

INFORMATIQUE

Enseignement Secondaire Général

KAMAHA Théophile Alain

Elève-Professeur en 5ème année Informatique de l'ENS de Yaoundé

Licencié en Biochimie

MENGUE ANGO Christelle

Elève-Professeur en 5ème année Informatique de l'ENS de Yaoundé

Licencié en Droit

MENGUE Annie Nadège

Elève-Professeur en 5ème année Informatique de l'ENS de Yaoundé

Licencié en Lettres Modernes Françaises

MOYOU METCHEKA Léonel

Elève-Professeur en 5ème année Informatique de l'ENS de Yaoundé

Licencié en Informatique

Avant - propos

Objectifs de l'ouvrage

Le projet de rédaction de cet ouvrage répond à un triple besoin :

- L'indisponibilité d'un manuel scolaire conforme aux exigences du nouveau programme d'informatique en terminale de l'enseignement général secondaire, sauf pour ce qui est de quelques projets fort bien initiés par l'Ecole Normale Supérieure de Yaoundé ;
- L'inadaptation des manuels existants à l'approche par compétences qui s'impose aujourd'hui ;
- La forte dépendance des contenus des manuels existants aux technologies présentes et souvent dépassées

Face à ces besoins, cet ouvrage poursuit les objectifs suivants :

- Mettre à disposition des élèves et des enseignants un manuel conforme au nouveau programme et à l'approche par compétences ;
- Consolider les savoir-faire par des activités et des projets de consolidation ;
- Introduire une logique invariante par des contenus transposables à toutes les technologies présentes et futures.

D'autre part, afin de coller aux objectifs de la pédagogie par intégration des acquis, le langage PHP a été introduit pour faire de chaque thème développé une partie d'un tout cohérent.

Structure de l'ouvrage

Cet ouvrage s'articule autour des thèmes abordés lors de la réalisation et la publication d'un site Web dynamique intégrant les technologies JavaScript et PHP, le tout exploitable en local à travers un réseau informatique filaire ou sans fil.

Un module introductif, intitulé *Rappels* présente les notions de base d'exploitation de l'ordinateur et de conception de page web. Il est structuré ainsi qu'il suit :

- Le *Chapitre 1 : Environnement informatique*, rappelle les notions de base sur l'ordinateur (hardware et software) ;

- Le *Chapitre 2 : Le langage HTML*, revient sur l'utilisation du HTML pour la création de pages Web;

Le premier module, intitulé *Les réseaux informatiques* présente les concepts de réseaux informatiques pour déboucher sur la mise en place d'un réseau filaire couplé à un réseau sans fil. Il est structuré ainsi qu'il suit :

- Le *Chapitre 3 : Création d'un réseau local*, introduit les notions de base sur les réseaux informatiques et leur classification, ainsi que les éléments nécessaires à l'interconnexion des ordinateurs dans un réseau filaire ou sans fil ;
- Le *Chapitre 4 : Configuration d'un réseau local*, présente le protocole TCP/IP et les techniques d'adressage qui permettent aux ordinateurs de communiquer ;
- Le *Chapitre 5 : Montage d'un réseau local*, véritable application des deux premiers chapitres, il permet de mettre en pratique les techniques de montage d'un réseau local, depuis le sertissage des câbles jusqu'au test du réseau ;
- Le *Projet de Consolidation 1*, est un protocole à suivre par les élèves pour le montage d'un réseau local filaire, éventuellement couplé à un réseau sans fil.

Le deuxième module, intitulé *Programmation interactive de pages Web* permet de compléter les notions de création de page Web statique étudiées en classe de première en y ajoutant de l'interactivité :

- Le *Chapitre 6 : Introduction à JavaScript*, présente le langage HTML pour la création des pages web statiques, puis aborde l'importance, les limites et la sécurité de JavaScript ;
- Le *Chapitre 7 : Le langage JavaScript*, présente la syntaxe de JavaScript, pour écrire efficacement des scripts exécutables côté client;
- Le *Chapitre 8 : Intégration de JavaScript à HTML*, aborde les différentes méthodes qui permettent d'intégrer de l'interactivité à une page Web ;
- Le *Projet de consolidation 2*, est un protocole à suivre par les élèves pour concevoir et publier un site Web de huit pages Web qui propose des exercