, etc. ;
- évaluer la notoriété et l'e-réputation des marques, des produits, des services, des outils ou encore des personnes qui gravitent autour de nos sites ;
- mesurer la fidélité et le niveau de satisfaction des visiteurs afin de leur donner sans cesse envie de revenir ;
- calculer si possible un retour sur investissement (ROI, *Return On Investment*), un retour sur engagement (ROE, *Return On Engagement*), un retour sur attention (ROA, *Return On Attention*) ou un retour sur objectifs (ROO, *Return On Objectives*) afin de mieux prendre conscience de notre travail.

Pour clôturer ce chapitre, nous allons justement évoquer ce dernier point en essayant de donner des méthodes de calcul des ROI, ROE, ROA et ROO. Il n'est pas toujours aisé de mesurer ces facteurs car nous manquons souvent de données suffisantes pour cela.

L'idéal est de s'appuyer sur les statistiques fournies par Google Analytics ou une autre solution du même type, voire de réaliser votre propre suivi stratégique en utilisant les URL referers. L'idée est de capter le pourcentage de visiteurs provenant de diverses sources, cela permet de déduire rapidement quelle part a joué le référencement naturel dans le trafic global du site web.

Les entreprises souhaitent généralement calculer leur retour sur investissement afin de savoir si les actions effectuées en SEO et l'argent dépensé pour ce travail en a valu la peine. Dans les faits, il s'agit du facteur le plus dur à mesurer car nous ne vendons pas tous des produits ou nous ne cherchons pas tous à produire directement de l'argent. Nous manquons donc de données et de précisions pour donner un ROI crédible et viable.

C'est là qu'interviennent les retours sur objectifs, sur attention ou encore sur engagement. Tous ces sigles parfois peu équivoques sont souvent plus en adéquation avec le travail fourni par les référenceurs. En effet, le ROI retourne une rentabilité dans un secteur qui ne permet pas toujours de le mesurer. En revanche, ces nouveaux modes de calcul vont permettre d'analyser l'impact réel du référencement naturel, même indépendamment du référencement payant.

Voici quelques définitions générales de ces termes dont le sens est parfois approchant.

- ROI : calcul du chiffre d'affaires total généré en fonction d'un budget initial dépensé pour les actions menées. Ce facteur doit déterminer les retombées économiques des actions SEO.
- ROO : méthode de calcul qui analyse les statistiques générales relatives au référencement pour mesurer le succès des actions menées en fonction des objectifs préalablement fixés. Par exemple, si l'objectif initial et principal est de gagner des visiteurs sur le site, il faut calculer le ratio entre le nombre moyen de lecteurs avant la campagne SEO et après avoir effectué le travail. Il est également possible d'ajouter une notion temporelle afin de mesurer l'efficacité des actions pour mener à bien les objectifs.

- ROE : calcul visant à mesurer l'impact et le rôle des gestionnaires de site dans la participation et la fidélisation des visiteurs mais aussi dans l'amélioration de la notoriété globale. Il s'agit davantage d'un facteur basé sur la qualité des contenus, des actions menées sur les plates-formes sociales ou encore des efforts fournis pour améliorer l'efficience et le confort sur les sites web. Ce n'est donc pas un facteur directement lié au SEO, mais sa finalité peut l'être en revanche... En d'autres termes, le ROE a un double rôle : savoir si les gestionnaires interagissent avec les visiteurs et font tout pour les satisfaire, mais aussi déterminer si les visiteurs partagent et effectuent un bouche à oreille positif à l'égard du site ou de l'entreprise.
- ROA : technique ayant pour objectif de calculer une estimation de la popularité d'un site ou d'une entreprise en fonction des actions effectuées par la communauté web et les visiteurs. Il s'agit de savoir si les actions menées avec le SEO ont permis aux visiteurs de tomber sur des contenus, des services ou des produits qui leur ont plu au point de partager et promouvoir le site (ou les actions relatives au site). Le ROA se différencie du ROE dans le sens où le calcul ne porte plus sur les gestionnaires du site mais sur les visiteurs. Nous devons déterminer la notoriété obtenue par l'entreprise grâce aux actions menées via le site web.

Ces différents facteurs montrent que la notion d'argent n'est pas toujours au centre des préoccupations. Il arrive fréquemment que l'argument des clients voulant référencer un site soit d'obtenir un maximum de visiteurs.

Certes, nous nous doutons bien que l'objectif final est d'améliorer la notoriété ou d'augmenter le nombre de ventes, de souscriptions à des formulaires ou de téléchargements de fichiers, etc., mais au fond, le but de le SEO est de booster le nombre de visiteurs.

Dire qu'un référencement est raté car le ROI est faible est en soi une forme d'antagonisme, sauf si le nombre de visiteurs uniques et de nouveaux visiteurs est resté stable malgré les efforts fournis. En effet, ce n'est pas parce que nous gagnons des centaines voire des milliers de visiteurs par mois grâce au référencement que nos ventes vont nécessairement exploser. C'est plutôt le travail consenti pour réaliser un site qualitatif, clair, efficient et ergonomique qui permet de convertir ces nouveaux visiteurs. Il est donc un peu sévère de mettre tout le poids d'un échec sur une campagne SEO...

Le retour sur objectifs est sûrement le facteur le plus efficace pour mesurer l'impact du référencement. De ce dernier découle le calcul des ROI, ROE et ROA qui permettent d'analyser la stratégie marketing complète et les retombées en termes d'économie, de notoriété et d'e-réputation.

Partant de ce constat, nous pouvons analyser toutes les données statistiques que nous avons obtenues lors de nos diverses phases de suivi et mesurer dans tous les sens l'impact du référencement naturel dans le plan marketing du site. Voici quelques exemples de formules pour calculer chacun de ces facteurs :

Le calcul du ROI suit une formule simple et directement liée à des notions économiques.

- ROI (en %) = (gains réalisés – coût de l'investissement) / coût de l'investissement.

Le ROO peut être calculé de plusieurs manières différentes en fonction des objectifs fixés initialement. L'avantage est de pouvoir multiplier les calculs pour savoir en quoi le SEO a été le plus efficace. Voici quelques exemples simples.

- ROO (en %) = nombre de visiteurs total (durée *t*) / nombre de visiteurs total (durée *t+1*).
- ROO (en %) = nombre de nouveaux visiteurs / nombre de visiteurs fidèles.
- ROO (en %) = nombre total de pages vues (durée *t*) / nombre total de pages visitées (durée *t+1*).
- ROO (en %) = nombre de téléchargements effectués après une visite issue d'un moteur de recherche (obtenu grâce à un tracking) / nombre total de téléchargements.
- ROO (en %) = nombre de formulaires souscrits après une visite issue d'un moteur de recherche (obtenu grâce à un tracking) / nombre total de souscriptions.
- ROO (en %) = nombre de ventes réalisées après une visite issue d'un moteur de recherche (obtenu grâce à un tracking) / nombre total de ventes.

Le ROE peut également être calculé de plusieurs façons différentes en fonction des actions menées.

- ROE (en %) = (nombre total de réponses et d'interactions des gestionnaires du site) / (nombre total de messages + commentaires + autres échanges).
- ROE (en %) = (nombre de J'aime + commentaires + partages + souscriptions...) / nombre de visiteurs actifs.

Enfin, le ROA est certainement la méthode de calcul la plus complexe à mesurer, sauf peut-être en s'appuyant sur des outils de mesure sociale comme Klout ou PeerIndex. Le ROA peut être lié à une question de temps écoulé pour obtenir satisfaction (visibilité, notoriété, etc.). Voici deux exemples de formules bien différentes pour déterminer ce type de facteur.

- ROA (en %) = temps écoulé × (nombre d'impressions + nombre de personnes atteintes + notoriété et retours obtenus).
- ROA (en %) = nombre de partages × (commentaires + J'aime + souscriptions (...) obtenus) / nombre de personnes touchées (ou nombre d'impressions).

Le paradoxe de toute campagne est parfois de calculer ce que les marketeurs appellent le RONI *(Return On Non Investment)*. En effet, il s'agit d'un critère qui vise à déterminer le risque de pertes envisageables si nous ne menons pas des actions SEO ou sociales sur la Toile. Le RONI sert surtout à montrer qu'il est préférable de dépenser quelques sommes pour améliorer sa visibilité et sa notoriété sur le Web plutôt que de passer inaperçu. En d'autres termes, le RONI démontre qu'il vaut mieux prévenir que guérir...

Maintenant que nous savons mener une analyse qualitative des données statistiques, il est intéressant de savoir comment auditer un site web pour le rendre meilleur et améliorer son référencement global. La suite au prochain chapitre...

5

L'audit SEO

L'audit SEO est un état des lieux d'un site qui permet d'analyser les critères importants pour favoriser un bon référencement et identifier les faiblesses du site ou les éventuels facteurs bloquants. L'audit contient des analyses et recommandations qui sont établies selon trois grands axes principaux : technique, contenu et popularité.

Figure 5-1

Trois axes principaux d'un audit SEO

- **L'audit technique** va mettre en exergue les principaux facteurs bloquants pour la bonne prise en compte du site par les moteurs de recherche. L'objectif ici est de s'assurer du bon fonctionnement du site web et de sa bonne indexation dans les index des moteurs. On parle donc d'étudier tous les critères favorisant un bon crawl par les robots d'indexation ainsi que les quelques facteurs d'optimisation du positionnement.
- **L'audit du contenu** a comme objectif d'analyser les contenus d'un site et de trouver les optimisations possibles à effectuer afin d'améliorer la compréhension des contenus par ses deux cibles principales : les moteurs de recherche et les internautes.
- **L'audit de la popularité** analyse la popularité (notoriété et e-réputation) et l'autorité d'un site sur la Toile. Des solutions pour améliorer la notoriété seront envisagées.

Avant de voir quels facteurs il convient d'analyser, sachez qu'il existe des outils en ligne qui vous proposent des mini audits que nous présenterons dans la suite de ce chapitre. Il suffit d'insérer l'URL de votre

choix et les outils lancent alors l'analyse d'une multitude de critères. Attention, certaines informations sont à prendre avec des pincettes. Il s'agit d'indicateurs parfois peu fiables, c'est plutôt l'idée générale qui doit être prise en compte. Dans tous les cas, rien ne vaut un audit fait à la main par un professionnel.

L'audit technique

Le nom de domaine

Le premier critère à analyser est le nom de domaine car il s'agit d'un facteur majeur, et pas uniquement en termes de SEO. En effet, le nom de domaine a un aspect marketing notable car sa longueur, sa facilité de mémorisation et surtout les mots qui le composent ont un rôle déterminant sur les prospects mais aussi pour les moteurs de recherche. Il convient de se poser quelques questions majeures.

- S'agit-il d'un EMD *(Exact Match Domain)* ou d'un PMD *(Partiel Match Domain)*. Peut-il avoir une conséquence néfaste sur le positionnement du site ?
- S'agit-il d'un ccTLD (*CountryCode Top Level Domain* : .fr, .ro, .be, .de...) ou d'un gTLD (*Generic Top Level Domain* : .com, .net, .info, .org...) ?

D'autres points sont aussi à surveiller sur un nom de domaine :

- le propriétaire *(registrar)* ;
- l'ancienneté (son âge à partir de sa date de création). En théorie, plus un nom de domaine est ancien, plus les moteurs de recherche lui accordent de l'importance. Plus le nom de domaine est jeune, plus il faut multiplier les efforts en matière de SEO afin d'être positionné de manière « stable » dans les moteurs de recherche ;
- le lieu d'hébergement. L'hébergement est-il cohérent avec le ccTLD ? Par exemple, un .fr doit être hébergé en France en théorie (bien qu'il puisse être intéressant dans certains cas d'avoir des adresses IP localisées à l'étranger). Seuls les gTLD sont neutres et n'impliquent pas de localisation particulière ;
- son historique. Le nom de domaine a-t-il été pénalisé auparavant ? La Google Search Console vous-vous informe sur les éventuelles pénalités déclarées (spam, liens artificiels...) ainsi que les demandes de réexamens antérieures.

Faisons tout de même une remarque, il existe un mythe jamais vérifié qui prétend que la durée de « location » d'un domaine a un impact sur le positionnement. Par exemple, si vous achetez le domaine site.com pour trois ans, il sera mieux valorisé que s'il vous l'aviez commandé pour une seule année chez un *registrar*. En réalité, cela n'a jamais été prouvé et même Matt Cutts, le célèbre porte-parole non officiel de Google, a affirmé que la durée d'achat d'un nom de domaine n'avait aucun impact sur le positionnement.

En tout état de cause, cela semble relativement logique puisque techniquement, les moteurs ne sont pas des comptables qui passent leur temps à éplucher les fichiers clients des registrars. Certes, les dates d'expiration sont souvent accessibles mais des hébergeurs permettent de masquer ces données. Il arrive donc fréquemment qu'elles soient inaccessibles aux moteurs et il serait encore plus étonnant qu'une telle protection devienne un désavantage pour les propriétaires de site web en matière de positionnement...

Figure 5-2

Site avec données cachées (dont la date d'expiration)

```
domain: internet-formation.fr
status: ACTIVE
hold: NO
holder-c: ANO00-FRNIC
admin-c: ANO00-FRNIC
tech-c: OVH5-FRNIC
zone-c: NFC1-FRNIC
nsl-id: NSL22817-FRNIC
registrar: OVH
anniversary: 19/05
created: 19/05/2009
last-update: 20/05/2009
source: FRNIC

ns-list: NSL22817-FRNIC
nserver: dns15.ovh.net
nserver: ns15.ovh.net
source: FRNIC

registrar: OVH
type: Isp Option 1
address: 2 Rue Kellermann
address: ROUBAIX
country: FR
phone: +33 8 99 70 17 61
fax-no: +33 3 20 20 09 58
e-mail: support@ovh.net
website: http://www.ovh.com
anonymous: NO
registered: 21/10/1999
source: FRNIC
```

Pour trouver les informations relatives à un nom de domaine, vous pouvez utiliser plusieurs sites web reconnus tels que whois.net pour trouver la date de création et d'expiration du nom de domaine, des renseignements sur le propriétaire et l'hébergeur du site, etc. La Wayback Machine du projet Internet Archive (source : http://goo.gl/yzcx3w) vous permettra quant à elle de suivre l'historique d'un site web et de visualiser l'état du site depuis ses origines, lorsque les données sont disponibles.

Notez également que vous pouvez effectuer des captures de votre site lors de votre visite afin d'en conserver une trace…

Se méfier des Whois anonymes ?

Certains référenceurs pensent qu'un Whois anonyme peut parfois renvoyer un mauvais signal aux moteurs de recherche car c'est un signe d'une volonté de « cacher » des informations, généralement pour des sites spammy. Optez pour un Whois transparent si vous craignez un impact négatif (source : http://goo.gl/Q3D74g). Toutefois, sachez que cela n'a jamais été vérifié ni confirmé de la part des divers moteurs de recherche. Il s'agit encore d'une inconnue autour de laquelle le débat reste ouvert…

Figure 5-3

Données non masquées pour le blog miss-seo-girl.com

Résultat whois pour miss-seo-girl.com

```
Domain Name: miss-seo-girl.com
Registry Domain ID: 1724075652_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.gandi.net
Registrar URL: http://www.gandi.net
Updated Date: 2014-03-10T11:35:08Z
Creation Date: 2012-05-31T06:39:58Z
Registrar Registration Expiration Date: 2015-05-31T06:39:58Z
Registrar: GANDI SAS
Registrar IANA ID: 81
Registrar Abuse Contact Email: abuse@support.gandi.net
Registrar Abuse Contact Phone: +33.170377661
Reseller:
Domain Status: clientTransferProhibited
Domain Status:
Domain Status:
Domain Status:
Domain Status:
Registry Registrant ID:
Registrant Name: Alexandra Martin
Registrant Organization:
Registrant Street: Gandi, 63-65 boulevard Massena
Registrant City: (Gandi) Paris
Registrant State/Province:
Registrant Postal Code: (Gandi) 75013
Registrant Country: (Gandi) FR
Registrant Phone: (Gandi) +33.170377666
Registrant Phone Ext:
Registrant Fax: (Gandi) +33.143730576
Registrant Fax Ext:
Registrant Email: 40d275927f662edbf3f59e69d3567a3a-1504080@contact.gandi.net
Registry Admin ID:
Admin Name: Alexandra Martin
```

Le fichier robots.txt

Un fichier `robots.txt` doit idéalement être présent sur tous les sites web. S'il n'est pas obligatoire, son rôle est tel qu'il serait étonnant qu'un site optimisé ne possède pas ce fichier si important. Il est destiné aux robots des moteurs et a pour objectif de leur interdire d'indexer certaines pages inutiles ou confidentielles ainsi que des fichiers de notre choix (par exemple, les images, les fichiers PDF...).

Il faut contrôler sa présence mais aussi son exactitude. Comme le fichier `robots.txt` porte toujours le même nom et doit toujours être placé à la racine du site, il suffit de taper dans la barre d'adresse /robots.txt après le nom de domaine pour voir s'il est présent ou pas (par exemple, www.miss-seo-girl.com/robots.txt).

Il convient de vérifier que les pages inutiles sont interdites au crawl et à l'indexation : pages dupliquées, pages sans contenu, pages de résultats de recherche sur le site, pages de connexion à la partie « administration », parties relatives au backoffice (souvent un dossier complet).

Assurez-vous également que l'URL du fichier `sitemap.xml` est bien indiquée car elle permet à tous les moteurs compatibles avec le protocole de mieux crawler votre site. Pour rappel, la ligne ressemble à ceci :

```
Sitemap: http://www.miss-seo-girl.com/sitemap.xml
```

Vérifiez surtout que vous n'interdisez pas l'indexation totale de votre site (avec la règle `disallow: /`) et que le fichier `robots.txt` ne comporte aucune erreur car cela bloquerait l'indexation complète du site. Cela s'explique car le fichier `robots.txt` est l'un des tous premiers fichiers lus par les crawlers et une erreur bloque alors la lecture de tout ce qui suit comme l'a indiqué Eric Kuan sur le forum d'entraide de Google (source : http://goo.gl/9khUhR).

Le fichier robots.txt n'est pas toujours lu

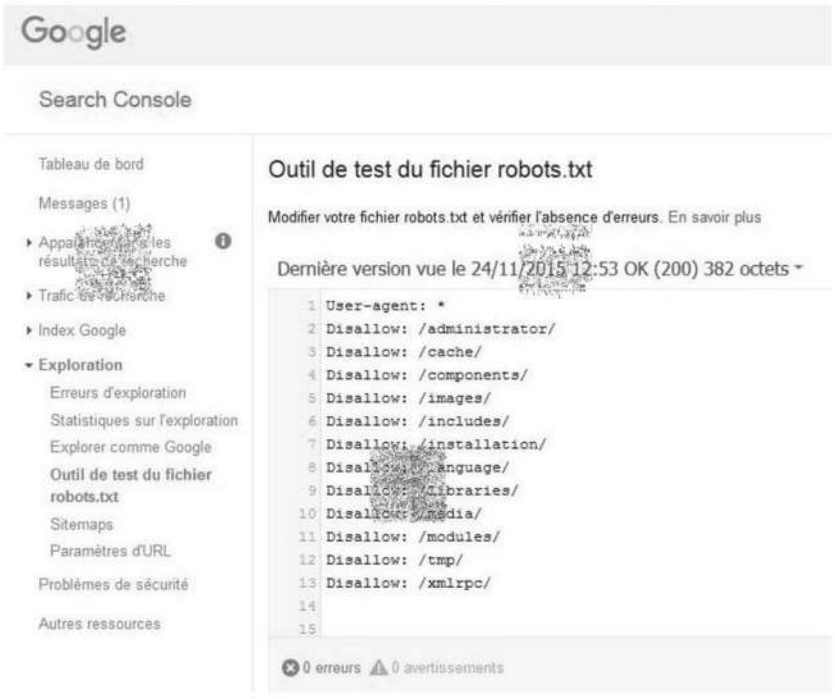
Google nous a indiqué que le fichier `robots.txt` ne devait pas être généré de façon automatique car il n'est pas lu à chaque passage des robots. Cela sous-entend que le `robots.txt` est crawlé seulement de temps en temps. C'est une raison de plus pour en prendre soin, car une erreur pourrait mettre plus de temps à être résolue et réparée par Google (source : http://goo.gl/YVLbBi).

Contrôlez que le fichier `robots.txt` renvoie un code d'erreur 200 (ce qui signifie qu'il n'existe aucune erreur) afin de ne pas bloquer la lecture des robots d'indexation. Notez surtout qu'il est important de ne pas bloquer les ressources utiles à Googlebot comme les feuilles de styles CSS, les scripts JavaScript (notamment pour le crawl de l'Ajax) ou les ressources HTTPS.

Il est possible de contrôler le fichier `robots.txt` dans la Google Search Console, via l'option *Outil de test du fichier robots.txt*. Ainsi, vous pourrez détecter des erreurs. Cela ne suffit pas car l'outil ne fait que mettre en avant les erreurs « techniques » au sein du fichier et non les blocages du crawl. Dans ce cas, testez le crawl avec l'option *Explorer comme Google* puis optez pour l'option *Explorer et afficher*. Cela vous indiquera si des ressources sont injustement bloquées.

Figure 5-4

Test du fichier robots.txt dans la Google Search Console



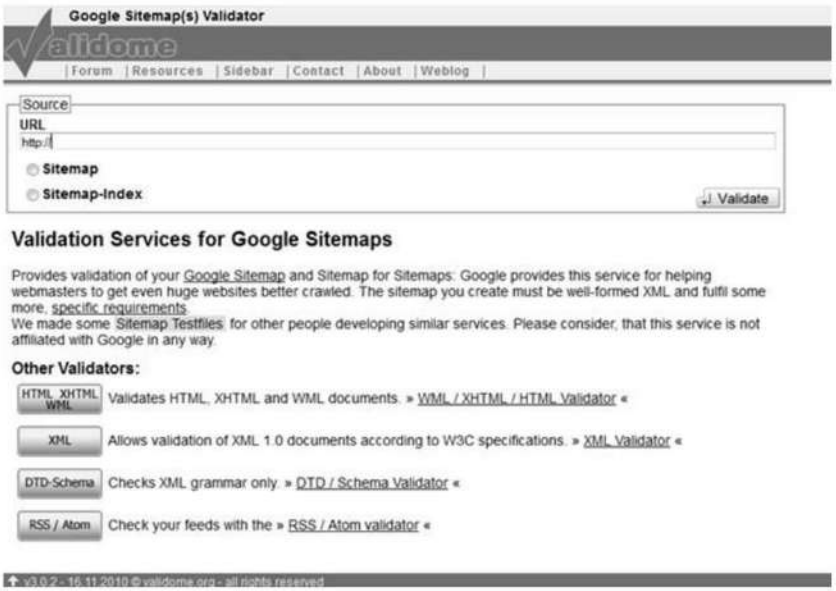
Le fichier sitemap.xml

L'existence du ou des fichiers `sitemap.xml` est essentielle tant l'indexation prend une autre mesure lorsqu'il(s) existe(nt). Veillez à ce que toutes vos URL importantes soient présentes. N'oubliez pas de soumettre votre fichier via la Google Search Console et Bing Webmaster Center, par exemple, et assurez-vous de sa mise à jour régulière. Parallèlement, ajoutez bien l'URL du ou des fichiers Sitemap dans le fichier `robots.txt` comme nous l'avons indiqué auparavant.

Vous prémâchez ainsi le travail des crawlers. N'oubliez pas que même si le fichier n'a pas une importance directe sur le référencement d'un site web, il demeure primordial pour accélérer le processus d'indexation, faciliter le crawl et fournir à Google des informations sur le site : fréquence des mises à jour des pages, date de dernière modification de chacune d'elles, priorité relative au goût du webmaster...

Figure 5-5

Validome vérifie l'exactitude des fichiers sitemap.xml



Trois outils permettent de vérifier que vos fichiers Sitemap respectent les règles :

- http://www.xml-sitemaps.com/validate-xml-sitemap.html ;
- http://sitemapxml.net/sitemap-validator.php ;
- http://www.validome.org/google/.

La qualité du code source

De nombreux points sont à aborder lorsque nous vérifions la qualité des codes sources d'un site.

- Vérifiez si votre code est bien valide W3C (HTML et CSS) via le site http://validator.w3.org pour le code HTML ou avec http://jigsaw.w3.org/css-validator/ pour le CSS.

Figure 5-6

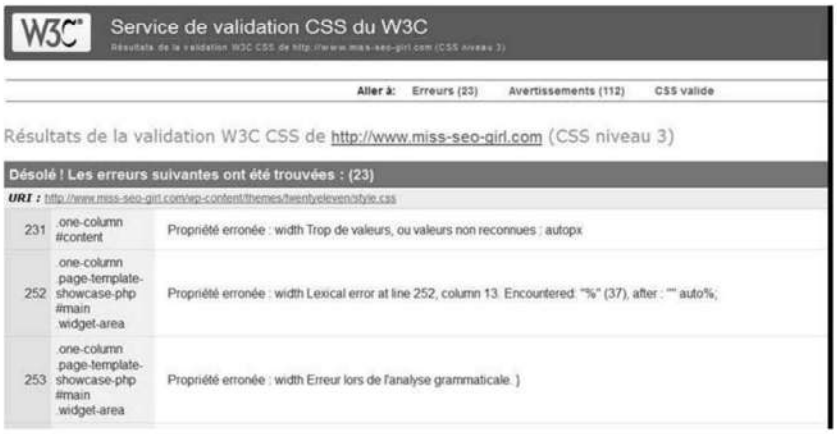
Validateur HTML (et CSS) du W3C pour contrôler la propreté du code source

Figure 5-7

Des erreurs ont été trouvées et devront être corrigées pour améliorer le site web.

Figure 5-8

Exemple de résolutions pour des erreurs de CSS



- Vérifiez l'encodage des caractères. Ce critère est important pour assurer un bon affichage partout dans le monde et sur tous les navigateurs (de préférence, optez pour l'UTF-8 car il contient tous les caractères mondiaux dont les kanjis asiatiques et surtout parce que la plupart des outils internationaux tels que les webmails ou les plug-ins sont basés sur cet encodage en Unicode).

> **Vérifier l'encodage des caractères**
>
> Pour contrôler l'encodage, dans le code source, cherchez la balise `<meta charset="UTF-8"/>` en HTML 5 ou `<meta content="text/html; charset=UTF-8" http-equiv="content-type"/>` en xHTML (ou HTML 4). Sinon, sachez qu'il peut être indiqué uniquement à l'aide d'un fichier `.htaccess` ou via une fonction PHP. Il convient de vérifier les en-têtes HTTP dans ce cas...

- Vérifiez que les balises `<title>` (longueur, optimisation, nombre d'occurrences des mots-clés…) et les métadonnées (balises meta `description` voire `keywords`) sont remplies, soignées et différentes sur toutes les pages du site.
- Vérifiez éventuellement que des balises meta `robots` (avec des valeurs telles que `noindex, nofollow` ou `noindex, follow`) sont bien utilisées, notamment si un fichier `robots.txt` vient à manquer. En revanche, sachez que les valeurs `index, follow` sont tout bonnement inutiles et peuvent même avoir un

impact un peu négatif sur le site (surcharge de la page, augmentation du temps de lecture par les robots et les serveurs, ligne supplémentaire qui va repousser un peu plus les contenus vers le bas du code...).

- Vérifiez que votre fichier CSS est appelé dans votre code source et que votre style CSS n'est pas présent directement dans le code (autour de balises `<style>...</style>`), cela diminue les performances de chargement.
- Attention car beaucoup d'extensions présentes dans des CMS tels que WordPress rentrent dynamiquement du style en « dur » dans le code. Il convient idéalement d'avoir le moins possible de fichiers CSS mais surtout de nettoyer tous ces codes en les implantant dans le fichier CSS principal. Cela demande parfois un peu de temps et de technique mais le gain en performances est important et ne doit pas être négligé.

Figure 5-9

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

Implantation d'une feuille de style CSS

- Externalisez au maximum les codes en JavaScript comme pour le CSS. En effet, ces codes, souvent conséquents, ralentissent la lecture par les serveurs et les robots. Ils nuisent aussi à la lisibilité du code source et doivent être placés dans des fichiers externes appelés via les balises suivantes :

```
<script href="URL_script/fichier.js" type="application/JavaScript"> </script>
```

D'une manière générale, analysez votre code source pour vous assurez qu'il est propre et structuré, et que les robots n'auront aucun mal à le lire et à comprendre l'entièreté du code lors du crawl.

Beaucoup de spécialistes ne sont pas des techniciens dans l'âme et omettent les facteurs liés au code source mais ils représentent la base du référencement en réalité. N'oublions pas qu'avant de lire les contenus des pages web, les robots lisent tout d'abord des fichiers techniques (`robots.txt` et `.htaccess` notamment, voire `cache.manifest` parfois) et du code. Il faut donc les optimiser au maximum pour éviter des risques éventuels de sanctions ou de mauvaise interprétation du code.

En outre, retenons que tous les efforts fournis pour améliorer l'aspect technique des sites auront également un impact du côté des serveurs et donc pour le confort des utilisateurs, au-delà même du sérieux que cela peut dégager auprès des internautes expérimentés.

Enfin, un code propre assure une certaine stabilité sur les différents navigateurs. Il n'est pas rare de voir des sites bien codés qui n'ont pas besoin de CSS spécifiques pour Internet Explorer, par exemple, sans pour autant être mal affichés sur les anciens navigateurs...

Les URL

Pour améliorer la compréhension de vos URL par les moteurs de recherche, il est important de disposer d'adresses relativement simples, munies de quelques mots-clés, faciles à retenir, etc. Il n'est pas évident d'utiliser une adresse avec des dizaines de chiffres et lettres, par exemple, ou de mémoriser des URL infinies. Pour les utilisateurs, une adresse bien construite sera plus facilement enregistrée et attirera

davantage l'attention le jour où elle se représentera devant eux. Ne négligez jamais l'expérience utilisateur, elle doit être constamment améliorée même si notre objectif est d'optimiser notre référencement.

Figure 5-10

Exact Match Domain quasi impossible à mémoriser

www.bureaudetude-renovation-maison-**bati2000**-construction-86.com/

Attention aussi aux caractères spéciaux, aux accents et aux identifiants de session qui nuisent à la bonne lecture, compréhension et indexation des adresses web. Les URL doivent être claires et représentatives des pages web visitées et concernées.

Dans certains cas, il convient de procéder à de la réécriture d'URL pour obtenir des URL propres et simples à retenir *(URL SEO Friendly)*. Nous avons vu auparavant que la technique n'est pas aisée et demande beaucoup d'efforts, donc il est recommandé de prendre en compte ce facteur dès la création du site et de chaque page. Un site qui est parti du mauvais pied aura bien du mal à rattraper son retard si la réécriture n'a pas été bien pensée voire si elle a été omise lors de l'élaboration du cahier des charges.

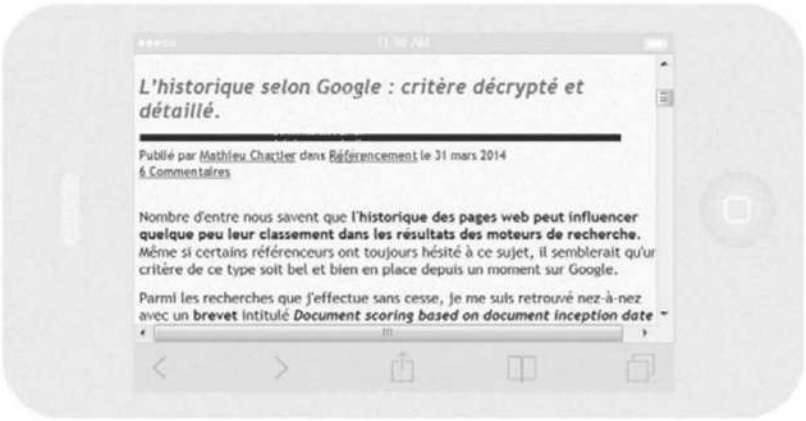
Compatibilité de votre site

Lors d'un audit technique, il faut impérativement vérifier la compatibilité des sites avec les divers navigateurs du marché (Chrome, Firefox, Internet Explorer en plusieurs versions, Safari, Opera voire Konqueror), avec plusieurs systèmes d'exploitation mais aussi sur différents terminaux (PC ou Mac, consoles de jeux connectées, mobiles et tablettes). Ces vérifications ne demandent pas nécessairement beaucoup de temps mais sont importantes.

Pour l'anecdote, il est arrivé plusieurs fois que des boutiques en ligne fonctionnent parfaitement sur Chrome et Firefox mais que des erreurs de code JavaScript bloquent totalement l'accès aux fiches produits et au panier de commande sur différentes versions d'Internet Explorer. Au-delà même de l'impact extrêmement négatif sur les internautes et sur les ventes, cela montre bien que le code peut s'avérer bloquant pour les crawlers…

Figure 5-11

Test du blog.internet-formation.fr sur un iPhone3 au format paysage



Quelques outils pratiques permettent de vérifier la compatibilité des sites web sur des plates-formes diverses :

- Browser Sandbox (source : http://goo.gl/lfQCxo) et browser Shots (source : http://browsershots.org/) pour tester votre site sous divers navigateurs (ou versions) ;
- IETester, Utilu IE Collection (source : http://goo.gl/3Ubj0A), NetRenderer (source : www.netrenderer.com.com) ou Multi IE (source : http://goo.gl/r4YgvK) pour vérifier la compatibilité d'un site sur plusieurs versions d'Internet Explorer ;
- Responsinator (source : www.responsinator.com), Mobilizer (source : http://goo.gl/YxvXaa) ou le puissant logiciel Keynote MITE (source : http://mite.keynote.com) pour vérifier le rendu sur les mobiles et les tablettes voire MobiReady (source : http://ready.mobi) pour analyser la qualité du site sur les supports mobiles.

Les erreurs 404 et leur page dédiée

Il est conseillé d'avoir une page 404 optimisée pour le référencement (avec des liens vers d'autres pages), travaillée pour l'internaute (avec des indications pour ne pas abandonner sa visite et aux couleurs de votre entreprise. La page d'erreur classique a tout ce qu'il faut pour rebuter les visiteurs et faire perdre toute crédibilité à un site web.

Une page d'erreur personnalisée devient au contraire une vraie arme en matière de webmarketing dans laquelle des offres spécifiques peuvent être sporadiquement proposées ou des contenus uniques peuvent être intégrés.

Figure 5-12

Page 404 de Blue Fountain Media et son jeu Pacman

Les liens cassés ou mal remplis impliquent des erreurs 404. Elles sont très néfastes pour le référencement, envoient un mauvais signal aux moteurs et génèrent un inconfort dans la navigation des internautes voire

une frustration dans l'expérience utilisateur. Vous pouvez et vous devez absolument toutes les réparer au risque de pénaliser votre notoriété et votre travail d'optimisation.

Pour savoir si votre site contient des pages d'erreurs 404, vous pouvez lancer un scan avec les logiciels gratuits Xenu (source : http://goo.gl/bHnhfl) et Linkexaminer (source : http://goo.gl/ah6zIX) ou analyser votre *linking* interne via la Google Search Console ou Bing Webmaster Center.

Figure 5-13

Analyse des erreurs de crawl dans un site avec la Google Search Console

Figure 5-14

Liste d'erreurs 404 dans la Google Search Console

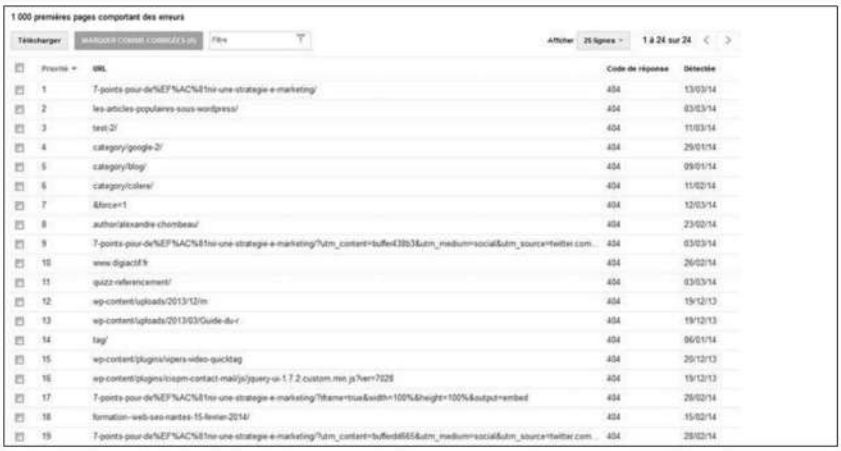


Figure 5-15

Erreurs trouvées et décrites avec Xenu

Hiérarchisation et structure interne

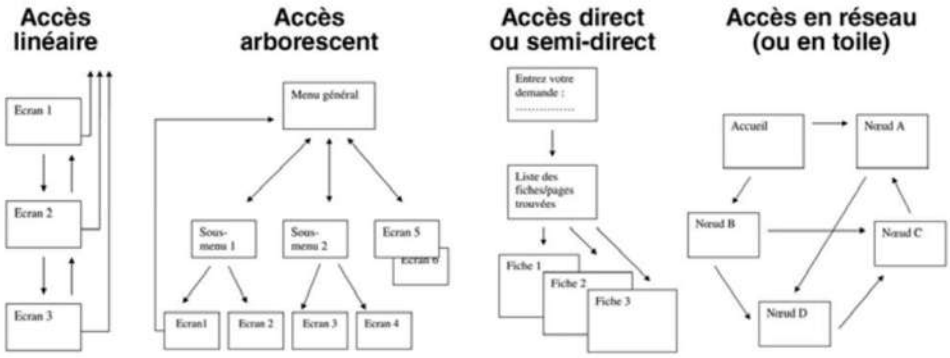
La structure d'un site web est très importante pour éviter aux internautes de se perdre et pour améliorer l'expérience utilisateur. Comme pour les visiteurs, les robots profitent également du soin apporté à la hiérarchisation interne des documents car l'ensemble permet d'améliorer l'indexation et la compréhension globale des sites. Il est primordial d'utiliser des intitulés de répertoires clairs et compréhensibles, de bien classer les documents et de disposer d'une structure optimisée pour l'utilisateur et SEO Friendly.

Pour se faire une idée d'une bonne structure, il faut analyser le nombre de clics nécessaires avant d'arriver à l'information recherchée. Si ce nombre est trop élevé, cela signifie que le niveau de profondeur est trop important, tandis qu'une architecture adaptée impliquera seulement deux ou trois clics pour arriver aux données. Globalement, les pages les plus importantes doivent être accessibles dès la page d'accueil, puis le site déroule des sous-menus en fonction des besoins. Pour les cas exceptionnels pour lesquels les sites web sont vastes et possèdent un haut niveau de profondeur, la présence d'un moteur de recherche interne s'impose pour contourner les petites lacunes de l'architecture interne. Toutefois, cela s'avère pratique pour les utilisateurs mais ne rendra pas service pour autant aux robots d'indexation. C'est pourquoi, la meilleure solution à envisager consiste à retravailler la structure interne.

Si vous souhaitez optimiser la hiérarchisation des documents et l'architecture interne, il faut penser aux concepts de siloing (architecture hiérarchique et ergonomique des contenus dans un site web) ou de Bot Herding (voir chapitre 2, section « Optimiser le Rank Sculpting et le Bot Herding ») mais aussi avoir une bonne connaissance de la typologie des accès sur la Toile. En effet, quel que soit le site web, plusieurs possibilités s'offrent à nous pour créer des structures internes plus ou moins qualitatives. La meilleure solution pour les robots est très nettement l'accès direct (le plus commun de nos jours).

Figure 5-16

Exemples de types d'accès et de structures internes pour des sites web



Attention à la tendance du smooth scrolling

La grande tendance est d'utiliser la technique du *smooth scrolling* pour réaliser des « pages infinies » dynamiques. Dans les faits, cela est représenté par des pages longues coupées en sous-parties accessibles par des liens, à la manière des ancres nommées en HTML, mais avec des effets JavaScript pour avoir un rendu plus actuel. En effet, il suffit de cliquer sur un lien pour être amené dans la zone spécifique de la page. Certes, c'est très pratique et très agréable pour l'utilisateur mais en matière de référencement, tous les mots-clés sont noyés et, surtout, une seule et, unique page est indexée et positionnée, cela peut donc s'avérer catastrophique.

Fil d'Ariane

Le fil d'Ariane permet de visualiser le fil conducteur qui a mené vers chaque page d'un site en présentant des listes de liens. Il améliore l'accessibilité, la navigation et le maillage interne de votre site et permet également à l'utilisateur de savoir à tout moment où il se trouve et comment il pourrait procéder pour remonter d'un ou plusieurs niveaux dans le site.

Le fil d'Ariane est une vraie arme pour l'indexation et le maillage interne mais si votre site est très peu profond, il peut en effet être omis, c'est d'ailleurs un cas courant. Idéalement, retenons qu'il est préférable d'utiliser ce type de procédé pour aider les visiteurs mais aussi pour améliorer considérablement le crawl au sein des pages web lues par les robots.

Figure 5-17

Exemple du fil d'Ariane du site www.service-public.fr

Publicité et pop-ups

Comme nous l'avons évoqué précédemment, les publicités ne sont pas interdites dans les sites web, loin de là, mais il faut veiller à tout prix à ce qu'elles ne soient pas placées en trop grand nombre au-dessus de la ligne de flottaison. Si vous avez besoin d'insérer des publicités dans des pages web, répartissez-les sur l'ensemble de l'espace disponible dans la page web afin d'éviter une sanction causée par Google Page Layout, par exemple.

Attention également aux publicités sous forme de pop-ups, qui nuisent fortement à la bonne expérience utilisateur. Google ne semble pas encore pénaliser ce type d'annonce mais cela ne saurait tarder si les abus continuent en ce sens.

Lors d'un audit SEO, nous devons analyser la présence massive ou réduite de publicités dans les pages, mais aussi les types de publicités employés afin de déterminer le degré de gêne imposé aux moteurs et aux usagers. En général, il est donc important de placer vos publicités et pop-ups avec modération et de façon stratégique.

Logo cliquable

Un des principaux critères mis en avant par la norme ISO 9241-11 en ergonomie est d'avoir un logo cliquable en toute circonstance. En effet, les visiteurs sont habitués à cliquer sur les logos pour revenir vers l'accueil, parfois même en parallèle d'un lien vers la première page d'ailleurs. Les grands ergonomes du web tels que Scapin et Bastien ou encore Jakob Nielsen ont toujours été désireux de faciliter la navigation web en forçant les créateurs de site à opter pour les habitudes des utilisateurs et ils ont certainement raison.

Il faut savoir que Google ne lit qu'une seule fois un lien pointant vers une même page au sein des pages web. Dans ce cas, si vous possédez un lien vers la page d'accueil derrière le logo et un ou deux autres

plus loin dans la page (dans un menu principal et un menu de bas de page, par exemple), c'est le premier qui sera pris en compte. Avouons que nous avons beaucoup plus de chances d'optimiser un lien dans un logo avec des mots-clés forts que le lien classique Accueil dont l'ancre a peu de valeur pour le SEO…

Techniquement, il existe plusieurs méthodes valides pour rendre cliquable un logo, chacun se fera son opinion sur celle qui lui semble la plus adéquate car les avis divergent.

- Mettre une image avec un attribut `alt` optimisé au sein d'un lien (avec parfois un `<h1>` également) comme ceci :

```
<a href="index.html"><img src="super-logo.png" alt="MOTS-CLÉS" /></a>
<h1><a href="index.html"><img src="super-logo.png" alt="MOTS-CLÉS" /></a></h1>
```

- Profiter d'un `<h1>` contenant un lien classique et passer l'image de fond en background via CSS. Cette technique est parfois critiquée mais s'avère pourtant tout aussi intéressante que la première, elle est juste un peu plus technique puisqu'il faut ajouter un `<span>` à l'intérieur du lien pour cacher le texte placé devant l'image :

  – HTML :

```
<h1><a href="index.html"><span>MOTS-CLÉS</span></a></h1>
```

  – CSS :

```
h1 a {
    background:url(super-logo.png) no-repeat;
    width:50px;
    height:50px;
    display:block;
}
h1 a span {
    display:none;
}
```

Contrairement aux idées préconçues, le `display:none;` n'est pas un frein au référencement ici et n'empêche pas les robots de lire les mots-clés. Cette technique devient un problème quand des abus sont réalisés (trop de texte dissimulé), comme le prouve le brevet anti-spamdexing de Google (source : http://goo.gl/ghoMjz). Vous pouvez donc utiliser l'une ou l'autre des techniques selon vos préférences et vos aptitudes techniques, le plus important est de bien remplir le texte du logo ou l'attribut `alt` avec des mots-clés valorisants, au même titre que les balises `<title>` en quelque sorte.

Pour les personnes qui aiment les fonctions en tous genres, voici une courte fonction en PHP qui permet d'afficher dynamiquement le logo en image ou en texte, avec ou sans balise `<h1>`. Il suffit de remplir les quelques paramètres pour que la fonction ajoute automatiquement le code HTML.

```
// addLogo('MOTS CLES', 'URL_ACCUEIL', array(true/false, 'URL_IMAGE'), true/false)
function addLogo($keywords = '', $url = '', $img = array(false, ''), $h1 = true) {
    // Ajout du logo en image ou en texte
    if($img[0] == true) {
```

```
    $logo = '<a href="'.$url.'"><img src="'.$img[1].'" alt="'.$keywords.'" /></a>';
    } else {
    $logo = '<a href="'.$url.'"><span>'.$keywords.'</span></a>';
    }
    // Ajoute automatiquement le <h1> si désiré
    if($h1 == true) {
    $logo = preg_replace('#('.$logo.')#iU', '<h1>$1</h1>', $logo);
    }
    // Retourne le résultat
    echo $logo;
}
// Exemple d'usage avec un lien textuel avec <h1>
addLogo('Mots-clés du titre', 'http://www.site.com');
```

Jouer avec les balises HTML…

La balise `<span>` placée entre les balises de lien peut être remplacée par une balise `<strong>` si vous souhaitez optimiser encore davantage le logo du site. Mais attention à ne pas tomber dans le spamdexing…

Favicon

Un autre critère intéressant est l'icône de favori (souvent appelée « favicon »). Si elle n'influe pas directement sur le référencement, son impact est non négligeable pour l'expérience utilisateur mais aussi en termes d'image de marque et de crédibilité.

La favicon est une toute petite icône qui reprend généralement le visuel du logo du site, qui attire le regard des internautes et influence favorablement le taux de clics (car elle peut être affichée dans plusieurs endroits sur la Toile ou dans des outils).

La favicon peut prendre deux tailles différentes : 16 × 16 ou 32 × 32 pixels. Il peut s'agir d'une icône au sens propre (avec l'extension .ico) ou une petite image carrée dans un autre format. Il convient juste de veiller à modifier le type MIME indiqué dans la balise `<link />` qui permet de l'insérer.

Voici deux exemples de balises `<link />` spécifiques pour ajouter une favicon :

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />
<link rel="shortcut icon" href="favicon.png" type="image/png" />
```

Prendre garde aux compatibilités

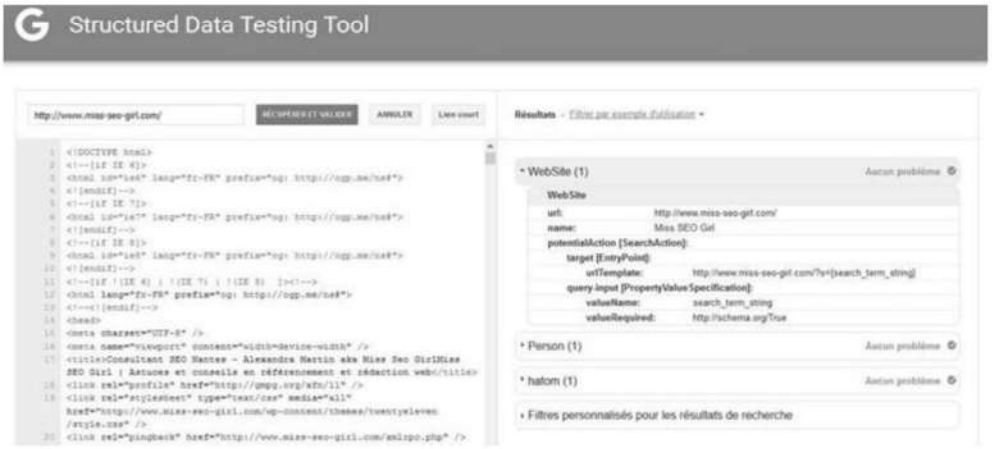
L'extension .ico est compatible sur tous les navigateurs, contrairement à l'extension .png, par exemple, qui ne fonctionne pas sur Internet Explorer.

Rich snippets

Les rich snippets (microdonnées, microformats ou RDF…) sont de plus en plus importants pour le SEO. Non pas que leur impact soit reconnu en matière de positionnement, leur avantage est d'ajouter de la sémantique dans les codes sources, d'améliorer l'accessibilité des sites web mais aussi d'ajouter parfois des indications dans les SERP afin d'être mieux repéré par les visiteurs. N'hésitez pas à les utiliser, Google vous remerciera en quelque sorte... Les rich snippets apportent plus de visibilité dans les SERP et donc éventuellement plus de trafic. Dans le pire des cas, c'est le taux de clics qui doit être affecté, ce n'est donc pas un mal. Nous avons détaillé ce point au chapitre 1 (voir sections « Maîtriser les rich snippets » et « Outils d'aide au balisage des extraits de code enrichis »), nous n'allons donc pas revenir dessus ici. Mais rappelez-vous que vous pouvez tester vos codes sémantiques via l'outil de test des données structurées (source : http://goo.gl/yUNdPM).

Figure 5-18

Test des rich snippets avec l'outil dédié



Hébergements et serveurs

Hébergement web

Le choix de l'hébergement est primordial, que ce soit pour le SEO comme pour la survie du site dans le temps. En effet, selon le projet, il convient d'adapter l'hébergement au trafic réel afin de ne pas avoir de problèmes de latence, de sécurité ou autres.

Par exemple, un serveur mutualisé de petite ou moyenne gamme supportera difficilement un projet e-commerce avec plusieurs milliers de visiteurs par jour. Dans ce cas, il vaudrait mieux opter pour un serveur dédié ou une solution haute disponibilité. En contrepartie, un site de présentation de quelques pages ne nécessite pas du tout l'usage d'un serveur dédié, bien plus coûteux, alors que le nombre de visiteurs reste relativement faible en général. Chaque site doit s'adapter à la réalité, il ne sert à rien d'avoir une machine de course pour un site de dix pages, mais si le trafic est important, des solutions plus puissantes conviendront mieux.

Sur le plan fonctionnel et technique, il ne fait aucun doute que les serveurs dédiés sont bien meilleurs, sans oublier les solutions *cloud* qui permettent d'améliorer encore certains chargements de fichiers. Un serveur dédié apporte souvent plus de sécurité, de souplesse (options paramétrables contrairement aux hébergements mutualisés) et de stabilité, mais ces avantages ont un prix donc il faut toujours peser le pour et le contre.

En termes de référencement, il est plutôt conseillé de se diriger vers des solutions dédiées car elles améliorent les performances et évitent les risques de mutualisation avec des sites de spammeurs. Cela ne signifie pas pour autant que des sites hébergés sur des serveurs mutualisés ne peuvent pas être bien référencés, cela représente tout de même la très large majorité du marché et les résultats sont aussi au rendez-vous…

Localisation du serveur

La localisation du serveur correspond au lieu où se trouve le datacenter qui accueille votre machine, c'est-à-dire la « salle des machines » qui regroupe tous les ordinateurs et serveurs distants. En fonction du public visé et de la portée du site, il est préférable de choisir un serveur situé dans le pays principal concerné.

Généralement, le choix de la bonne localisation améliore quelque peu la crédibilité du site vis-à-vis des visiteurs et la prise en compte par des moteurs de recherche locaux (google.fr, par exemple). Par conséquent, les résultats affichés dans les SERP sont liés au bon pays et permettent de toucher encore mieux le public cible.

Toutefois, il est parfois intéressant de jouer avec des adresses IP situées dans d'autres pays si l'extension du nom de domaine permet déjà de rattacher un site à un pays donné, la localisation du serveur a essentiellement un rôle pour les sites qui portent des extensions tels que les .com, .net, .org, .eu…

Emplacement du domaine

Sur les serveurs mutualisés, les sites web se comptent parfois par centaines voire par milliers et ils possèdent tous des adresses IP similaires (ou presque), ce qui signifie que sur une même machine, des sites valorisés ou de piètre qualité peuvent être confrontés. Indirectement, ce point risque peut-être d'amoindrir la confiance accordée par les moteurs de recherche.

Dans les faits, cela n'a jamais été vérifié et confirmé à 100 % par les diverses firmes, mais le doute est permis. Cependant, nous pouvons penser que les moteurs de recherche sont largement assez puissants pour distinguer des sites différents portant des adresses IP équivalentes sur des serveurs, notamment grâce aux noms de domaines attribués à chacun. Il est fort probable que les sites soient reconnus par leur nom de domaine et qu'en fonction de cela, les sites spammy soient sanctionnés lourdement sans que cela affecte les autres sites de l'hébergement mutualisé.

Si toutefois vous craignez des sanctions, optez pour un serveur dédié ou analysez les sites hébergés sur le même serveur que le vôtre. Pour ce faire, les outils suivants sont bien pratiques : ewhois.com, spyonweb.com ou Yougetsignal.com (source : http://goo.gl/IIf7CJ)...

Figure 5-19

Liste de sites hébergés sur un même serveur

Temps de chargement du site

Le chargement du site correspond à la vitesse d'affichage des contenus des pages web. En général, plus un site est lent, plus l'internaute est susceptible de changer de site et de ne plus jamais revenir. Certaines études menées dans le domaine de l'e-commerce ont même tenté de démontrer que des sites trop lents pouvaient abaisser les ventes d'un tiers voire plus par seconde écoulée au chargement des pages, il convient donc d'être très vigilant sur ce point, au-delà même des aspects SEO.

De plus, les moteurs devront consacrer plus de temps pour lire, indexer et valoriser les sites web lents, ce qui n'est pas leur objectif premier. Il convient donc de minimiser au maximum ce temps de chargement pour les robots et surtout les internautes, notamment pour les personnes qui possèdent une connexion bas débit ou qui utilisent des terminaux mobiles connectés en Edge, par exemple.

Pour obtenir une estimation du temps de chargement des pages web, vous pouvez utiliser des outils en ligne tels que :

- Pingdom : http://goo.gl/9mwb23 ;
- GTmetrix : www.gtmetrix.com ;
- Neustar : http://goo.gl/sLW40M ;
- Webwait : www.webwait.com ;
- Load Impact : www.loadImpact.com ;
- Web Page Test : www.webpagetest.org ;
- Website Pulse : http://goo.gl/HkgZ3Z.

Vous pouvez aussi vous référer à l'outil PageSpeed Insights de Google (source : http://goo.gl/a5BZ75).

Figure 5-20

Temps de réponse du serveur avec Website Pulse

Website test results

URL tested:	http://www.miss-seo-girl.com
Test performed from:	New York, NY
Test performed at:	2014-03-16 16:46:07 (GMT +00:00)
Resolved As:	82.165.186.152
Status:	OK
Response Time:	5.643 sec
DNS:	5.121 sec
Connect:	0.093 sec
Redirect:	0.000 sec
First byte:	0.148 sec
Last byte:	0.281 sec
Size:	89620 bytes

Figure 5-21

Mesure du temps de chargement avec Pingdom

Gérer le sous-domaine www

Par défaut, un même site est généralement accessible à partir du domaine seul domaine.com et du sous-domaine www.domaine.com. Le problème de ce genre de pratique est que les moteurs peuvent considérer deux sites distincts si l'optimisation interne est mal adaptée. Il faut donc utiliser soit l'un ou soit l'autre et le faire comprendre aux moteurs de recherche.

Trois problèmes majeurs ressortent de ce doublon d'adresse.

- L'affichage aléatoire des URL avec ou sans www dans les résultats de recherche, ce qui n'est pas toujours un gage de sérieux et de confort pour les internautes. Ce problème peut être réglé via la Google Search Console, par exemple, en définissant un domaine favori dans la section Paramètres du site (figure 5-22).

Figure 5-22

Définir un domaine favori dans la Google Search Console pour améliorer l'affichage dans les SERP

- Des doublons se multiplient et génèrent des contenus dupliqués sur le site (problème de DUST), notamment pour la page d'accueil comme le montrent ces exemples :
  - http://www.monsite.com et http://monsite.com ;
  - http://www.monsite.com/index.html ;
  - http://monsite.com/index.html.
- Division du ranking (PageRank, BrowseRank, etc.) pour les pages concernées à cause des doublons réalisés. Si des liens pointent vers les noms de domaines avec ou sans les www, cela impacte nécessairement la transmission du jus de liens.

Il existe aussi des hébergeurs qui ne mettent pas en place par défaut l'usage du sous-domaine avec www. Si cela n'est pas forcément un mal au premier abord, rappelez-vous que les internautes ont pris des habitudes depuis l'arrivée du Web et il est fortement recommandé en termes d'ergonomie et d'accessibilité de proposer l'accès au site par ce biais en plus de l'accès classique. Cela est très important, il faudra veiller à ne pas mélanger les URL comme nous l'avons indiqué précédemment. Pour contrer ces problématiques, l'usage de redirections permanentes (redirections 301) reste la meilleure solution et la plus simple à mettre en œuvre. Vous pouvez également utiliser la réécriture d'URL pour ce cas, elle s'avère même souvent plus intéressante pour favoriser le sous-domaine portant les www. Dans les deux cas, il suffit d'ajouter un fichier `.htaccess` contenant quelques lignes de code.

```
# Redirection permanente
RedirectPermanent / http://www.domaine.fr/

# Réécriture d'un domaine sans les www vers un site portant les www
Options +FollowSymlinks
RewriteEngine on
RewriteCond %{HTTP_HOST} ^domaine.fr$
RewriteRule ^(.*)http://www.domaine.fr/$1 [QSA,L,R=301]

# Réécriture d'un site portant les www vers un domaine sans les www
Options +FollowSymlinks
RewriteEngine on
RewriteCond %{HTTP_HOST} ^www.domaine.fr$
RewriteRule ^(.*)http://domaine.fr/$1 [QSA,L,R=301]
```

Si les fichiers `.htaccess` vous semblent complexes à utiliser, vous pouvez aussi opter pour une simple redirection 301 via un script en PHP placé dans la page d'accueil du site, comme ici :

```
<?php
header("Status: 301 Moved Permanently", false, 301);
header("Location: http://www.domaine.fr");
exit();
?>
```

Audit de contenu

L'audit de contenu constitue la seconde partie majeure d'un audit SEO. Cette phase regroupe l'étude de l'ensemble des critères en rapport avec le contenu à forte valeur ajoutée. De nos jours, il est extrêmement difficile de positionner un site dénué de contenu ou possédant un contenu de faible qualité, il est important de proposer du texte riche et soigné dans la majorité des cas.

Par conséquent, si vous souhaitez plaire aux robots, apporter un minimum d'informations sur vos sites web et fidéliser votre communauté, il faut rédiger proprement, qualitativement et quantitativement en respectant les critères éditoriaux. Votre stratégie de rédaction doit être réfléchie à long terme et être adaptée pour chaque page de vos sites web, quel que soit le type de site réalisé (blog, e-commerce, site de présentation, outil en ligne...).

La balise <title>

La balise `<title>` demeure la plus importante en termes de contenu à optimiser. Elle fournit à la fois une description de la page web, des mots-clés aux moteurs de recherche mais incite aussi les internautes à cliquer sur votre lien dans les SERP. Comme nous l'avons déjà dit auparavant, il faut absolument travailler le contenu de cette balise page après page.

Dans cette étape de l'audit, notre rôle est de vérifier la présence de cette balise dans le `<head>`, et non de la balise meta `title` qui n'a aucun impact direct sur le positionnement (l'amalgame est souvent fait, il vaut mieux éviter de se tromper). Ensuite, il faut surveiller les rapports sémantiques entre les divers `<title>` et les pages concernées et enfin, veiller à ce que ces contenus soient assez optimisés pour améliorer le positionnement des pages. Ne négligez pas cette étape importante. Elle peut prendre du temps et demander parfois une veille concurrentielle voire une analyse des mots-clés utilisés dans les balises `<title>`.

La balise meta description

La balise meta `description` n'influe pas directement sur le positionnement d'un site web, mais elle est très importante pour les internautes car son contenu est affiché en dessous du lien proposé dans les résultats de recherche. La description sert donc en quelque sorte d'appât, elle doit correspondre à la thématique de la page et correspondre au contenu de la balise `<title>` de la page concernée, en étant plus détaillée bien entendu.

Argumentaire marketing ou commercial, la balise meta `description` doit inciter au clic, voire convertir le visiteur, mais également apporter une valeur ajoutée par rapport à la concurrence. Comme pour les

titres de pages, l'audit SEO permet de vérifier leur présence et leur qualité d'optimisation en fonction des pages. N'omettons pas aussi de vérifier l'emplacement de la balise meta `description` dans le `<head>`. Elle est idéalement positionnée juste après les balises `<title>...</title>` et doit surtout être placée avant les balises `<script>...</script>` pour être validée à 100 % par le W3C.

L'utilisation des titres internes avec <hn> (<h1> à <h6>)

Les titres internes peuvent être créés en HTML sur six niveaux grâce aux balises `<h1>` à `<h6>`, souvent appelées `<hn>` par commodité. Elles organisent et hiérarchisent les contenus en titres, sous-titres...

Il est important d'avoir une bonne organisation dans les contenus proposés et ces balises sont parfaites pour cela. Ces indications permettent de faire comprendre aux internautes et aux moteurs de recherche la structure interne des contenus et d'améliorer l'expérience utilisateur en termes de lisibilité.

L'audit de contenu vise à vérifier plusieurs aspects liés aux balises `<hn>`.

- Les balises sont-elles présentes ou non dans les pages ?
- L'ordre hiérarchique des balises est-il respecté (`<h1>`, puis `<h2>`, puis `<h3>`...) ?
- Existe-t-il des omissions de balises de titres ? En effet, il ne serait pas intéressant de passer directement d'un `<h1>` à un `<h4>` par exemple, il faut respecter la logique sémantique.
- L'usage de ces balises peut-il être considéré comme du spam ? Par exemple, si une page concentre beaucoup trop de balises `<h1>`, les moteurs de recherche peuvent estimer qu'il s'agit de suroptimisation des contenus au point de sanctionner la page.

Nombre de balises <h1> autorisées en HTML 5

En HTML 5, le W3C autorise l'usage d'une balise `<h1>` pour chaque élément `<article>` présent dans une page, ce qui signifie que nombre de `<h1>` peuvent se retrouver au sein d'une même page. Google a appuyé le projet HTML 5 mais n'a pas encore confirmé une éventuelle tolérance dans ce cas, il faut donc rester vigilant jusqu'à nouvel ordre et se contenter d'un minimum de titres de premier niveau...

Sémantique et structure HTML

Étudier la sémantique et la structure HTML est primordial car les balises jouent un vrai rôle pour le positionnement comme nous venons de le voir. Il est important de bien vérifier si nos balises sont fermées, si elles respectent le type de document choisi ou encore si nous appliquons l'ordre logique du code. Par exemple, placer un `<h1>` après un `<h4>`, mettre une balise `<p>` au sein d'un titre de page et intégrer des balises block dans des balises inline, etc., sont autant de pratiques qui ne respectent pas la logique sémantique de l'HTML.

Nous allons étudier une fonction PHP qui permet d'analyser la structure interne des pages web afin de faire ressortir rapidement la hiérarchie sémantique et les balises HTML. Ainsi, nous pourrons rapidement vérifier si notre structure est correcte mais aussi si nos balises de structure sont fermées.

Il suffit de créer un fichier intitulé par exemple `hierarchie.php` dans lequel nous intégrons la fonction d'analyse suivante :

```
<?php
// Fonction de vérification de la structure
// $titlemeta sert à afficher ou non les <title> et <meta> (true/false)
// $fermantes permet d'afficher ou non les balises fermantes (true/false)
function verifStructure($page, $tags, $titlemeta = true, $fermantes = true) {
// Ouverture du fichier en lecture seule
$ouverture = fopen($page,'r');

// Si l'ouverture fonctionne, on enregistre tout le contenu
if($ouverture) {
    while (!feof($ouverture)) {
        $texteTotal[] = fgets($ouverture);
    }
    $contenu = implode('', $texteTotal);

    // Fermeture du fichier
    fclose($ouverture);
}

// Conditionne l'affichage des balises fermantes ou non
if($fermantes == true) {
    $close = '[\/]?';
} else {
    $close = '';
}

// Analyse complète du contenu
preg_match_all("#(<".$close.".*>)#iU", $contenu, $tabTags);

// On affiche les balises structurelles
foreach($tabTags[0] as $balise) {
    // Si nous voulons vérifier la présence des titres et meta
    if($titlemeta == true){
        if(preg_match("#<".$close."(title|meta)#iU", $balise)) {
            echo "<strong>".htmlspecialchars($balise)."</strong>";
            echo "<br/>\n";
        }
    }
    // Analyse les balises structurelles
    foreach($tags as $tag) {
        if(preg_match("#<".$close.$tag."#iU", $balise)) {
            if(preg_match("#<".$close."(p|h[1-6])#iU", $balise)) {
                echo '<div style="text-indent:1em; font-size:.9em">'.
                htmlspecialchars($balise)."</div>\n";
            } else {
                echo htmlspecialchars($balise);
                echo "<br/>\n";
```

```
                }
            }
        }
    }
exit();
}
?>
```

Une fois le fichier créé, il suffit de lancer la fonction dans les fichiers que l'on souhaite analyser. Attention cependant, ces fichiers doivent porter l'extension PHP, la fonction ne pourra donc pas s'appliquer aux fichiers .html. Dans ce cas, il faudra renommer les fichiers HTML en `nom-fichier.php` afin d'effectuer le test.

Nous devons ajouter en début de fichier les lignes suivantes pour que cela fonctionne, en sachant que la liste des balises analysées et les paramètres peuvent être modifiés.

```
<?php
// Inclusion de la fonction
include_once('hierarchie.php');
// Liste des balises structurelles à lire (xHTML et HTML 5 ici)
$tags = array('div', 'p', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'header', 'footer',
'aside', 'nav', 'section', 'article');
// Lancement de la fonction
verifStructure(basename(__FILE__), $tags, true, true);
?>
```

La fonction `verifStructure()` prend quatre paramètres dont deux optionnels :

- le premier argument est le fichier à lire par la fonction, il suffit d'écrire `basename(__FILE__)` pour que la fonction lise le fichier en cours de lecture ;
- le deuxième paramètre correspond à un tableau de données PHP qui comprend toutes les balises PHP que nous souhaitons analyser. Dans notre exemple, toutes les balises de structure xHTML et HTML 5 sont vérifiées, nous ne devrions donc pas avoir à modifier les données ;
- le troisième argument est un booléen qui permet de vérifier la présence des balises `<title>` et `<meta/>` (valeur `true`) ou non (valeur `false`) ;
- la dernière option est également un booléen qui permet d'afficher les balises fermantes (`true`) ou non (`false`).

Une fois le paramétrage effectué, il faut lancer la page à tester. Le résultat sera un affichage spécifique de la structure sémantique et technique de la page web (figure 5-23).

Figure 5-23

Arborescence structurelle HTML avec la fonction verifStructure()

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>
</title>
<header>
  <h1>
  </h1>
  <h2>
  </h2>
</header>
<nav>
  <h2>
  </h2>
</nav>
<aside>
  <p>
  </p>
</aside>
<section>
<article>
  <h2>
  </h2>
  <p>
  </p>
</article>
<article>
  <h2>
  </h2>
  <p>
  </p>
</article>
<section>
<footer>
  <p>
  </p>
</footer>
```

Vérifier la structure HTML et le CSS

Il est important de bien vérifier la structure HTML, mais n'oubliez pas de vérifier la compatibilité mobile (ou l'ergonomie mobile comme indiqué dans la Google Search Console). Ce critère SEO datant du 21 avril 2015 n'est pas à exclure ; il convient de vérifier que les pages web sont bien structurées et fonctionnelles sur les supports mobiles également...

Les contenus textuels

Les articles ou autres contenus apportent généralement la valeur ajoutée dans un site et constituent le point fondamental pour plaire aux internautes, fidéliser des visiteurs et augmenter son taux de conversion (ventes, contacts, inscriptions…).

Il convient donc de toujours se remettre en question pour améliorer au maximum les contenus internes qui attirent les visiteurs en quête d'informations. Vos articles doivent être soignés, structurés, de qualité et apporter une réelle valeur ajoutée pour l'internaute.

Gunning Fog Index (source : http://gunning-fog-index.com) est un outil qui va vous permettre d'avoir une idée de la qualité, de la lisibilité et de la compréhension des contenus. Lors de la rédaction, nous n'employons pas les mêmes termes s'il s'agit d'un public d'adolescents, d'adultes ou de personnes spécialisées. Le service fournit un indice pour vous orienter dans la rédaction web afin que les textes s'approchent au plus près des attentes de votre cible.

L'indice Gunning Fog, issu du nom de son inventeur Robert Gunning, est un calcul mathématique qui correspond au nombre d'années de scolarité nécessaires pour réussir à lire et à comprendre un texte donné sans difficulté. De facto, un résultat élevé signifie qu'il s'agit d'un texte difficile pour un certain public, il faudra donc veiller à coller au maximum à votre cible.

La formule mathématique peut se décrypter ainsi : additionner le pourcentage des mots de plus de trois syllabes et le nombre moyen de mots par phrase, puis multiplier ce résultat par l'indice 0.4.

Figure 5-24

Formule de l'indice Gunning Fog

$$0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right)+100\left(\frac{\text{complex words}}{\text{words}}\right)\right]$$

La figure5-25 montre comment interpréter les scores.

Figure 5-25

Table de l'indice Gunning Fog (source : Sébastien Billard)

GUNNING FOG	TYPE D'ÉCRIT
8-9	Littérature junior et ado, Paris Match, Elle
10-11	Télérama, Libération
14-15	Marcel Proust, Le Monde Diplomatique, L'Express
16-17	Rapports parlementaires
17-18	Article universitaire d'Olivier Ertzscheid
22 et plus	Directives européennes

À titre d'exemple, un hebdomadaire en kiosque est généralement d'un indice de 10-11 alors que les ouvrages de littérature ont un indice 14-15, et les ouvrages professionnels, universitaires ou textes légaux ont des indices entre 16 et 22, voire plus.

Figure 5-26

Indice Gunning Fog de www.miss-seo-girl.com

Il faut idéalement écrire un minimum de signes par page pour que les textes aient un impact fort en matière de SEO mais aussi utiliser un vocable compréhensible à l'image de l'indice Gunning Fog. L'audit permet de tout analyser et de vérifier la qualité des contenus, (longueur et optimisation en rapport avec le thème traité).

Choix et utilisation des mots-clés

L'audit lexical permet de faire ressortir un ensemble de mots et expressions clés liés à votre activité et qui sont saisis par les internautes sur les moteurs de recherche.

Les mots-clés doivent être utilisés au niveau des titres et descriptions de pages mais également au niveau de la structure sémantique (balises `<h1>`, `<h2>`, `<h3>`...) et du contenu (balises `<strong>`, attributs `alt`...).

Enfin, l'analyse textuelle doit aussi vérifier la densité des mots-clés dans les pages. Non pas que ce facteur ait un rôle pour le positionnement, l'objectif est ici de vérifier si des mots sont trop utilisés dans les pages au point de risquer des sanctions (ou d'être ignorés comme sur Google parfois en cas de spamdexing).

Longue traîne

La longue traîne *(Long Tail)* qualifie l'ensemble des mots et expressions clés qui sont recherchés en proportion réduite mais dont la somme des recherches peut dépasser celles des mots et expressions clés les plus recherchés.

En référencement, il ne faut pas se limiter à une liste précise de mots-clés mais proposer aussi ceux qui sont au cœur de l'activité principale et qui semblent pourtant secondaires. Le positionnement sur des mots génériques doit être utilisé avec parcimonie et ne peut pas suffire pour obtenir de bons résultats, notamment face à la concurrence montante des dernières années.

L'investissement n'est pas le même sur des mots-clés génériques que sur ceux de seconde classe, il est souvent plus intéressant de mixer les deux types de termes clés pour obtenir des résultats satisfaisants. L'audit de contenu doit justement mettre en exergue l'usage réfléchi et optimisé de la longue traîne (au contraire d'une longue traîne non travaillée et présente par défaut dans les pages).

Contenu dupliqué

S'il existe une chose que Google n'aime absolument pas, ce sont bien les contenus dupliqués, aussi bien dans les pages internes d'un site en doublon que plusieurs sites web différents composés de contenus plagiés. On parle alors de contenu dupliqué interne *(on site)* et externe *(off site)*.

Cela est d'autant plus vrai puisqu'en 2011, Google a mis en place le filtre Panda pour lutter contre ces contenus dupliqués et/ou de mauvaise qualité. Il est d'ailleurs actuellement greffé au processus d'indexation du moteur, toutes les pages visitées par les robots sont donc soumises au test anti-plagiat d'une certane manière.

Pour rappel, il convient d'utiliser avec maîtrise les attributs `rel="canonical"` (ou `rel="prev"` et `rel="suiv"`), des redirections via un fichier `.htaccess` ou des techniques de déréférencement (balises meta `robots`, fichier `robots.txt`...) pour contrer ce type de problèmes au maximum.

Pour vérifier le contenu dupliqué off site (ou le plagiat en d'autres termes), vous pouvez utiliser des outils tels que :

- Copyscape (source : http://goo.gl/6u1FPD) ;
- Plagiarism Checker (source : http://goo.gl/Ni3Wxp) ;
- Positeo, l'outil d'analyse de Dustball (source : http://goo.gl/WlSQHl) ;
- Plagium (source : http://www.plagium.com) ;
- KillDC de Linkomatic (source : http://goo.gl/oxEZz8).

Vous pouvez aussi tout simplement copier une phrase de votre contenu et la rechercher via les moteurs de recherche.

Prendre garde au spinning, aux parseurs et scrapeurs…

Attention aux contenus syndiqués ou générés par des processus considérés comme du Black Hat SEO. Nous pouvons notamment mentionner le *content spinning* (ou *spin*) dont les contenus sont souvent remplis de fautes d'orthographe et de grammaire, mais aussi la génération automatique des contenus à l'aide de parseurs PHP et XML.

Figure 5-27

Vérification des contenus plagiés avec l'outil Plagium

Les contenus des médias

Pour illustrer vos articles, vous allez souvent faire appel à des médias (images, vidéos, fichiers PDF…). Soignez ces médias avec des titres, des descriptions uniques et intéressantes pour les moteurs de recherche et les internautes. Les légendes et textes alternatifs (attribut `alt` des images) ainsi que les contenus environnants ont un intérêt dans votre site.

Depuis HTML 5, il existe les balises `<figure>…</figure>` et `<figcaption>…</figcaption>` qui permettent d'encadrer un média dans le code et de lui ajouter une légende textuelle en plus du texte de remplacement. Ces balises ne sont pas compatibles sur les anciens navigateurs mais il existe des polyfills pour régler ce problème, en sachant que dans le pire des cas, la légende est affichée comme du texte classique s'il existe une incompatibilité. Si rien ne confirme encore que ces balises jouent un rôle en matière de référencement, nous pouvons penser que cela sera peut-être le cas un jour. Mais surtout, elles permettent d'ajouter un vrai contenu en relation avec les médias affichés et d'améliorer l'expérience et l'efficience pour les visiteurs.

Voici comment intégrer les nouvelles balises dans le code :

```
<figure>
    <img src="URL_IMAGE" alt="Logo SEO"/>
    <figcaption>Logo SEO</figcaption>
</figure>
```

Figure 5-28

Rendu des légendes HTML 5 alignées par défaut

Fig.1 - Logo SEO.

Prenez garde aux poids des fichiers multimédias. S'ils sont trop lourds, cela influencera négativement le temps de chargement des pages et pourra rebuter les robots et les visiteurs. L'audit doit donc permettre de répondre à toutes ces interrogations facilement. Des outils tels que Firebug ou Web Developper sur Firefox, par exemple, permettent d'analyser rapidement le poids des fichiers multimédias, au même titre que les outils déjà présentés précédemment tels que GTMetrix et Google PageSpeed Insights.

La fréquence de mise à jour

Lorsque l'on administre un site web, la fréquence de mise à jour des contenus est très importante. En effet, les robots d'indexation passent plus souvent en fonction du nombre de mises à jour effectuées sur le site.

Cela influence également le FreshRank mis en place dès 2010 chez Google (source : http://goo.gl/swi1LJ). Cet algorithme souvent méconnu a pourtant un rôle à jouer dans les sites web. En effet, il permet de déterminer un score de fraîcheur des contenus au fur et à mesure des passages du robot selon plusieurs facteurs :

- le degré de mise à jour des contenus ;
- l'étude des requêtes les plus fréquemment tapées qui permettent de trouver les pages web (plus une page répond à des requêtes différentes, plus elle est considérée comme une page d'actualité pour Google) ;
- le nombre de liens entrants obtenus par les pages et leur fréquence d'apparition ;
- l'analyse des mises à jour des ancres pointant vers les pages ;
- la mesure du trafic obtenu par une page (si une page voit son trafic se réduire de plus en plus, cela est signe d'obsolescence et le score du FreshRank sera dévalué) ;
- l'étude des liens favoris *(bookmarks)* afin de déterminer le niveau d'intérêt des pages.

D'autres critères moins importants sont également mesurés pour le FreshRank et montrent à quel point un site mis à jour fréquemment peut avoir un réel impact sur le positionnement. Qui plus est, une bonne fréquence de mise à jour d'un site améliore considérablement son activité, sa vie et influence par conséquent sa notoriété, son trafic et sa crédibilité sur la durée.

Le maillage interne

Internet est le résultat de millions de pages s'interconnectant entre elles grâce aux liens. Votre site possède lui aussi un environnement de liens spécifique, il faut donc optimiser son maillage interne et sa stratégie de linking afin d'optimiser sa présence sur le Web. Les points suivants sont importants à analyser lors d'un audit SEO :

- maillage interne du site : il s'agit ici de connecter plusieurs pages entre elles grâce aux liens hypertextes. Au fur et à mesure que vous créez des liens pour envoyer vers une autre page de votre site, le maillage interne commence à se développer. Il est primordial de l'optimiser pour les robots d'indexation mais également pour les utilisateurs qui visitent et naviguent dans le site. Les pages les plus importantes de votre site (les plus stratégiques pour votre activité) doivent bénéficier d'un maximum de liens internes pour être encore plus valorisées auprès des moteurs de recherche ;
- nombre de liens par page : si vous avez trop de liens dans des pages, il se peut que les moteurs de recherche réduisent leur valeur ou les considèrent tout simplement comme du spam. Il faut donc limiter et minimiser le nombre de liens par page et vérifier leur pertinence ;
- gestion des follow/nofollow : il n'existe pas de ratio parfait, il convient juste d'avoir un profil de liens entrants qui semble naturel avec un mélange des deux types de liens hypertextes. Dans le maillage interne, les nofollow se font de plus en plus rares au sein des sites web en toute logique puisque Google a précisé qu'ils devaient essentiellement être utilisés dans le cas des liens pointant vers des pages d'administration et de connexion ou dans les commentaires de blogs, par exemple. Si trop de nofollow sont présents en interne, il faudra se poser la question de leur pertinence réelle… ;
- maîtrise des ancres de liens : ne répétez pas trop les mots-clés dans les ancres, essayez de diversifier vos ancres afin de créer un profil naturel de liens ;
- nombre de liens externes : lorsque vous mettez en place des liens externes pointant vers d'autres sites, il faut vous assurer de leur bonne qualité, utilité et quantité. En effet, un trop-plein de liens externes dilue le jus de liens assigné à la page cible et la popularité que vous lui transmettez. De plus, ceci peut être assimilé à du spam de liens si vos liens sont de faible qualité et trop nombreux.

Attention donc à vos contenus, ils doivent être bien rédigés, structurés et illustrés ! Prenez garde de ne pas suroptimiser en voulant trop bien faire (*keyword stuffing* par exemple) et pensez à l'essentiel : écrivez pour les internautes, pensez à apporter de la valeur ajoutée et à utiliser toutes les techniques d'optimisation de contenu avec modération.

Audit de popularité

Terminons notre audit SEO avec l'analyse de la popularité. Par « popularité », on entend aussi bien l'étude des backlinks (profil des liens) que la visibilité et la notoriété d'un site (e-réputation) ou d'une personne *(personal branding)* sur les réseaux sociaux.

Analyse des backlinks

Les backlinks correspondent à tous les liens entrants en provenance de divers sites. Chaque lien est considéré comme un « vote » pour les moteurs de recherche comme nous l'avons évoqué auparavant.

Plus une page reçoit de backlinks de qualité, plus elle est considérée comme populaire dans les résultats de recherche (notamment Google et Bing qui utilisent ce type de procédé). La difficulté est de qualifier ce qu'est un « bon lien », notamment depuis la mise à jour Google Penguin qui, rappelons-le, détecte et sanctionne les liens de mauvaise qualité, les réseaux de liens ou toute autre technique de manipulation du moteur (paidlinks, acquisition massive et rapide de liens...).

Il faut désormais privilégier la qualité des liens à leur quantité pour ne pas être pénalisé, voire mis sur la liste noire pour cause de spam.

Gardez en tête qu'un bon netlinking se construit sur le long terme et que cette popularité n'a de sens que si le profil des liens est naturel et assez réaliste. En effet, quel est l'intérêt de multiplier les liens entrants s'ils sont tous de piètre qualité ? Si aucun internaute ne nous trouve, cela signifie que tout ce travail n'est effectué qu'à des fins de référencement, mais un bon PageRank ne suffit pas si les contenus ne sont pas travaillés par exemple. Il faut donc avant tout penser aux utilisateurs (ou tout du moins au trafic que l'on veut gagner) et obtenir des liens valorisants à la fois pour le référencement et pour augmenter le nombre de visiteurs du site.

L'audit de liens permet de dessiner un profil de liens, afin de savoir si vous êtes « naturel » aux yeux des moteurs de recherche. Pour ce faire, il faut analyser plusieurs critères.

- Les sources des liens entrants :
  - Les backlinks proviennent-ils de sites de qualité ?
  - Combien de domaines différents ont mis en place des liens pointant vers votre site ?
- Les pages affectées par des liens entrants : le plus souvent il s'agit de la page d'accueil, mais une bonne stratégie de netlinking consiste à obtenir des liens vers les pages profondes les plus intéressantes afin d'optimiser tout le site et pas uniquement la première page...
- Le type de lien utilisé (texte, image...) : si vous pouvez choisir, évitez les liens en Flash, JavaScript ou passant des redirections. Il faut essentiellement obtenir des liens en dur (en HTML, cela se traduit par les classiques balises `<a href="URL">ANCRE</a>`).
- L'emplacement des liens dans la page : pour rappel, il faut veiller à ce que les liens soient placés dans des zones favorables à la propagation du jus de liens. Souvent, les échanges de liens se font par le biais d'une page Liens ou Partenaires, ce qui a peu d'impact et de valeur en réalité. Dans d'autres cas, les liens sont positionnés dans le pied de page (le cas le plus fréquent) mais idéalement, c'est dans les contenus que les liens apportent le plus de valeur au site ciblé (en prenant garde de ne pas tomber dans des sites proposant de faux communiqués de presse qui font croire que les liens ont plus de valeur en étant inséré dans des articles sans valeur ajoutée...).

Figure 5-29

Analyse des domaines proposant des backlinks

Figure 5-30

Diversité de domaines qui fournissent des backlinks

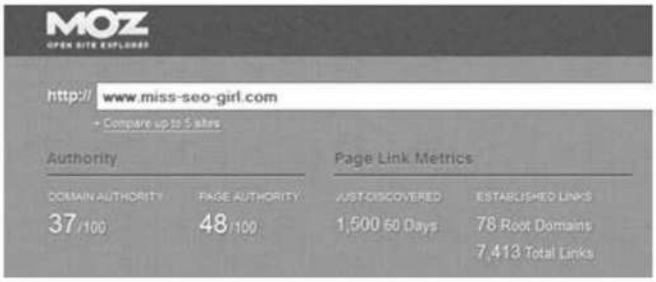


Figure 5-31

Analyse des pages les plus ciblées par des backlinks

- L'ancre des liens : attention aux ancres trop similaires et très optimisées, elles ne sont pas toujours un gage de réussite et peuvent même entraîner des pénalités par Google Penguin, par exemple. Avoir plusieurs ancres différenciées bien réparties dans la masse globale des backlinks assure de bien meilleurs résultats.

Figure 5-32

Analyse des ancres de liens pour un site web

Anchor Text Term	Linking Root Domains Containing Anchor Text	Links Containing This Anchor Text
miss seo girl	47	6,993
alexandra martin	11	15
miss-seo-girl.com	6	15
http://www.miss-seo-girl.com	5	39
alexandra	4	4

Figure 5-33

Analyse du profil de liens pour un site

	www.miss-seo-girl.com
Page Authority:	48
Page MozRank:	5.49
Page MozTrust:	5.62
Internal Followed Links:	305
External Followed Links:	6,956
Total Internal Links:	305
Total External Links:	7,108
Total Links:	7,413
Followed Linking Root Domains:	57
Total Linking Root Domains:	78
Linking C Blocks:	27

- Le ratio contenus/liens : attention à ne pas avoir une page remplie de liens, le contenu textuel doit prévaloir afin que la page ne soit pas considérée comme du spam. Si toutefois vous possédez de vastes pages remplies de liens, sachez que Google pourra tous les lire normalement. En effet, il fut un temps durant lequel Google ne pouvait lire que cent liens par page, cette ère est désormais révolue…
- Le ratio follow/nofollow : nous avons évoqué ce point lors de l'audit de contenu mais vous pouvez l'étudier ici lors de l'analyse complète du profil de liens pour déterminer la popularité d'un site. Comme mentionné précédemment, il n'existe pas de ratio idéal mais considèrent généralement les spécialistes qu'il faut 25 % à 30 % de backlinks en nofollow et le reste en liens entrants classiques afin d'envoyer un signal positif et de confiance aux moteurs de recherche.

Plusieurs outils gratuits ou payants existent pour étudier le profil de liens d'un site :

- Open Site Explorer : http://www.opensiteexplorer.org ;
- Ahrefs : https://ahrefs.com ;
- Majestic SEO : http://fr.majesticseo.com ;
- Ranks : http://www.ranks.fr ;
- Link Research Tools : http://www.linkresearchtools.com ;
- Explorer de Cognitive SEO : http://explorer.cognitiveseo.com ;
- Link Diagnosis : http://linkdiagnosis.com ;
- Backlink Watch : http://www.backlinkwatch.com ;
- Advanced Link Manager : http://www.advancedlinkmanager.com.

Fluctuation de valeur pour le netlinking

Matt Cutts a évoqué dans une vidéo publiée le 5 mai 2014 sur YouTube (source : http://goo.gl/vsYaJC) la probabilité que les liens entrants et le PageRank perdent encore de l'importance dans les mois et années à venir car ce système datant des origines de Google est trop souvent une cause de spam. D'autres procédés tels que l'AuthorRank ou encore les analyses sémantiques (analyse approfondie des contenus et recherches conversationnelles captées par Google Hummingbird notamment) permettraient de faire la balance avec ce système, sans pour autant supprimer son rôle et son intérêt à 100 %.

Les réseaux sociaux

Les réseaux sociaux sont très importants pour toute stratégie de visibilité sur le Web, au-delà même des aspects de référencement pur. Ils constituent une vraie arme pour favoriser et améliorer la notoriété d'une marque, d'une société ou d'une personne en ligne.

L'usage des médias sociaux prend toute sa place dans une stratégie webmarketing et un plan de communication, la notoriété et la popularité engrangées grâce à une bonne gestion des communautés peut impacter nombre de facteurs (nombre de liens entrants, trafic, crédibilité, taux de clics, taux de rebond…).

En termes de SEO, les moteurs de recherche prennent en compte les signaux sociaux grâce à certaines interconnexions entre membres (AuthorRank de Google et Bing) ou par le biais de liens entrants (PageRank/BrowseRank). Au minimum, il convient de créer des comptes sur les principaux réseaux sociaux tels que Google+, Twitter, Facebook, LinkedIn, Viadeo, YouTube voire sur des seconds réseaux de grande qualité comme Instagram (pour l'échanges de photos et vidéos), FlickR, Foursquare (géolocalisation) ou encore Pinterest…

Le marché des réseaux sociaux est en constante mutation mais certains outils profitent des tendances. Nous sommes actuellement dans une phase de réseaux sociaux proposant des options de confidentialité avancée par exemple, donc il est peut-être intéressant de se pencher sur certains d'entre eux. De manière générale, il faut choisir les plus intéressants pour votre domaine d'activité sachant qu'il existe plus de 600 réseaux sociaux rien qu'en France. Nous avons donc l'embarras du choix et même si tous ne sont pas excellents, beaucoup peuvent permettre d'aider à améliorer notre visibilité, notre notoriété et notre référencement.

Quoi qu'il en soit, retenons que depuis quelques années et pour l'avenir plus ou moins proche, nous devrons être *social friendly*… Il convient donc de s'inscrire et de jouer le jeu des communautés sur les plates-formes qui vous conviennent. Certes, il s'agit d'un travail parfois fastidieux et long, au même titre que celui du référencement, mais les résultats portent généralement leurs fruits après plusieurs mois d'efforts. Même si cela peut paraître difficile et coûteux en temps (et en argent parfois), le retour sur investissement est très souvent au rendez-vous. Il faut juste s'armer de patience et comprendre petit à petit comment mieux maîtriser les réseaux sociaux.

Lors d'un audit de popularité, nous devons nous efforcer de suivre au maximum les tendances qui concernent notre site ou notre nom (en cas de personal branding). Quelques outils permettent d'évaluer globalement la notoriété d'une personne :

- Brandwatch : https://goo.gl/yUJIP9 ;
- CircleCount : http://www.circlecount.com ;
- Klear : http://klear.com ;

Figure 5-34

Suivi de l'e-réputation avec Klear

- Klout : http://klout.com ;

Figure 5-35

Suivi de la notoriété avec l'outil gratuit Klout

- Kred : http://kred.com ;
- How Sociable : http://www.howsociable.com ;
- Reppler : http://www.reppler.com ;
- Social Bakers : http://www.socialbakers.com.

Une autre technique peut s'avérer intéressante pour mesurer la notoriété et l'e-réputation d'un site. Elle consiste à utiliser des outils d'alertes en créant quelques alertes simples sur votre nom, le nom de domaine du site, l'intitulé des produits, la marque ou encore la raison sociale de l'entreprise.

Ainsi, les systèmes d'alertes permettent de suivre en temps réel les mentions qui sont faites de toute votre activité sur la Toile. Il suffira de compiler les données reçues sur plusieurs jours ou plusieurs semaines pour mesurer votre impact ou votre notoriété sur le Web. Il existe une multitude d'outils d'alertes pratiques – gratuits et payants – pour effectuer un suivi efficace, en voici quelques-uns :

- Google Alertes : http://www.google.fr/alerts ;
- Alerti : http://fr.alerti.com ;
- ContentGems : https://contentgems.com ;
- GigaAlert : http://www.gigaalert.com ;
- Diphur : https://diphur.com ;
- FaveBot : http://favebot.com ;
- TalkWalker : http://goo.gl/lvEKU1 ;
- OnWebChange : http://onwebchange.com ;
- HyperAlerts : http://www.hyperalerts.no ;

Figure 5-36

Paramétrage des alertes avec HyperAlerts

- InfoMinder : http://www.infominder.com ;
- Infoxicate : http://infoxicate.me ;
- Mention : https://fr.mention.net ;
- PinAlerts (pour Pinterest) - http://pinalerts.com ;
- Social Mention Alerts : http://socialmention.com/alerts/ ;
- TweetBeep : http://tweetbeep.com ;
- Twilert (pour Twitter) : http://www.twilert.com.

Enfin, le dernier point à vérifier absolument dans le cadre d'un audit SEO est la mise en place de l'AuthorShip pour Bing via Klout (voire pour Google via les microdonnées) afin que l'AuthorRank puisse être pris en compte et que le site profite des actions menées sur les réseaux sociaux. Vous savez tout ce qu'il vous reste à faire désormais alors retenez l'essentiel : be social !

Techniques avancées et outils d'audit

Le suivi de l'indexation et du positionnement ne peuvent pas suffire pour faire du bon travail, il est souvent préférable de disposer d'outils de qualité pour vérifier si nos pages sont bien optimisées ou si elles ont des chances d'obtenir de bons résultats dans les SERP.

Nous allons donc nous pencher sur des outils clés en main ou des codes PHP pour scruter les pages web à la volée et vérifier si nos principales optimisations sont de qualité.

De bons outils sur le marché

Avant d'étudier comment analyser notre contenu par le code ou via un robot personnalisé, nous pouvons citer quelques outils qui permettent d'analyser facilement le contenu des pages web.

Ils présentent souvent l'avantage d'être plus rapides que le code PHP, par exemple, mais ont l'inconvénient d'être souvent payants ou de ne pas faire ressortir toutes les données que nous souhaitons suivre dans les pages web.

Voici quelques exemples d'outils en ligne gratuits et/ou payants :

- Screaming Frog SEO : http://www.screamingfrog.co.uk/seo-spider/ ;
- SEO Chat : http://tools.seochat.com ;
- SEO Grader : http://grader.rezoactif.com ;
- DareBoost : https://www.dareboost.com/fr/ ;
- WooRank : http://www.woorank.com/fr ;
- SiteAnalyzer : http://www.site-analyzer.com/fr ;
- SEOh : http://www.seoh.fr/audit-seo ;
- SEO Mastering : http://www.seomastering.com ;
- InfoWebmaster : http://www.infowebmaster.fr/outils/ ;
- Yakaferci : http://www.yakaferci.com ;
- SEO PowerSuite : http://www.seopowersuite.fr ;
- Advanced Web Ranking : http://www.advancedwebranking.com ;
- SeeUrank : http://www.yooda.com/produits/soft/.

Nous allons présenter quelques outils succinctement pour se faire une idée des possibilités intéressantes en matière d'audit SEO.

Screaming Frog

Screaming Frog (source : http://www.screamingfrog.co.uk/seo-spider/) est un logiciel performant et compatible avec Windows, Linux Ubuntu et Mac OS. Il parcourt rapidement un site complet et affiche nombre d'informations sur les contenus utiles pour le suivi SEO.

Licence Premium pour un accès illimité

Le logiciel est gratuit jusqu'à 500 URL crawlées. Au-delà, il faut souscrire à une licence pour suivre l'ensemble des liens d'un site web.

Une fois le logiciel téléchargé et installé, il est possible de paramétrer en profondeur le système de crawl de l'outil grâce au menu *Configuration*. Par défaut, les options sont plutôt efficaces mais dans certains cas, vous devrez affiner le paramétrage pour obtenir de meilleurs résultats.

Il suffit ensuite de saisir l'URL à analyser dans le champ prévu à cet effet, puis d'attendre le traitement. Le résultat est très intéressant car le logiciel fournit une grande quantité d'informations :

- balise `<title>` et métadonnées pour chaque page ;
- code HTTP (200 si aucune erreur n'est rencontrée) ;
- listes des titres internes (balises `<h1>` à `<h6>`) par page ;
- vérification de la présence de la balise meta `robots` ou `refresh` ;
- nombre de liens internes et externes ;

- nombre de mots par page ;
- poids de la page.

C'est un logiciel très performant et intuitif, nous pouvons donc très rapidement obtenir ce que nous cherchons. L'analyse des données se fait alors très rapidement, il ne nous reste plus qu'à adapter nos contenus ou réparer les erreurs rencontrées pour optimiser le référencement naturel.

Figure 5-37

Suivi des données internes avec Screaming Frog

SEO Chat

SEO Chat (source : http://tools.seochat.com) est un ensemble de services en ligne gratuits qui permettent de contrôler les données internes des pages web, les codes sources mais aussi l'AuthorShip et l'influence sociale.

Tous les outils ne donnent pas des résultats toujours probants mais globalement, il faut bien admettre que SEO Chat apporte des réponses sur de nombreux aspects touchant au référencement naturel.

Figure 5-38

Liste des services SEO de SEO Chat

Par exemple, le service Title Tag and Meta Description Data for Multiple URLs permet de capturer rapidement les titres des pages et les métadonnées de plusieurs URL différentes. Le service Page Image and Link Analysis Tool permet quant à lui de vérifier la présence et le remplissage des attributs `alt` des images mais aussi la présence ou non de liens morts. Il faut donc tester divers outils pour obtenir une masse d'informations intéressantes.

DareBoost

DareBoost est un outil pour réaliser un audit complet d'un site web (source : https://www.dareboost.com/fr/). Il contrôle une centaine de facteurs différents : des critères d'accessibilité, de compatibilité, de qualité, de référencement, de performance et même de sécurité.

L'avantage de l'outil est qu'il détecte rapidement les facteurs bloquants et ceux à optimiser. Des recommandations sont fournies avec des explications claires et des solutions pour chaque problème. C'est un outil vraiment intéressant en somme.

Figure 5-39

Audit de site web avec DareBoost

Yakaferci

Yakaferci (source : http://www.yakaferci.com) est un outil gratuit qui permet d'analyser très vite les contenus des pages web et de voir rapidement si les optimisations que nous avons effectuées sont de bonne facture.

L'outil peut nous faire gagner pas mal de temps dans notre processus d'audit SEO tant il analyse de facteurs. En voici la liste :

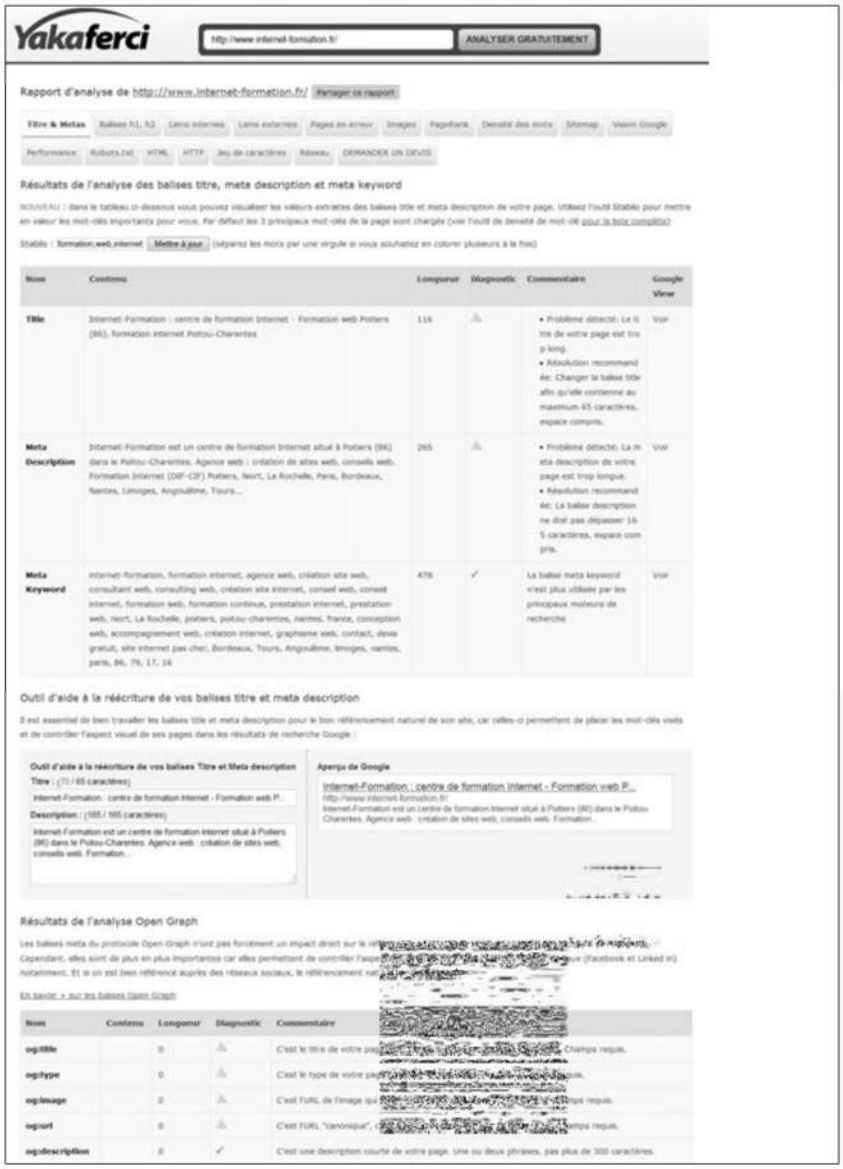
- analyse des balises `<title>` et des métadonnées ;
- analyse des titres internes `<h1>` à `<h6>` ;
- analyse des liens internes et externes, avec PageRank associé ;
- détection des pages d'erreurs ;

- étude des contenus et de la densité des mots-clés ;
- indicateurs de performance ;
- analyse du code source et du réseau ;
- analyse des en-têtes HTTP ;
- vérification des fichiers `sitemap.xml` ;
- vérification du fichier `robots.txt`.

Pour l'utiliser, il suffit d'entrer une adresse web (page par page pour certaines fonctionnalités) et de suivre les indications fournies.

Figure 5-40

Analyse des contenus avec Yakaferci



Les outils pour webmasters

Parmi les outils d'analyse, nous retrouvons les Webmasters Tools fournis par les moteurs de recherche qui peuvent nous être d'une grande utilité pour obtenir des données intéressantes sur les contenus et les recherches des internautes. La Google Search Console permet tout d'abord de vérifier les erreurs d'indexation, et donc de déceler si des pages posent des problèmes ou si des liens morts persistent, par exemple. Pour cela, connectez-vous à votre compte et sélectionnez le menu *Exploration>Erreurs d'exploration*.

Le menu *Apparence dans les résultats de recherche>Données structurées* permet de vérifier la présence des extraits de code enrichis dans les pages du site. Il est également intéressant de contrôler les éventuelles erreurs HTML grâce au menu *Améliorations HTML*.

Figure 5-41

Analyse des rich snippets avec Google

Enfin, le menu *Index Google* permet quant à lui de suivre l'état de l'indexation comme nous l'avons vu précédemment mais aussi de suivre les mots-clés présents dans les textes (avec leur nombre d'occurrences) grâce au sous-menu *Mots-clés de contenu*.

Bing et Yandex proposent aussi des services équivalents dans leur interface pour webmasters. L'outil le plus efficace pour vérifier les titres et les métadonnées est certainement l'analyseur SEO fourni par Bing. Il permet de rapidement mettre en exergue les points à rectifier au sein des pages web. Le validateur de balisage de Bing est également intéressant pour analyser la qualité des extraits de code enrichis. N'hésitez pas à tester les services de Bing qui s'avèrent souvent tout aussi efficaces que les outils proposées dans la Google Search Console.

Suivre les données avec PHP

Tous les outils que nous venons de présenter permettent de récolter rapidement et efficacement de nombreuses informations. Cependant, il est souvent fastidieux de devoir utiliser de multiples services et logiciels pour obtenir certaines données. Qui plus est, il n'est pas toujours aisé de traiter ces informations car elles toutes sont fournies dans le désordre et ne sont pas toujours récupérables dans des bases de données ou des fichiers.

Pour ceux qui préfèrent gérer eux-mêmes leurs informations, il est toujours possible de coder ses propres services afin de parcourir les pages web et recueillir les données qui sont pertinentes pour le suivi SEO. Il ne s'agit que d'exemples de code et tous peuvent être modifiés, réadaptés et même améliorés. L'idée est surtout de présenter diverses solutions pour suivre et crawler nos sites web.

S'il ne fallait en choisir qu'un...

Comme pour une multitude de codes de l'ouvrage, le choix du langage PHP s'est fait par commodité avec le Web. Il est plus fréquent et commun de programmer avec ce langage plutôt qu'en Python, Java, VBScript ou encore C#, bien que tous aient leurs propres avantages et intérêts. Toutefois, des outils similaires peuvent être réadaptés dans ces langages en se basant sur les systèmes présentés par la suite, pour plus de performances dans certains cas...

Surveiller les balises <title> et les métadonnées

Vérifier la présence et la longueur des balises

Nous savons qu'il est important de vérifier l'existence des balises `<title>` dans les pages web voire les métadonnées si vous souhaitez aller plus loin dans l'optimisation. De nombreux outils permettent de vérifier page après page les caractéristiques des pages web mais cela s'avère parfois fastidieux.

Nous allons créer un fichier PHP avec une fonction et un paramétrage par défaut afin de répondre à ces quelques questions pour les sites web statiques.

- Existe-t-il un titre ou non pour la page ? Si oui, quel est-il et quelle est sa longueur ?
- Existe-t-il une description de page ? Si oui, quelle est-elle et quelle est sa longueur ?
- Existe-t-il des métadonnées `keywords` ? Si oui, quels sont-elles et combien en dénombre-t-on ?

La fonction va créer un fichier HTML (au nom de notre choix) pour tenir une sorte de journal des informations avec un code couleur simple : le vert détermine les critères considérés comme optimisés et le rouge va indiquer les points à retravailler.

Par défaut, la fonction vérifie l'existence des balises `<title>` et des métadonnées `description` et `keywords`. Si elles existent, elle les notifie et fournit des indications entre parenthèses :

- nombre de signes du titre sur les 70 caractères visibles sur Google (65 sur Bing) ;
- nombre de signes de la description sur les 160 caractères conseillés (les descriptions n'ont pas de longueur fixes mais les plus larges comptent environ 160 caractères, d'où ce choix) ;
- nombre de mots-clés contenus dans les balises meta `<eywords` (aucune limite imposée mais attention au bourrage de mots-clés).

Nous allons créer un fichier intitulé `titremeta.php` dans lequel nous allons définir et lancer une fonction de crawl des pages web statiques (la méthode ne peut pas fonctionner si les données proviennent d'un traitement PHP via une base de données, par exemple, il faudrait modifier la fonction en conséquence). L'objectif est d'avoir un rendu global et rapide de toutes les balises sans avoir à travailler page par page.

Nous devons tout d'abord placer des paramètres, un peu comme pour le générateur de sitemaps que nous avons étudié auparavant. Suivez les étapes suivantes dans l'ordre pour composer le fichier.

Paramétrage initial

```
// Dossier initial pour lancer la fonction
//('.' par défaut pour la racine, '.NOM-DOSSIER' pour commencer dans un dossier)
$cheminBase = '.';

// URL de base à afficher dans le fichier Sitemap (sans slash à la fin)
$URLSource = 'http://'.$_SERVER['HTTP_HOST'];

// Nom à donner au fichier journal
$fichierSitemap = 'titlemeta.html';

// Liste des extensions à crawler
$extensionsOK = array('php', "asp", "aspx", "py", "xhtml", "phtml", "php3");
$dossiersOK = array();
$fichiersIgnores = array('404.php', '403.php', '500.php', 'footer.php');
// On exclut automatiquement le fichier du script
array_push($fichiersIgnores, basename(__FILE__));
Ouverture du fichier et ajout des bases HTML
// Ouverture du fichier
$crawler = fopen($fichierSitemap,"w");

// On ajoute le doctype et les balises utiles
fputs($crawler, "<!DOCTYPE html>\n");
fputs($crawler, '<meta charset="utf-8"/>'."\n");
fputs($crawler, "<head>\n");
fputs($crawler, "<style type='text/css'>\n");
fputs($crawler, ".green{color:green}\n");
fputs($crawler, ".red{color:red}\n");
fputs($crawler, "</style>\n");
fputs($crawler, "</head>\n");
fputs($crawler, "<body>\n");
```

Ajout de la fonction de traitement

```
function crawlFichier($chemin = '.', $urlBase = '', $extensionsOK = array(),
$fichiersIgnores = array(), $dossiersOK = array()) {
// On ouvre le répertoire
$repertoire = opendir($chemin);

// Formatage du résultat
$result = '';

// On fait une boucle pour lister tous les dossiers et fichiers
while($fichier = readdir($repertoire)) {
    // On récupère l'extension des fichiers listés
    $extensions= strtolower(pathinfo($fichier,PATHINFO_EXTENSION));
```

```
// On exclut les répertoires './' et '../' inutiles
if($fichier != '.' && $fichier != '..' && is_dir($chemin.'/'.$fichier) &&
in_array($fichier,$dossiersOK)) {
    // On encode les fichiers en UTF-8 si ce n'est pas le cas
    if(mb_detect_encoding($fichier) != 'UTF-8') {
    $fichier = utf8_encode($fichier);
    }
    // On lance la fonction récursive jusqu'à la fin du crawl
    CrawlFichier($chemin.'/'.$fichier, $urlBase, $extensionsOK, $fichiersIgnores,
    $dossiersOK);
} elseif(in_array($extensions,$extensionsOK) && !in_array($fichier,$fichiersIgnores)) {
    // Gestion des fichiers
    $ouverture = fopen($fichier,'r');
    $contenu = file_get_contents($fichier);

    // Affichage du nom du fichier ciblé
    $result.= "<b>Fichier : ".$fichier."</b><br/>\n";

    // Extraction du contenu des balises <title>
    preg_match("#<title>(.*)</title>#iU", $contenu, $tab);
    if(!empty($tab[1])) {
        $title = $tab[1];
        $longueurTitle = strlen($title);

        // Formatage de l'affichage
        $result.= "<b>Title : </b>".$title." ";
        if($longueurTitle < 71) {
            $result.= "<span class='green'>(".$longueurTitle." signes sur 70 visibles)
            </span>";
        } else {
            $result.= "<span class='red'>(".$longueurTitle." signes sur 70 visibles)
            </span>";
        }
        $result.="<br/>\n";
    } else {
        $result.= "<span class='red'>Titre manquant !</span><br/>\n";
    }

    // Extraction du contenu des balises meta
    $metas = get_meta_tags($fichier);
    if(isset($metas['description'])) {
        $description = $metas['description'];
        longueurDesc = strlen($description);

        // Formatage de l'affichage
        $result.= "<b>Description : </b>".$description." ";
        if($longueurDesc < 161 && $longueurDesc > 0) {
            $result.= "<span class='green'>(".$longueurDesc." signes sur 160 maximum
            conseillés)</span>";
```

```
            } else {
                if($longueurDesc == 0) {
                    $result.= "<span class='red'>(Champ vide !)</span>";
                } else {
                    $result.= "<span class='red'>(".$longueurDesc." signes sur 160 maximum
                    conseillés)</span>";
                }
            }
        } else {
            $result.= "<span class='red'>Description manquante !</span>";
        }
        $result.= "<br/>\n";

        // Extraction des keywords
        if(isset($metas['keywords'])) {
            $keywords = $metas['keywords'];
            $cleanWords = preg_replace("#(, |,| )#iU", " ", $keywords);
            $tabWords = explode(" ",$keywords);
            $nbWords = count($tabWords, 1);
            $result.= "<b>Keywords : </b>".$keywords." ";
            if($nbWords != 0) {
                $result.= "<span class='green'>(".$nbWords." mots-clés)</span>";
            }
        } else {
            $result.= "<span class='red'>Aucun mot-clé !</span>";
        }
        $result.= "<br/><br/>\n";
        fclose($ouverture); // Fermeture du fichier ouvert
    }
}
global $crawler;
fputs($crawler, $result);
echo $result;
}
```

Lancement de la fonction de crawl

```
// CrawlFichier() avec 5 paramètres utiles :
// 1 -> chemin d'origine
// 2 -> URL de base
// 3 -> tableau des extensions à prendre en compte
// 4 -> tableau des fichiers à ignorer
// 5 -> tableau des dossiers à crawler
CrawlFichier($cheminBase, $URLSource, $extensionsOK, $fichiersIgnores, $dossiersOK);
Fin du code HTML à appliquer
fputs($crawler, "</body>\n");
fputs($crawler, "</html>");
```

Une fois le fichier créé avec tous ces codes, il suffit de le placer à la racine de notre site, que ce soit en ligne ou sur un serveur local (tel que WampServer ou EasyPHP), puis de le lancer. Une fois la tâche réalisée, la fenêtre du navigateur va afficher un aperçu rapide et un fichier de journal va être créé dans le dossier correspondant avec les codes couleurs.

Figure 5-42

Fichier journal d'analyse des titres et métadonnées

Un code pour les sites statiques

Le programme ne fonctionne que sur des balises entrées statiquement dans les fichiers. Si nous voulons un système équivalent pour récupérer des données dynamiques, il faudra modifier la fonction de crawl soit en accédant à une base de données, soit en améliorant le système de lecture des fichiers.

Ainsi, nous pouvons en un seul coup d'œil vérifier l'existence ou non des balises ainsi que leur longueur et capacité d'optimisation. Toutefois, la fonction ne comptabilise pas le nombre d'occurrences des mots, c'est pourquoi nous allons créer trois autres fonctions associées.

Comptabiliser le nombre d'occurrences des mots-clés

Il peut être intéressant de savoir quels mots-clés sont les plus répétés au sein des balises `<title>` et dans les métadonnées afin d'avoir une perception rapide du travail d'optimisation déjà effectué ou à réaliser. Nous allons créer un fichier que nous pouvons appeler `titlemetacount.php` dans lequel seront insérées trois fonctions PHP utilisées en cascade :

- une fonction de découpage des chaînes de caractères, c'est-à-dire pour distinguer les mots-clés du titre et des métadonnées ;
- une fonction d'affichage des résultats sous forme de tableau (qui peut être totalement modifiée selon vos envies) ;
- une fonction de traitement des données qui utilisent les deux codes précédents. C'est cette fonction que nous utiliserons en appel pour faire fonctionner le système.

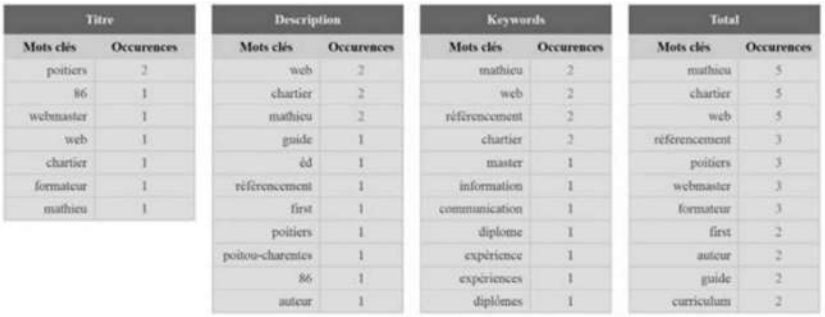
La fonction nous permet de faire ressortir quatre tableaux, bien que nous pourrions l'agrémenter pour aller bien plus loin et analyser la totalité des contenus des pages si nous le voulions. Nous obtenons :

- un tableau pour les mots-clés de la balise `<title>` ;
- un tableau pour les termes de la description ;

- un tableau pour les termes placés dans la balise meta `keywords` ;
- un tableau combiné qui résume le nombre total d'occurrences dans ces trois balises HTML.

Figure 5-43

Aperçu des tableaux récapitulatifs du nombre d'occurrences des mots-clés



Titre	
Mots clés	Occurences
poitiers	2
86	1
webmaster	1
web	1
chartier	1
formateur	1
mathieu	1

Description	
Mots clés	Occurences
web	2
chartier	2
mathieu	2
guide	1
éd	1
référencement	1
first	1
poitiers	1
poitou-charentes	1
86	1
auteur	1

Keywords	
Mots clés	Occurences
mathieu	2
web	2
référencement	2
chartier	2
master	1
information	1
communication	1
diplome	1
expérience	1
expériences	1
diplômes	1

Total	
Mots clés	Occurences
mathieu	5
chartier	5
web	5
référencement	3
poitiers	3
webmaster	3
formateur	3
first	2
auteur	2
guide	2
curriculum	2

Comme pour le système précédent, nous allons procéder par étape en copiant successivement les fonctions suivantes dans le fichier final pour obtenir un code fonctionnel.

Fonction de découpage des mots-clés

```
function cutStr($chaine = '', $stopwords = array(), $ordre = array("VALUE", "DESC"),
$codage = 'UTF-8') {
// Liste de caractères à tolérer
$chars = "ÀÁÂÃÄÅàáâãäåÒÓÔÕÖØòóôõöøÈÉÊËèéêë";
$chars.= "ÇçÌÍÎÏìíîïÙÚÛÜùúûüÿÑñ0123456789";

// Découpage des mots
$chaine = mb_strtolower($chaine, $codage);
$tabClean = str_word_count($chaine, 1, $chars);

// Nettoyage des caractères inutiles
foreach($tabClean as $key => $value) {
    // Liste de caractères spéciaux
    $deleteValue = array("-","(",")","[","]","{","}","_");
    // Exclusion des stopwords éventuels
    if(in_array($value, $stopwords)) {
        $value = '';
    }
    // Exclusion des caractères spéciaux
    if(in_array($value, $deleteValue)) {
        $value = '';
    }
    // Suppression des clés gênantes
    if(empty($value)) {
        unset($tabClean[$key]);
    }
}
```

```
// On compte le nombre d'occurrences
$nbValues = array_count_values($tabClean);

// Ordre d'affichage des données
if($ordre[0] == "VALUE" || $ordre[0] == "value") {
    if($ordre[1] == "ASC") {
        asort($nbValues);
    } else {
        arsort($nbValues);
    }
}
if($ordre[0] == "KEY" || $ordre[0] == "key") {
    if($ordre[1] == "ASC") {
        ksort($nbValues);
    } else {
        krsort($nbValues);
    }
}
    return $nbValues;
}
```

La fonction `cutStr()` peut prendre quatre paramètres utiles :

- le texte à découper (obligatoire) ;
- un tableau contenant des *stop words* à exclure, c'est-à-dire tous les caractères ou mots courts inutiles que nous ne voulons pas compter (les articles, les conjonctions de coordination…) ;
- un tableau à deux paramètres pour ordonnancer les résultats dans les tableaux avec `array(parametre1, parametre2)` :
  - le premier paramètre est `"VALUE"` (classer par occurrence) ou `"KEY"` (trier par mot) ;
  - le second paramètre est `"ASC"` (tri ascendant) ou `"DESC"` (tri descendant) ;
- un encodage particulier des caractères si nécessaire (UTF-8 par défaut) pour éviter des problèmes avec les accents mal encodés et donc les mots mal découpés.

Les paramètres seront à régler dans la fonction générale du système pour que tout corresponde à vos attentes.

Fonction d'affichage des tableaux

```
function displayTable($tab, $titre = '') {
$result = "<table style='background:#ccc; width:23%; margin-right:2%; float:left;'>\n";
if(!empty($titre)) {
$result.= "<caption style='color:#eee; background:#666;
padding:.5em;'><b>".$titre."</b></caption>\n";
}
$result.= "<tr style='color:#000; background:#ccc'>\n";
$result.= "<th style='padding:.2em .5em;'>Mots clés</th>\n";
```

```
$result.= "<th style='padding:.2em .5em;'>Occurences</th>\n";
$result.= "</tr>\n";
foreach($tab as $key => $value) {
    $result.= "<tr style='color:#444; background:#ddd;'>\n";
    $result.= "<td align='right' style='padding:.2em .5em;'>".$key."</td>\n";
    if($value > 1) {
        $result.= "<td align='center' style='padding:.2em .5em; color:green;'>".$value."
        </td>\n";
    } else {
        $result.= "<td align='center' style='padding:.2em .5em;'>".$value."</td>\n";
    }
    $result.= "</tr>\n";
}
$result.= "</table>\n";
return $result;
}
```

Cette fonction peut totalement être personnalisée selon l'affichage que vous désirez. Ici, elle génère plusieurs tableaux en fonction des données présentées afin de voir rapidement le nombre d'occurrences par type de balise. Elle prend deux paramètres :

- un tableau PHP de mots ou d'expressions (dès que la découpe est effectuée dans notre système) ;
- un titre pour le tableau correspondant.

Fonction générale du système

```
function TitleMetaCount($page = '', $stopwords = array()) {
// Ouverture du fichier en lecture seule
$ouverture = fopen($page,'r');
$contenu = file_get_contents($page);
fclose($ouverture);

// Formatage du résultat
$result = '';

// Extraction du contenu des balises <title> et des métadonnées
$strTotal = ''; // Chaîne complète

// Titre
preg_match("#<title>(.*)</title>#iU", $contenu, $tab);
if(!empty($tab[1])) {
    $title = $tab[1];
    $cut = cutStr($title, $stopwords);
    $result .= displayTable($cut, 'Titre');
    // Ajout du titre à la chaîne complète
    $strTotal .= $title;
}
// Métadonnées
$metas = get_meta_tags($page);
```

```
if(isset($metas['description'])) {
    $description = $metas['description'];
    $cut = cutStr($description, $stopwords);
    $result .= displayTable($cut, 'Description');
    // Ajout de la description à la chaîne complète
    $strTotal .= " ".$description;
}
if(isset($metas['keywords'])) {
    $keywords = $metas['keywords'];
    $cut = cutStr($keywords, $stopwords);
    $result .= displayTable($cut, 'Keywords');
    // Ajout des mots-clés à la chaîne complète
    $strTotal .= " ".$keywords;
}

// Tableau total
$cutTotal = cutStr($strTotal, $stopwords);
$result .= displayTable($cutTotal, 'Total');
$result .= "<p style='clear:both'></p><br/>\n";

return $result;
// exit(); // optionnel
}
```

Il s'agit de la fonction principale que nous lancerons pour activer le système de comptage du nombre d'occurrences. Cette dernière peut être personnalisée si besoin sur plusieurs aspects :

- paramétrage des fonctions `cutStr()` et `displayTable()` selon nos envies ;
- ajout ou non de fonction de comptage supplémentaire (par exemple, nous pourrions ajouter des codes pour compter le nombre d'occurrences dans les balises `<h1>` à `<h6>`…) ;
- personnalisation de l'affichage du résultat ;
- la fonction `TitleMetaCount()` peut prendre deux paramètres :
  - la page à analyser (obligatoire) ;
  - un tableau des stop words qui sera renvoyé automatiquement vers la fonction `cutStr()`.

Une fois le fichier final créé, il suffit de l'inclure et de lancer la fonction en haut des pages que nous souhaitons analyser avec le code suivant, par exemple :

Lancement et usage du système de comptabilisation

```
<?php
// Inclusion de la fonction
include_once('titlemetacount.php');
// Lancement de la fonction
$stopwords = array('le', 'la', 'les', 'un', 'une', 'des', 'de', 'du', 'mais',
'ou', 'et', 'donc', 'or', 'ni', 'car', 'se', 'en');
echo TitleMetaCount(basename(__FILE__), $stopwords);
?>
```

Coupler analyse et comptage des mots-clés par le code

Le défaut du système précédent est que le calcul du nombre d'occurrences se fait fichier par fichier. Nous allons donc le coupler à la première fonction (`crawlFichier()`) pour appliquer le comptage dynamiquement en fonction du crawl des fichiers. En réalité, c'est extrêmement simple à réaliser puisqu'il suffit d'ajouter seulement quelques lignes de code dans la fonction `crawlFichier()` pour rendre le système fonctionnel. Tout d'abord, nous devons inclure les fonctions de comptage dans le fichier de crawl (que nous avions appelé `titremeta.php`) avec la ligne suivante placée en haut du fichier :

```
include_once('titlemetacount.php');
```

Ensuite, nous devons aller vers la fin de la fonction `crawlFichier()`, avant la commande `fclose($ouverture);`, et ajouter ceci :

```
$stopwords = array('le', 'la', 'les', 'un', 'une', 'des', 'de', 'du', 'mais',
'ou', 'et', 'donc', 'or', 'ni', 'car', 'se', 'en');
$result.= TitleMetaCount($fichier, $stopwords);
```

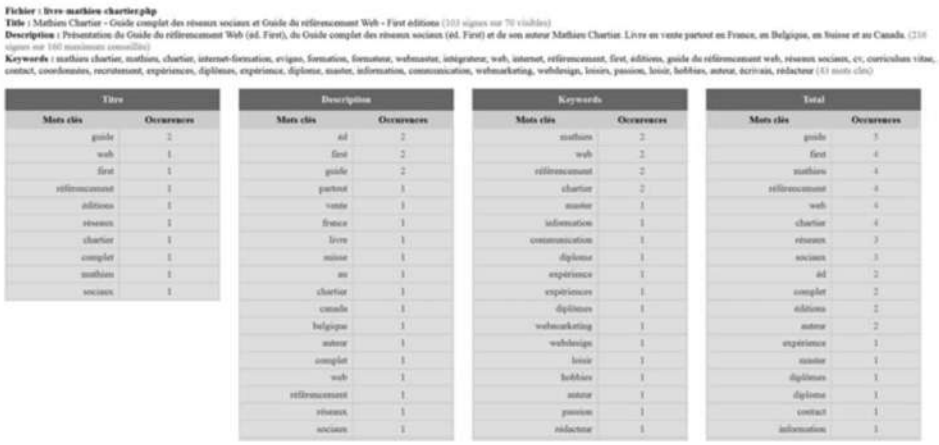
Optimiser les performances du programme

Idéalement, il faut placer le tableau des stop words avant la boucle `WHILE` pour éviter des redondances inutiles, donc n'hésitez pas à le faire.

Une fois ces modifications effectuées, il suffit de lancer le fichier `titlemeta.php` via la barre d'adresse du navigateur pour lancer le programme complet et obtenir une analyse approfondie des mots-clés sur les balises de la section `<head>`.

Figure 5-44

Analyse complète des mots-clés pour chaque page



Possibilité d'extension du système

Le système pourrait être amélioré pour compter également des expressions clés, mais la fonction de découpage `cutStr()` n'est pas prévue pour cela ici. Il faudrait donc l'améliorer pour obtenir un rendu encore plus puissant et créer un système équivalent à ce que nous pouvons trouver sur des outils tels que celui proposé sur le site alyze.info.

Développer son propre robot en PHP

Intérêt d'un robot personnalisé

Depuis le début de cet ouvrage, une multitude de fonctions a été présentée pour contrôler, vérifier ou générer des données à partir d'une adresse web simple ou d'un site complet. Toutefois, nous n'avons jamais eu affaire à un robot d'indexation dans les règles de l'art, le problème étant lié en général à la quantité de ressources utiles pour faire fonctionner ce type de système.

En effet, les méthodes que nous utilisons depuis le début de notre propos sont plutôt rapides et permettent d'obtenir des résultats convaincants pour la plupart des usages. Néanmoins, elles présentent toutes le même inconvénient : il est impossible de crawler des sites externes aisément et donc de procéder à une extraction complète de données comme le ferait un moteur de recherche.

L'avantage des robots d'indexation est qu'ils récupèrent à la volée les liens présents dans les pages web, puis crawlent incessamment en passant de pages en pages tout en listant les liens présents et en récupérant des informations intéressantes. Le principal atout des robots est de pouvoir se connecter à n'importe quel site et de récupérer n'importe quel type d'information pertinente à nos yeux, c'est donc le meilleur moyen pour effectuer un suivi complet.

Méthode de développement

Il n'existe pas de meilleures méthodes pour créer des robots sur la Toile. Certains développeurs optent pour des systèmes programmés en Java, en C# ou encore en Python quand d'autres préfèrent le classique PHP. Quel que soit le langage usité, sachez qu'il est surtout question de ressources de serveur ici ou encore de mémoire RAM tant les crawlers imposent de longs traitements, il convient donc de ne pas tester ce type d'outils directement sur des sites d'envergure (avant d'être sûr du bon fonctionnement) si vous souhaitez développer votre propre système.

Avec ou sans programmation orientée objet

Les codes que nous allons présenter sont programmés en PHP structurel avec cURL mais auraient fort logiquement dû être adaptés en PHP orienté objet pour plus de clarté. Cela ne change ni la qualité du robot ni même le fonctionnement de l'outil, mais l'objet pourrait sembler plus logique pour ce type de programme.

Nous allons créer un spider « simple » dans son principe. Son rôle va être de récupérer le maximum de liens internes d'un site web en partant d'un nom de domaine (et éventuellement d'une page de départ) en crawlant récursivement jusqu'à obtenir satisfaction.

Prenons un exemple : une page contient cinq liens, le robot va faire un premier crawl pour récupérer ses cinq URL. Ensuite, si nous avons autorisé le script à faire plusieurs tours de lecture, le robot va extraire tous les liens issus des cinq pages précédemment trouvées, et ainsi de suite jusqu'à ce qu'il ait fait le tour des liens…

Pour des raisons évidentes de gestion des ressources, le script est muni de plusieurs « stops », c'est-à-dire de codes pour freiner ou arrêter le script en cas de plantage ou de dépassement du temps de traitement autorisé par le serveur (sécurité). En voici quelques-uns :

- la fonction `set_time_limit()` gère le temps d'exécution maximal du script. Par défaut, la valeur est faible et le robot ne peut pas crawler suffisamment, il faut donc mettre une valeur importante (`0` signifie qu'il n'existe aucune limite) ;
- la variable `$level` permet d'indiquer le nombre de niveau de récursivité à appliquer. Plus le chiffre est élevé, plus le robot va lire les liens dans les profondeurs du site. La valeur `0` ne fait qu'un « tour » de crawl, à savoir l'extraction des liens de la page donnée. La valeur `1` va extraire les liens des pages issues du premier crawl, etc. Il est déconseillé de mettre une valeur élevée si vous crawlez un site profond (il est préférable d'effectuer des tests à bas niveau dans un premier temps) ;
- un tableau appelé `$liensVisites[]` contient la liste des pages déjà crawlées, ce qui évite au robot de relire sans cesse des pages déjà parcourues. Ce système est l'un des meilleurs moyens pour réduire le temps de traitement et économiser des ressources serveur ;
- plusieurs variables viennent bloquer les pages qui pourraient causer des troubles dans le script. Il arrive en effet que des pages web génèrent des erreurs imprévues. Par exemple, si vous avez pris l'habitude d'utiliser des systèmes de CSS dynamiques avec Less CSS, SASS ou même via PHP, vous avez sans doute appelé les feuilles de styles `style.php`. Comme le script lit les pages PHP (ainsi que d'autres extensions de pages web classiques comme ASP, HTM, HTML ou encore PY), le script va crawler la page `style.php` qui pourrait générer une erreur. Bien sûr, ce problème ne doit pas avoir lieu normalement, mais il vaut mieux prévoir une surcouche sécuritaire et c'est pourquoi des stops ont été prévus et un tableau de pages à exclure a été ajouté…

Le script lit toutes les principales extensions des langages courants mais il prévoit aussi les cas de réécritures d'URL avec des adresses sans extension finale. Cela évite les mauvaises surprises si vous voulez extraire les liens internes d'un site aux adresses réécrites.

Pour des raisons techniques, le script a été découpé en plusieurs fonctions distinctes qui s'entremêlent pour créer le robot final :

- la fonction `extractionLiens()` permet d'extraire les liens internes d'une page unique ;
- la fonction `extractionRecursive()` sert à lancer l'extraction des liens de manière répétée sur une liste d'URL donnée ;
- la fonction `crawlRecursif()` permet d'effectuer des tours de crawl pour aller de plus en plus en profondeur dans les méandres des sites web. C'est ici qu'intervient la variable `$level` pour fixer les niveaux de profondeurs des liens à explorer et `$max` pour fixer le nombre maximal de pages à crawler selon le niveau de profondeur déterminé par `$level`.

Le robot rend un tableau de liens (sans doublon) à la fin du traitement. Il suffit de stocker chacune de ces adresses dans un fichier externe ou dans une base de données pour les enregistrer. Contrairement à certains robots qui multiplient les traitements, ce dernier n'a pour but que de récupérer les liens internes. L'objectif est justement de limiter au maximum les surcharges inutiles dans un premier temps.

Les caprices de Firefox

Il arrive que pour des crawls importants, Internet Explorer, Safari ou Opera soient beaucoup moins capricieux que Firefox…

Si nous souhaitons effectuer d'autres traitements, récupérer les liens externes ou encore extraire des données pour notre suivi, il suffit de le faire ultérieurement à partir de la liste de liens que nous avons récupérée lors du crawl…

Figure 5-45

Crawl sur deux niveaux de profondeur dans un site

Niveau 1 de crawl (indice $level = 0;)

Nombre de liens internes : 18

```
Array
(
    [0] => http://www.internet-formation.fr/
    [1] => http://www.internet-formation.fr/agence-internet-formation.php
    [2] => http://www.internet-formation.fr/conseil-web-coaching.php
    [3] => http://www.internet-formation.fr/creation-site-web.php
    [4] => http://www.internet-formation.fr/cursus-metiers-web.php
    [5] => http://www.internet-formation.fr/devis.php
    [6] => http://www.internet-formation.fr/formation-html-5-css-3.php
    [7] => http://www.internet-formation.fr/formation-referenceur-web-emarketing.php
    [8] => http://www.internet-formation.fr/formation-web-2.php
    [9] => http://www.internet-formation.fr/formation-web-internet.php
    [10] => http://www.internet-formation.fr/formation-web-mobile.php
    [11] => http://www.internet-formation.fr/formation-webmaster.php
    [12] => http://www.internet-formation.fr/internet-formation-contact.php
    [13] => http://www.internet-formation.fr/internet-formation-credits.php
    [14] => http://www.internet-formation.fr/internet-formation-faq.php
    [15] => http://www.internet-formation.fr/plan-du-site.php
    [16] => http://www.internet-formation.fr/prestation-service-web.php
    [17] => http://www.internet-formation.fr/references-internet-formation.php
)
```

Niveau 2 de crawl (indice $level = 1;)

Nombre de liens internes : 78

```
Array
(
    [0] => http://www.internet-formation.fr/
    [1] => http://www.internet-formation.fr/accompagnement-communication-web.php
    [2] => http://www.internet-formation.fr/accompagnement-ecriture-web.php
    [3] => http://www.internet-formation.fr/accompagnement-ergonomie-web.php
    [4] => http://www.internet-formation.fr/accompagnement-site-web2.php
    [5] => http://www.internet-formation.fr/accompagnement-webdesigner.php
    [6] => http://www.internet-formation.fr/agence-internet-formation.php
    [7] => http://www.internet-formation.fr/conseil-web-coaching.php
    [8] => http://www.internet-formation.fr/creation-blog-professionnel.php
    [9] => http://www.internet-formation.fr/creation-site-web-cms.php
    [10] => http://www.internet-formation.fr/creation-site-web-ecommerce.php
    [11] => http://www.internet-formation.fr/creation-site-web-sur-mesure.php
    [12] => http://www.internet-formation.fr/creation-site-web-vitrine.php
    [13] => http://www.internet-formation.fr/creation-site-web.php
    [14] => http://www.internet-formation.fr/cursus-metiers-web.php
    [15] => http://www.internet-formation.fr/cursus-webdesigner-cif.php
    [16] => http://www.internet-formation.fr/cursus-webmaster-cif.php
    [17] => http://www.internet-formation.fr/devis.php
    [18] => http://www.internet-formation.fr/formation-cms-outils-web.php
    [19] => http://www.internet-formation.fr/formation-dreamweaver.php
    [20] => http://www.internet-formation.fr/formation-ergonomie-web-conception.php
    [21] => http://www.internet-formation.fr/formation-gestion-projet-web.php
    [22] => http://www.internet-formation.fr/formation-html-5-css-3.php
    [23] => http://www.internet-formation.fr/formation-html-css-integration.php
    [24] => http://www.internet-formation.fr/formation-integrateur-web.php
    [25] => http://www.internet-formation.fr/formation-newsletter-emailing.php
    [26] => http://www.internet-formation.fr/formation-photoshop-semiotique.php
    [27] => http://www.internet-formation.fr/formation-php-mysql-developpement.php
    [28] => http://www.internet-formation.fr/formation-redacteur-web-journaliste.php
    [29] => http://www.internet-formation.fr/formation-referenceur-web-emarketing.php
    [30] => http://www.internet-formation.fr/formation-web-2.php
    [31] => http://www.internet-formation.fr/formation-web-internet.php
    [32] => http://www.internet-formation.fr/formation-web-mobile.php
    [33] => http://www.internet-formation.fr/formation-webmaster.php
```

Améliorer le robot PHP pour plus de performances

Ce code est loin d'être parfait dans tous les cas de figure, il pourrait encore être optimisé davantage ou subir une refonte pour lui donner plus de possibilités. Libre à vous de vous imprégner du code et de mettre la main à la pâte pour extraire les liens externes ou ajouter dynamiquement les liens dans une base de données...

Code PHP du spider interne

Maintenant que vous connaissez le principe et le système général de développement du robot, voici les codes commentés de chaque fonction. Il suffit d'insérer ces codes les uns à la suite des autres dans un fichier et d'ajouter en dernier lieu les paramètres d'utilisation du robot.

Fonction d'extraction des liens

```
// Extraction des liens d'une page donnée
function extractionLiens($domaine = '', $page = '', $pageExclues = array(),
$queryString = false) {
if(!empty($domaine)) {
    // Ajout du protocole s'il est manquant
    if(!preg_match("#^https?://#iU", $domaine)) {
        $domaine = "http://".$domaine;
    }

    // Découpage des noms de domaines inexacts
    $regex = "#^(https?://.*)/([a-zA-Z0-9_./-]+)$#iU";
    if(preg_match($regex, $domaine, $parties)) {
        $domaine = $parties[1];
        $page = $parties[2];
    }

    // Suppression des domaines en trop
    // if(stripos($page, $domaine) != false) {
    if(preg_match("#".$domaine."#iU", $page)) {
        $page = str_ireplace($domaine, '', $page);
    }

    // Ajout du slash final s'il est manquant
    if(substr($domaine, -1, 1) != "/") {
        $url = $domaine."/".$page;
    } else {
        $url = $domaine.$page;
        $domaine = substr($domaine,0,-1);
    }

    // Récupération des données avec cURL
    $curl = curl_init($url);
    $userAgent = $_SERVER["HTTP_USER_AGENT"];
    curl_setopt($curl, CURLOPT_USERAGENT, $userAgent);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
```

```
curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, 0);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($curl, CURLOPT_FAILONERROR, 1);
$contenu = curl_exec($curl); // Contenu complet
$codeHTTP = curl_getinfo($curl, CURLINFO_HTTP_CODE); // Code HTTP
curl_close($curl);

// Extraction des liens de la page
// Variante avec http://simplehtmldom.sourceforge.net/
$regex = "#href=[<\"']("".$domaine."/|/)?([a-zA-Z0-9_./-]+([\?\#\&][^\"'>]+)?)[\"'>]#iU";
preg_match_all($regex, $contenu, $liens);

// Suppression des doublons dans la liste d'URL
$liens = array_unique($liens[2]);

// Suppression des liens qui ne sont pas des pages web
if(!empty($liens)) {
    foreach($liens as $lien) {
        // Suppression du slash de début de chaîne (gênant)
        if(substr($lien, 0, 1) == "/") {
            $lien = substr($lien, 1);
        }

        // Suppression des liens relatifs avec "../"
        // if(stripos("../", $domaine) != false) {
        if(preg_match("#../#", $lien)) {
            $lien = str_ireplace("../", "", $lien);
        }

        // Liste des extensions autorisées
        // Ne pas oublier "" et "/" (en cas de réécriture sans extension)
        $extensionsOK = array("/", "", "php", "htm", "html", "xhtml", "phtml", "dhtml",
        "asp", "aspx", "php3", "py", "jsp", "shtml", "rhtml");

        // Récupération de l'extension des fichiers (si elle existe)
        $extension = pathinfo($lien, PATHINFO_EXTENSION);

        // Vérification de la présence de queryString
        if($queryString == true) {
            if(preg_match("#[\?\#\&][^=]+=.*#iU", $extension)) {
                $qsOK = true;
            } else {
                $qsOK = false;
            }
        } else {
            $qsOK = false;
        }
```

```
            // Conservation des extensions valides uniquement !
            if($qsOK == true || in_array($extension, $extensionsOK)) {
                $urlValide = true;
            } else {
                $urlValide = false;
            }

            // Enregistrement des liens à crawler (si autorisé !)
            if($urlValide == true && $codeHTTP == 200 && !in_array($lien, $pageExclues)) {
                // Ajout du nom de domaine avant les liens
                $listeLiens[] = $lien;
            } else {
                $listeLiens[] = false;
            }
        }
    } else {
        $listeLiens[] = '';
    }

    // Suppression des entrées vides
    foreach($listeLiens as $lien) {
        if(!empty($lien)) {
            $liste[] = $lien;
        }
    }

    // Récupération des liens (tableau)
    if(!empty($liste)) {
        // Tableau des liens acquis
        return $liste;
    } else {
        // Retourne un tableau vide en cas de problème
        return array();
    }
}
}
```

Ensuite, collez la fonction d'extraction récursive. Elle retourne un tableau à deux indices. L'indice `[0]` correspond aux nouveaux liens récupérés dans la phase de crawl et l'indice `[1]` liste les pages déjà visitées et crawlées.

Fonction d'extraction récursive

```
// Fonction utilisée avec récursivité pour récupérer un maximum de liens
function extractionRecursive($tabLiens, $domaine, $liensVisites = array(),
$pageExclues = array(), $queryString = false, $max = 0) {
// Détermine si les liens sont valides ou non
$OK = false;
```

```
// Définition du maximum de pages à crawler
if($max == 0 || $max == '') {
    $max = count($tabLiens);
}

// Boucle pour extraire les liens de la nouvelle liste d'URL
$nb = 0;
foreach($tabLiens as $lien) {
    // Si le lien n'a pas encore été crawlé
    if(!in_array($lien, $liensVisites) && $lien != '' && $nb < $max) {
        // Récupération du contenu et des liens de chaque page
        $nouveauTabLiens[] = extractionLiens($domaine, $lien, $pageExclues, $queryString);

        // Ajout de l'URL de la page déjà visitée...
        if($lien != "/") {
            $liensVisites[] = $lien;
        }

        // Le lien était valide, on peut continuer...
        $OK = true;
    }
    $nb++;
}

// S'il s'agit d'un lien valide et d'une liste non vide
if($OK == true && !empty($nouveauTabLiens)) {
    // Rassemble les sous-tableaux et efface les doublons
    foreach($nouveauTabLiens as $liens) {
        foreach($liens as $lien) {
            $tab[] = $lien;
        }
    }
    $tab = array_unique($tab);

    // Tous les liens dédoublonnés (tableau)
    $resultat[0] = array_unique($tab);

    // Ajoute les liens déjà visités
    $resultat[1] = $liensVisites;

    return $resultat;
} else {
    // Tous les liens dédoublonnés (tableau)
    $resultat[0] = $tabLiens;

    // Ajoute les liens déjà visités
    $resultat[1] = $liensVisites;

    return $resultat;
}
}
```

Ajoutez ensuite la fonction de crawl récursif qui constitue le cœur technique du robot. Elle retourne la liste finale des liens obtenus lors de la recherche interne au sein d'un tableau PHP.

Fonction de crawl récursif

```
// Fonction du crawl récursif par niveau de profondeur
function crawlRecursif($domaine, $level = 1, $max = 0, $listeLiens = array(),
$liensVisites = array(), $queryString = false, $pageExclues = array()) {
// Création de variables de tableau
// Nécessaire pour la récursivité !
$nouveauTabLiens[0] = $listeLiens;
$nouveauTabLiens[1] = $liensVisites;

// Boucle avec récursivité (selon un niveau de profondeur)
// Nombre de niveaux fixé par $level
$nb = 0;
while($nb < 2) {
    if(!empty($nouveauTabLiens[0])) {
        $nouveauTabLiens = extractionRecursive($nouveauTabLiens[0], $domaine,
        $nouveauTabLiens[1], $pageExclues, $queryString, $max);
    }
    $nb++;
}

if(in_array("/", $nouveauTabLiens[0])) {
    $nouveauTabLiens[0] = array();
}

// Fusion des tableaux de liens (extraits + visités)
$resultat = array_merge($nouveauTabLiens[0], $nouveauTabLiens[1]);

// Dédoublonnage de la liste de liens finale
$resultat = array_unique($resultat);

// Ajout du protocole s'il est manquant
if(!preg_match("#^https?://#iU", $domaine)) {
    $domaine = "http://".$domaine;
}

// Suppression du slash de début de chaîne (gênant)
if(substr($domaine, 0, 1) == "/") {
    $domaine = substr($domaine, 1);
}

// Découpage des noms de domaines inexacts
$regex = "#^(https?://.*)/([a-zA-Z0-9_./-]+)$#iU";
if(preg_match($regex, $domaine, $parties)) {
    $domaine = $parties[1];
}
```

```
// Reforme les liens absolus et ajoute le domaine à la liste
if(substr($domaine, -1, 1) != "/") {
    $domaine = $domaine."/";
}

foreach($resultat as $lien) {
    $tabFinal[] = $domaine.$lien;
}
$tabFinal[] = $domaine;

// Classe le tableau par valeurs (alphabétiques)
sort($tabFinal);

return $tabFinal;
}
```

Si vous ne voulez pas perdre tous les liens obtenus et pour éviter de relancer incessamment le crawl, il est préférable d'enregistrer les données. Vous pouvez le faire au sein d'une base de données ou dans des fichiers. Voici une fonction pour l'exemple qui permet d'enregistrer la liste des liens dans des fichiers CSV.

```
// Fonction d'enregistrement des liens dans un fichier CSV
function enregistrementFichier($listeURL = array(), $domaine = '', $repertoireLogs = '') {
// Création du journal si inexistant
if(!empty($repertoireLogs) && !is_dir($repertoireLogs)) {
    mkdir($repertoireLogs, 0705);
}

// Suppression du protocole du domaine (pour le titre du fichier)
if(!empty($domaine)) {
    $regex = "#(https?://)#iU";
    $domaine = preg_replace($regex, '-', $domaine);
    $domaine = str_replace("/", "-", $domaine);
}

// Création et remplissage du fichier CSV
$nomFichier = $repertoireLogs.'liste-url'.$domaine.'.csv';
$fichier = fopen($nomFichier, 'w+');

foreach($listeURL as $lien) {
    fputcsv($fichier, array($lien), ";");
}

// Fermeture du fichier
fclose($fichier);
}
```

Programme modulaire pour enregistrer les données

Cette fonction peut être remplacée par un équivalent afin de recueillir les liens dans une base de données pour les traiter plus facilement par la suite.

Enfin, n'oubliez pas de terminer le fichier du robot par les paramétrages utiles afin de lancer le crawl. Certains paramètres sont optionnels. Par exemple, vous pouvez lancer le crawl à partir d'un nom de domaine ou d'une page, mais dans ce second cas, il faut distinguer la variable `$domaine` faite pour le nom de domaine et la page de départ placée dans la variable `$page`.

```
// Temps maximal d'exécution du script
set_time_limit(0);

// Nom de domaine (et page de départ si besoin) à crawler
$domaine = "www.domaine.ext";
$pageDepart = '/';

// Crawl des URL avec queryString ? (true/false)
// Beaucoup de ressources si "true"
$queryString = false;

// Nombre de niveaux de profondeur à crawler (0 à xxx)
// Plus le nombre est élevé, plus il faut de ressources
$level = 1;

// Nombre maximal d'URL à crawler (par niveau de profondeur)
// 0 ou vide pour tout crawler
$max = 0;

// Liste de pages à exclure (si nécessaire)
$pageExclues = array('style.php'); // Exemple

// Obligatoire : première liste de liens à extraire
// Ou $premiereListe = extractionLiens($domaine, $pageDepart, $pageExclues);
$premiereListe = array($pageDepart);
$pagesVisitees = array();

// Récupération récursive de la liste de liens dans une variable
$liste = crawlRecursif($domaine, $level, $max, $premiereListe, $pagesVisitees,
$queryString, $pageExclues);

// Enregistrement de la liste d'URL dans un fichier CSV (optionnel)
enregistrementFichier($liste, $domaine);
```

Une fois tout ce code placé dans un fichier, il suffit de le lancer à partir de votre serveur ou d'un serveur local en ajoutant l'URL de votre choix ainsi que le niveau de profondeur à crawler. Cela peut prendre plusieurs minutes en fonction du niveau choisi.

Dès le traitement terminé, un fichier CSV (par site crawlé) liste les liens récupérés. Nous allons maintenant voir comment utiliser ce robot pour notre suivi SEO.

Récupérer des contenus de pages web

Pour tous les traitements qui vont suivre, nous allons devoir utiliser une fonction de récupération des contenus et autres données pour chaque page crawlée et enregistrée (que ce soit dans un fichier ou une base de données).

Il vous suffit d'appliquer la méthode de votre choix pour récupérer la liste des liens, puis de créer une fonction celle présentée ci-après pour récupérer les contenus des pages. Dans notre cas, elle s'appuie une nouvelle fois sur l'extension cURL pour PHP qui s'avère pratique dans bien des situations.

Grâce à la fonction `curl_getinfo()` de cURL, nous récupérons ici d'autres informations comme le code HTTP de la page crawlée, son poids ou encore son type de contenu (avec `charset` si renseigné). La fonction `cURL()` renvoie donc un tableau à quatre entrées :

- indice `[0]` : contenu de la page ;
- indice `[1]` : code HTTP de la page ;
- indice `[2]` : content-type de la page ;
- indice `[3]` : poids de la page.

```
// Fonction de récupération des contenus
function cURL($url = '', $timeout = 0) {
if(!empty($url)) {
    // Ajout du protocole s'il est manquant
    if(!preg_match("#^https?://#iU", $url)) {
        $url = "http://".$url;
    }

    // Récupération des données avec cURL
    $curl = curl_init($url);
    $userAgent = $_SERVER["HTTP_USER_AGENT"];
    curl_setopt($curl, CURLOPT_USERAGENT, $userAgent);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($curl, CURLOPT_CONNECTTIMEOUT, $timeout);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    curl_setopt($curl, CURLOPT_FAILONERROR, 1);

    // Récupération du contenu
    $contenu = curl_exec($curl);

    // Récupération des données
    $codeHTTP = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    $contentType = curl_getinfo($curl, CURLINFO_CONTENT_TYPE);
    $poids = curl_getinfo($curl, CURLINFO_SIZE_DOWNLOAD);
    curl_close($curl);

    $tab = array($contenu, $codeHTTP, $contentType, $poids);
```

```
    return $tab;
}
}
```

Récupérer d'autres informations que les liens

L'alternative est de modifier la fonction `extractionLiens()` du robot pour récupérer d'autres données à la volée. Vous pouvez aussi collecter des informations en créant des fonctions intermédiaires lancées via le robot. Ici, le choix a été porté sur des fonctions distinctes du robot par commodité et économie de ressources (nous pouvons traiter les données dans un second temps et non lors du crawl).

Récupérer des liens externes

Pour récupérer les liens externes, il existe plusieurs méthodes, l'étape la plus importante étant l'expression régulière qui permet d'extraire les données. Ici, nous allons programmer une fonction destinée à récupérer les liens externes provenant d'une liste d'URL à crawler. Cette fonction fera appel à la fonction `cURL()` que nous venons de présenter.

La fonction `liensExternes()` prend deux paramètres : un tableau d'URL (récupéré par notre robot) et le nom de domaine à exclure (pour que le code ne conserve que les liens externes). Elle retourne un listing des liens externes dans un tableau. Nous pouvons une fois encore les enregistrer par nos propres moyens pour analyser les données ultérieurement.

```
function liensExternes($tabLiens = array(), $domaine = '') {
foreach($tabLiens as $lien) {
    // Récupération du contenu des pages
    $contenu = cURL($lien);
    $contenu = $contenu[0]; // Contenu retourné par cURL()

    // Remplacement du nom de domaine par "#" (pour supprimer les liens internes)
    if(!empty($domaine)) {
        $contenu = str_ireplace($domaine, "#", $contenu);
    }

    // Extraction des liens différents de "#"
    $regex = "#href=[<\"'](https?://[^\#]+)[\"'>]#iU";
    preg_match_all($regex, $contenu, $resultat);

    // Tableau des liens externes (par page crawlée)
    $tabExternes[] = $resultat[1];
}

// Rassemble les sous-tableaux et efface les doublons
foreach($tabExternes as $liens) {
    foreach($liens as $lien) {
        $tab[] = $lien;
    }
}
```

```
$external = array_unique($tab);
// Classe les résultats
sort($external);
return $external;
}
```

Pour aller encore plus loin dans l'extraction, nous pourrions aussi optimiser cette fonction pour détecter les attributs `rel` et savoir si les liens externes sont en follow ou nofollow, par exemple.

La fonction suivante est un exemple basé sur la fonction précédente et qui peut être largement optimisé. Dans ce cas, la fonction ne retourne pas un tableau des liens externes mais une liste de tableaux contenant en premier indice le lien externe et en second indice la valeur de l'attribut `rel`. Il faut utiliser une boucle `foreach()` pour récupérer la liste des valeurs et les enregistrer dans des fichiers ou dans des bases de données.

```
function liensExternesRel($tabLiens = array(), $domaine = '') {
foreach($tabLiens as $lien) {
    // Récupération du contenu des pages
    $contenu = cURL($lien);
    $contenu = $contenu[0]; // Contenu retourné par cURL()

    // Remplacement du nom de domaine par "#" (pour supprimer les liens internes)
    if(!empty($domaine)) {
        $contenu = str_ireplace($domaine, "#", $contenu);
    }

    // Extraction des liens différents de "#"
    // $regex = "#href=[<\"'](https?://[^\#]+)[\"'>]#iU";
    $regex = "#<a(.*)>#iU";
    preg_match_all($regex, $contenu, $resultat);

    // Tableau des liens externes (par page crawlée)
    $tabExternes[] = $resultat[1];
}

// Rassemble les sous-tableaux et efface les doublons
foreach($tabExternes as $liens) {
    foreach($liens as $lien) {
        // Récupère les URL externes
        $regex = "#href=[<\"'](https?://[^\#]+)[\"'>]#iU";
        preg_match($regex, $lien, $urlLien);
        if(!empty($urlLien)) {
            $url = $urlLien[1];
        } else {
            $url = '';
        }
        // Récupère les attributs rel (follow/nofollow...)
        $regex = "#rel=[<\"'](.*)[\"'>]#iU";
        preg_match($regex, $lien, $rel);
        if(!empty($rel)) {
```

```
            $rel = $rel[1];
        } else {
            $rel = '';
        }
        // Ajoute l'URL et le rel dans un tableau
        // NB : séparé par un signe "#" pour découper ensuite
        if(!empty($url)) {
            $liste[] = $url.'#'.$rel;
        }
    }
}
// Dédoublonne la liste des liens + rel
$external = array_unique($liste);

// Classe les résultats
sort($external);
foreach($external as $lien) {
    $tab[] = explode('#', $lien);
}
return $tab;
}
```

Figure 5-46

Listing des liens internes et externes

Nombre de liens internes : 6

```
Array
(
    [0] => http://www.mathieu-chartier.com/
    [1] => http://www.mathieu-chartier.com/activites.php
    [2] => http://www.mathieu-chartier.com/competences-mathieu-chartier.php
    [3] => http://www.mathieu-chartier.com/contact.php
    [4] => http://www.mathieu-chartier.com/experience-mathieu-chartier.php
    [5] => http://www.mathieu-chartier.com/livre-mathieu-chartier.php
)
```

Nombre de liens externes : 33

```
Array
(
    [0] => http://annuaire.internet-formation.fr
    [1] => http://blog.internet-formation.fr
    [2] => http://blog.internet-formation.fr/2013/03/presentation-de-gestion-de-tarifs-extension-wordpress/
    [3] => http://blog.internet-formation.fr/2013/07/tag-to-link-extension-wordpress-pour-le-pagerank-sculpting/
    [4] => http://blog.internet-formation.fr/2013/07/wp-planification/
    [5] => http://blog.internet-formation.fr/2013/09/moteur-de-recherche-php-objet-poo-complet-pagination-surlignage-fulltext/
    [6] => http://blog.internet-formation.fr/2013/10/wp-advanced-search-moteur-de-recherche-avance-pour-wordpress/
    [7] => http://blog.internet-formation.fr/2013/10/wp-excerpt-generator/
    [8] => http://www.centre-presse.fr
    [9] => http://www.e-orientations.com
    [10] => http://www.editionsfirst.fr
    [11] => http://www.editionsfirst.fr/catalogue/1601-business/1602-entreprise/le-guide-complet-des-reseaux-sociaux-EAN9782754056052.html
    [12] => http://www.editionsfirst.fr/catalogue/1601-business/1602-entreprise/le-guide-du-referencement-web-EAN9782754049405.html
    [13] => http://www.evigeo.com
    [14] => http://www.facebook.com/EvigeoFormation
    [15] => http://www.facebook.com/pages/Internet-Formation-centre-de-formation-web-%C3%A0-Poitiers/165141539520
    [16] => http://www.gwanyong-taekwondo.com
    [17] => http://www.internet-formation.fr
    [18] => http://www.internet-formation.fr/inc/Internet-Mathieu-Chartier-auteur-SEO-8VO.mp3
    [19] => http://www.lanouvellerepublique.fr
    [20] => http://www.lanouvellerepublique.fr/Loir-et-Cher/Communautes-NB/n/Contenus/Articles/2013/04/30/Un-guide-pour-arriver-le-premier-sur-Google-1430924
    [21] => http://www.lereferenceur.fr/blog/interview-de-mathieu-charte-auteur-du-guide-du-referencement-web/
    [22] => http://www.linkedin.com/pub/mathieu-chartier/50/388/187
    [23] => http://www.miss-seo-girl.com/le-guide-du-referencement-web-interview-mathieu-chartier/
    [24] => http://www.oracom.fr
    [25] => http://www.radiovaldor.com/
    [26] => http://www.usinenouvelle.com
    [27] => http://www.viadeo.com/fr/profile/mathieu.chartier4
    [28] => https://plus.google.com/102466592139657070914/
    [29] => https://plus.google.com/102466592139657070914?rel=author
    [30] => https://plus.google.com/104686176004177830700
    [31] => https://plus.google.com/112254264255724316803
    [32] => https://twitter.com/Formation_web
)
```

Détecter les liens morts et les redirections

La fonction `cURL()` que nous avons mise en place au début de cette section récupère dynamiquement le code HTTP de la page crawlée. Il suffit donc de procéder à une boucle pour chaque lien de la liste récupérée par le robot pour obtenir le statut HTTP des pages web.

Nous devons obtenir un code 200 pour les pages valides, 404 pour les pages introuvables ou encore 301 pour les redirections. Il nous suffit d'interpréter ces données pour savoir à quel type d'erreur nous sommes confronté.

La fonction suivante est donc très simple. Pour éviter d'avoir une liste de codes, le script ajoute entre parenthèses l'URL crawlée. Il suffit alors d'utiliser la fonction `print_r()` de PHP pour afficher le statut de la page avec l'URL associée.

Bien entendu, l'idéal serait de modifier la fonction pour stocker toutes les indications de la fonction `cURL()` dans une base de données, cela éviterait de multiplier les fonctions diverses…

```
function httpCodes($tabLiens = array()) {
    if(!empty($tabLiens)) {
        foreach($tabLiens as $lien) {
            $codeHTTP = cURL($lien);
            // Code HTTP de cURL()
            $codes[] = $codeHTTP[1]." (".$lien.")";
        }
    }
    return $codes;
}
```

Vérifier les attributs alt des images

Dans notre logique de suivi des contenus, il peut être intéressant de vérifier la présence ou non des attributs `alt` des images, mais aussi leur contenu. Pour rappel, cet attribut est obligatoire dans les règles conseillées par le W3C, mais il arrive fréquemment qu'il soit omis ou vide. Il convient donc de vérifier la présence de ces attributs.

Les mots-clés contenus dans les textes alternatifs ayant plus de poids pour les moteurs de recherche, il ne faut jamais négliger cet aspect et oublier de remplir soigneusement ces attributs.

Comme pour les autres fonctions, il existe en réalité pléthore de méthodes pour procéder au suivi des attributs `alt`. Pour l'exemple, nous utilisons ici la liste fournie par notre robot d'indexation mais la technique employée pour la vérification des balises `<title>` et des métadonnées présentée au début de cette section serait sûrement plus rapide et efficace…

La fonction `displayALT()` affiche la liste des images et textes de remplacement pour chaque lien crawlé. Ainsi, nous pouvons avoir un rapide coup d'œil des attributs et valeurs manquantes dans nos pages web. Ensuite, il ne nous reste plus qu'à remplir et optimiser les textes absents. Voici le code de la fonction :

```
function displayALT($tabLiens = array()) {
foreach($tabLiens as $lien) {
    // Récupération du contenu des pages
    $contenu = cURL($lien);
    $contenu = $contenu[0]; // Contenu retourné par cURL()

    $regex = "#<img(.*)/?>#iU";
```

```
    preg_match_all($regex, $contenu, $tabALT);

    $resultat = "<big><strong>URL : </strong>".$lien."</big> <br/>\n";
    $resultat.= "<ol>\n";

    foreach($tabALT[0] as $uniqueALT) {
        // Extraction des attributs alt de chaque image par page
        $regex = "#(alt)=['\"](.*)['\"]#iU";
        preg_match($regex, $uniqueALT, $tableau);

        // Récupération de la balise d'image complète (repère)
        $resultat.= "<li><em>".htmlspecialchars($uniqueALT)."</em></li>\n";

        // Affichage personnalisé
        if(!empty($tableau[1])) {
            if(!empty($tableau[2])) {
                $resultat.= "<span style='color:green'>alt OK</span> : ";
                $resultat.= $tableau[2]."<br/><br/>\n";
            } else {
                $resultat.= "<span style='color:red'>alt vide !</span><br/><br/>\n";
            }
        } else {
            $resultat.= "<span style='color:red; font-weight:bold'>alt manquant !
            </span><br/><br/>\n";
        }
    }
    $resultat.= "</ol>\n";

    echo $resultat;
}
}
displayALT($liste);
```

Figure 5-47

Vérification des attributs alt des images

Check-list de l'audit SEO

Voici une petite liste récapitulative pour vous aider à mener à bien un audit SEO. Bien évidemment, tous les critères ne sont pas présentés dans la liste et vous pouvez en ajouter autant que vous le désirez, tout dépend du site que vous devez analyser et des objectifs que vous vous fixez au préalable. En règle générale, les principaux facteurs exposés dans cet exemple reviennent pour tous les audits de référencement, mais parfois, ils sont encore plus fournis, encore plus techniques afin d'aller chercher d'autres spécificités.

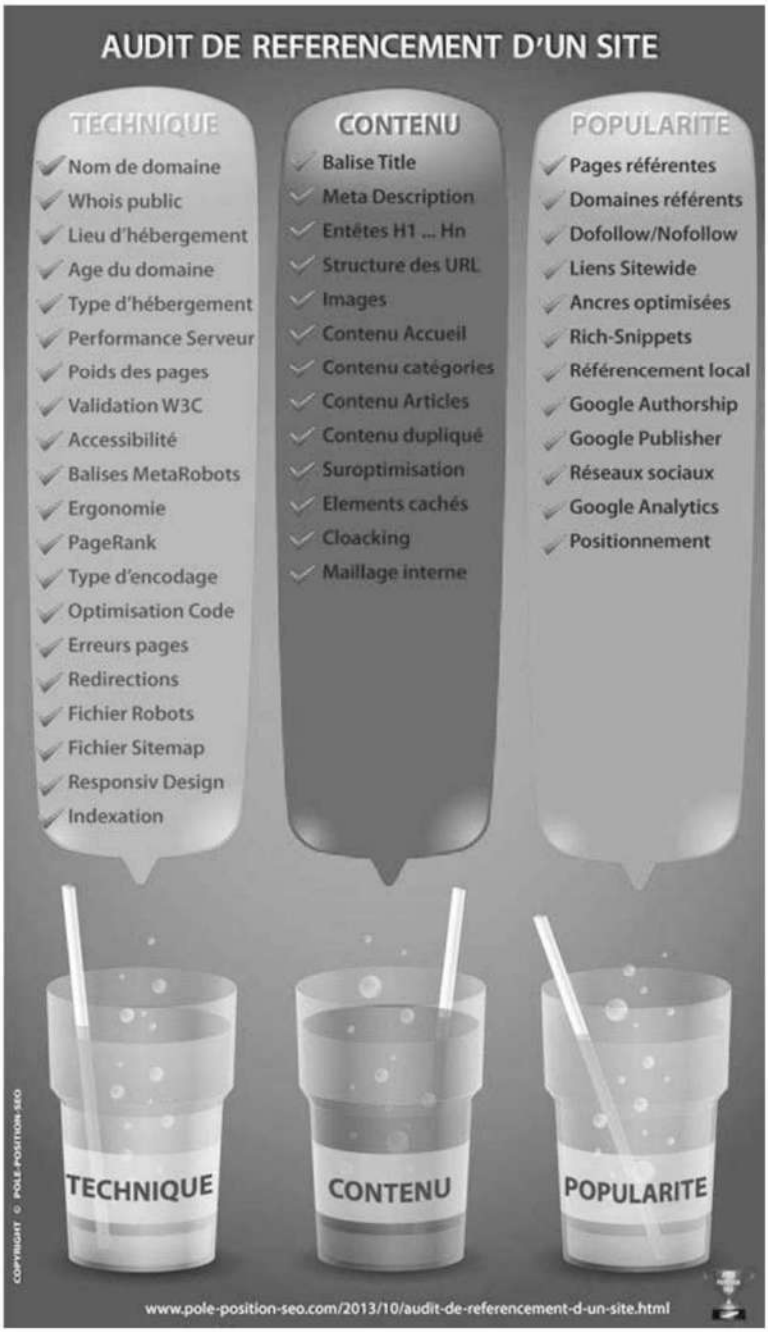
Critères techniques	**OK**	**Pas OK**	**Commentaires**
Nom de domaine ?			
EMD - Exact Match Domain ? - Âge du domaine – ancienneté - Whois anonyme ou pas - Date de création et d'expiration du NDD - Historique du NDD (Wayback machine)			
Présence d'un fichier `robots.txt` ?			
Présence d'un fichier `sitemap.xml` ?			
Présence des balises meta `robots` ?			
Réécriture des URL activée ?			
URL friendly (URL statiques ou dynamiques avec « ? » ou « & », ID de sessions...) ?			
Type de réponses HTTP (200, 404, 500, 410...) ?			
Temps réponse serveur > 1 seconde (test avec WebSitePulse, par exemple) ?			
PageRank / BrowseRank ?			
Encodage (UTF-8 ou ISO 8859-1 en France) ?			
Comptabilité et validité W3C ?			
Temps de chargement des pages (GTMetrix, Pingdom...) ?			
Présence d'erreurs 404 (crawl avec Xenu, par exemple) ?			
Page d'erreur 404 personnalisée ?			
Redirections 301 (noms de domaines, pages miroirs, doublons...) ?			
Poids des fichiers multimédias et web (images, vidéos, PDF, pages web...) ?			
Le site fonctionne-t-il avec et sans les www **?**			
Compatibilité sur les différents navigateurs du marché (mobiles et fixes) ?			
Compatibilité sur les différents supports mobiles (site adaptatif, site mobile, AMP HTML...).			
Applications mobiles : l'App Indexing est-il mis en place ?			
Feuilles de styles CSS externalisées ?			
Mise en place de rich snippets (microdonnées, microformats ou RDFa) ?			
Logo cliquable menant vers la page d'accueil ?			
Fil d'Ariane présent et fonctionnel ?			
Plan de site disponible ?			
Structure de site intuitive avec navigation simplifiée ?			
Présence de trop de publicités au-dessus la ligne de flottaison ?			
Netlinking	**OK**	**Pas OK**	**Commentaire**
Liens entrants (Ahrefs, Open Site Explorer...) ? Nombre total de liens Nombre de domaines référents Ancres			

Ratio follow/nofollow de qualité ?			
Qualité des liens et du PageRank ?			
Liens présents sur toutes les pages (sitewide) ou non (not sitewide) ?			
Nombre de liens par page limité ?			
Contenu	**OK**	**Pas OK**	**Commentaire**
Contenu cohérent et pertinent ?			
Pages monothématiques ?			
Qualité des contenus (orthographe, grammaire, valeur ajoutée...) ?			
Contenus régulièrement mis à jour ?			
Pertinence des mots-clés principaux ?			
Pertinence de la longue traîne ?			
Ratio texte/illustrations ?			
Contenu dupliqué (Positeo, Copyscape...) ?			
Balises `<title>` uniques relatives aux contenus de chaque page ?			
Balises `<title>` de moins de 70 caractères (taille maximale conseillée) ?			
Liens internes cohérents avec le contenu ?			
Balises meta `description` optimisées ?			
Utilisation idéale des balises sémantiques `<hn>` (`<h1>` à `<h6>`) ?			
Présence d'un bourrage de mots-clés (keyword stuffing) ?			
Longueur de texte suffisante ?			
Visuels cohérents avec le texte ?			
Présence d'autres types de fichiers multimédias (vidéo, audio, PDF...) ?			
Hiérarchisation logique et maîtrisée des contenus ?			
L'ensemble est-il lisible et attractif ?			
Images Images uniques et optimisées : TinEye Attributs `alt` et `title` Pertinence des illustrations Optimisation de la taille des images Présence d'une favicon (.ico ou .png en général)			
Rubriques et sous-rubriques facilement identifiables et explicites ?			
Arborescence claire et efficiente ?			
Nom de domaine simple et facilement mémorisable ?			
Nom de domaine court et URL optimisées ?			
Les menus de navigation sont-ils accessibles sur toutes les pages ?			
Est-il toujours possible de revenir à l'accueil ?			
Pertinence des intitulés des menus et sous-rubriques (navigation intuitive) ?			
Social et interactivité	**OK**	**Pas OK**	**Commentaire**
Présence de boutons de partage sur les principaux réseaux sociaux ?			
Présence sociale (pages ou profils Facebook, Twitter, Google+...) ?			
Maîtrise et usage des flux de syndication (RSS) ?			
Présence et possibilité de poster des commentaires ?			
Présence d'un blog ou d'une section Actualités ?			
Présence ou possibilité de s'inscrire à une newsletter ?			
Mise en place de l'AuthorShip ?			
Notoriété globale sur les réseaux sociaux ?			

Résumons tous ces facteurs grâce à une infographie complète réalisée par le site : www.pole-position-seo.com.

Figure 5-48

Résumé graphique de tous les critères utiles pour réaliser un audit SEO de qualit



Annexe

Sources de veille SEO

Vous trouverez dans cette webographie une liste non exhaustive d'outils et de logiciels relatifs au référencement et aux spécialités attenantes afin d'optimiser au mieux vos sites web.

Ressources techniques

- AlsaCreations : http://www.alsacreations.com
- ASP.net : http://www.asp.net
- Mozilla Developer : https://developer.mozilla.org/fr/docs/Web
- Developpez.com : http://www.developpez.com
- Manuel PHP : http://www.php.net/manual/fr
- Microsoft Visual Studio : http://msdn.microsoft.com/fr-fr/fr/vstudio/hh341490.aspx
- Open Classrooms : http://fr.openclassrooms.com
- Python : http://webpy.org
- W3Schools : http://www.w3schools.com

Interfaces pour les webmasters

- Baidu (en chinois) : http://zhanzhang.baidu.com
- Bing : http://www.bing.com/toolbox/webmaster
- Google : https://www.google.com/webmasters/tool
- Yandex : http://webmaster.yandex.com

Documentation et blogs officiels des moteurs de recherche

Les sources officielles font souvent office de référence à la fois pour rester dans les guidelines mais aussi pour apprendre parfois quelques subtilités propres aux différents moteurs.

- Ask.com : http://blog.ask.com
- Baidu Beat : http://beat.baidu.com
- Bing Blogs : http://blogs.bing.com/webmaster
- Exalead : http://blog.exalead.fr
- Google Official Blog : http://googleblog.blogspot.fr
- Google Webmaster: http://googlewebmastercentral.blogspot.fr
- Qwant – Le blog : http://blog.qwant.com
- Yandex (en russe) : http://blog.yandex.ru

Antipénalités, réexamen et vie privée

Voici des outils pour lutter contre le spam et protéger sa vie privée, ou pour réaliser des demandes de réexamen en cas de pénalités.

- Demande de réexamen sur Google : http://goo.gl/Fcwr57
- Désavouer les liens sur Bing : http://goo.gl/jb8M7G
- Désavouer des liens sur Google : http://goo.gl/FkIgAH
- Droit à l'oubli sur Google : http://goo.gl/k896dx
- EWhois : http://www.ewhois.com
- Panguin Tool : http://www.panguintool.com
- Serpomètre de Ranks.fr : http://www.ranks.fr/fr/serpometre
- Spam Report de Google : http://goo.gl/1CMib4
- Spy On Web : http://spyonweb.com
- Support Bing : https://support.discoverbing.com/eform.aspx
- Support Google : http://goo.gl/8mvCMz
- You Get Signal : http://goo.gl/CxLY1K

Soumission manuelle aux moteurs de recherche

Si vous souhaitez soumettre vos sites web aux différents moteurs de recherche, voici les adresses dont vous avez besoin…

- Baidu : http://zhanzhang.baidu.com/sitesubmit/index
- Bing : http://www.bing.com/toolbox/submit-site-url

- Exalead : http://www.exalead.com/search/web/submit
- Google : https://www.google.com/webmasters/tools/submit-url
- Voila/Orange : http://referencement.ke.voila.fr
- Yandex : http://webmaster.yandex.com/addurl.xml
- Yahoo! : http://search.yahoo.com/info/submit.html

Sources généralistes sur le référencement

Voici une liste de ressources utiles ou de référence pour suivre l'actualité de la discipline.

- Abondance : http://www.abondance.com
- AxeNet : http://blog.axe-net.fr
- Blog Internet-Formation : http://blog.internet-formation.fr
- Bruce Clay : http://www.bruceclay.com/blog
- Frères Peyronnet : http://www.peyronnet.eu/blog
- Google XXL : http://googlexxl.blogspot.fr
- Laurent Bourrelly : http://www.laurentbourrelly.com/blog
- Matt Cutts (blog officiel) : http://www.mattcutts.com/blog
- Miss SEO Girl : http://www.miss-seo-girl.com
- MoteurZine : http://www.moteurzine.com
- Moz : http://moz.com/blog
- PullSEO : http://www.pullseo.com/societe/actualites
- Search Engine Land : http://searchengineland.com
- Search Engine Watch : http://www.searchenginewatch.com
- Secrets2Moteurs : http://www.secrets2moteurs.com
- SEO Camp : http://www.seo-camp.org
- Seolius : http://www.seolius.com/dossiers
- SeoMix : http://www.seomix.fr
- WebRankInfo : http://www.webrankinfo.com
- Ya-Graphic : http://www.ya-graphic.com
- Yooda : http://blog.yooda.com

Baromètres, études chiffrées et statistiques

Si votre objectif est d'être incollable sur les chiffres concernant la SEO, ces sites sont faits pour vous.

- AT Internet : http://www.atinternet.fr/ressources/ressources

- comScore : http://www.comscore.com
- Hitwise d'Experian : http://www.experian.com/hitwise
- Journal du Net : http://www.journaldunet.com/web-tech
- Live Internet : http://www.liveinternet.ru/stat/ru/searches.html
- Médiamétrie : http://www.mediametrie.fr/internet
- Miratech : http://miratech.fr
- Searchmetrics : http://www.searchmetrics.com
- StatCounter Global Stats : http://gs.statcounter.com
- Tiobe : http://goo.gl/fP5p4X

Simulateurs de robots d'indexation

Si vous souhaitez voir votre site comme un robot, testez l'un des outils suivants :

- Internet-Formation : http://goo.gl/SBMnjS
- IWebTool : http://goo.gl/Z2Obqw
- SEO Chat : http://goo.gl/n5Yrwr
- Small SEO Tools : http://goo.gl/vy5ovG
- ToTheWeb : http://goo.gl/xI5IR3
- WebConfs : http://goo.gl/vy5ovG
- Webmaster Toolkit : http://goo.gl/Bas3C9
- YatooWeb : http://goo.gl/5ulSVh

Outils d'analyse des liens

L'analyse du profil de liens est primordiale pour faire évoluer le PageRank Google, le BrowseRank de Bing, le LinkRank de Qwant et tant d'autres tout en évitant des pénalités ou la présence de liens morts. Ces outils sont indispensables…

- AdvancedLinkManager : http://advancedlinkmanager.com
- Ahrefs : http://ahrefs.com
- Analyse Backlinks : http://www.analyzebacklinks.com
- Backlink Watch : http://www.backlinkwatch.com
- Explorer : http://explorer.cognitiveseo.com
- Link Diagnosis : http://www.linkdiagnosis.com
- Link Examiner : http://goo.gl/wjTf9o
- Majestic SEO : http://www.majesticseo.com

- Open Site Explorer : http://www.opensiteexplorer.org
- Outils Référencement : http://www.outils-referencement.com/outils
- SEO Profiler : http://www.seoprofiler.com
- W3C Link Checker : http://validator.w3.org/checklink
- Xenu : http://home.snafu.de/tilman/xenulink.html

Outils de recherche de mots-clés

La recherche de mots-clés est la première étape dans la mise en place d'un bon référencement, ces services en ligne et logiciels sont idéaux pour trouver les perles rares.

- Keyword Country : http://www.keywordcountry.com
- Keyword Discovery : http://goo.gl/2Rvq6Z
- Keyword Planner : http://www.google.com/sktool/
- Keyword Spy : http://www.keywordspy.com
- Keyword Suggest : http://goo.gl/JusV5n
- KGen : http://kgen.elitwork.com
- KwMap : http://www.kwmap.com
- Search Combination Tool : http://goo.gl/7XJ9ss
- SeCockpit : http://www.secockpit.fr
- Self SEO : http://www.selfseo.com/keyword_typo_generator.php
- Wordtracker : https://freekeywords.wordtracker.com
- Wordze : http://www.wordze.com

Outils d'analyse des contenus et des mots-clés

Dans la lignée des outils présentés précédemment, ces quelques adresses vous dirigeront vers des services d'analyse approfondie des contenus dans les pages web, ce qui s'avère souvent indispensable.

- Alyze : http://alyze.info
- Integrity (Mac OS) : http://peacockmedia.co.uk/integrity
- Outiref : http://www.outiref.com
- Screaming Frog : http://www.screamingfrog.co.uk/seo-spider
- Yakaferci : http://www.yakaferci.com
- Yooda Match Density : http://goo.gl/n0ydxP

Audit SEO, aide et suivi

Voici une liste d'outils de qualité à utiliser pour surveiller le référencement et suivre l'état du positionnement.

- Agent Web Ranking : http://www.agentwebranking.fr
- Allorank : http://www.allorank.com
- Botify : https://www.botify.com
- CrawlTrack : http://www.crawltrack.fr
- DareBoost : https://www.dareboost.com/fr/home
- Gamma SEO Tools : http://www.gammaseotools.com
- Grader : http://grader.rezoactif.com
- Gunning Fox Index : http://gunning-fog-index.com
- Myposeo : https://www.myposeo.com
- Netstorming : http://www.netstorming.fr
- Not Provided Kit : http://notprovidedkit.com
- Optimiz.Me : http://optimiz.me
- OnCrawl : http://fr.oncrawl.com
- Positeo : http://www.positeo.com/check-position
- Ranks.fr : http://www.ranks.fr
- Rank Tracker : http://www.link-assistant.com/rank-tracker
- SeeUrank : http://www.yooda.com/produits/soft
- SEMrush : http://www.semrush.com
- SEO Administrator : http://goo.gl/aXoVnw
- SEO Chat : http://tools.seochat.com
- SEOh : http://www.seoh.fr/audit-seo
- SeoMioche : http://www.seomioche.com
- SEOscope : http://www.seoscope.fr
- SEO Soft : http://goo.gl/XNn2Ds
- SiteAnalyzer : http://www.site-analyzer.com/fr
- SpyWords : http://www.spywords.com
- Track-Flow : http://www.cybercite.fr/track-flow.html
- WebRankChecker : http://www.webrankchecker.com
- Woorank : http://www.woorank.com

Outils antiplagiat et duplicate content

Il est primordial de surveiller constamment l'existence de contenus dupliqués ou de problème de DUST afin d'éviter des sanctions mais aussi pour se protéger du droit d'auteur. Cette liste d'outils sera votre meilleure arme pour lutter contre les fraudes.

- Copyscape : http://www.copyscape.com
- DustBall : http://www.dustball.com/cs/plagiarism.checker
- KillDC : http://killdc.linkomatic.org
- NoPlagiat : http://www.noplagiat.com
- Plagiarism Checker : http://www.plagiarismchecker.com/url
- Plagiarism Detector : http://www.plagiarism-detector.com
- Plagium : http://www.plagium.com
- PlagScan : http://www.plagscan.com/fr
- PlagSpotter : http://www.plagspotter.com
- Plagtracker : http://www.plagtracker.com
- Positeo : http://www.positeo.com/check-duplicate-content

Analyse du PageSpeed et de la vitesse de chargement

Ces services en ligne peuvent vous aider à étudier et optimiser la vitesse de chargement de vos pages web ainsi que les critères des PageSpeed et YSlow.

- GTMetrix : http://gtmetrix.com
- IMN Page Speed Tool : http://goo.gl/R5Os6H
- Load Impact : http://loadimpact.com
- PageSpeed Insights : http://goo.gl/adG6eE
- Pingdom : http://tools.pingdom.com/fpt
- ShowSlow : http://www.showslow.com
- SiteTimer : http://www.octagate.com/service/SiteTimer
- Small SEO Tools : http://goo.gl/S3dq1X
- SnapHost : http://goo.gl/XRarsF
- Web Page Analyzer : http://goo.gl/it34KN
- WebSite Pulse : http://www.websitepulse.com
- Webwait : http://webwait.com
- Yslow : https://developer.yahoo.com/yslow/

Réseaux sociaux

Une boîte à outils pour les réseaux sociaux, de la simple analyse de mots-clés au suivi de la notoriété.

- BrandWatch : https://www.brandwatch.com
- CircleCount : http://www.circlecount.com
- Empire Avenue : http://www.empireavenue.com
- How Sociable : http://www.howsociable.com
- Klear : http://klear.com
- Klout : http://klout.com
- Kred : http://kred.com
- Reppler : http://www.reppler.com
- Social Bakers : http://www.socialbakers.com
- Social Crawlytics : https://socialcrawlytics.com
- SumAll : https://sumall.com/20ft
- Who's Talking : http://www.whostalkin.com

Index

S

T

U

V

W

X

Y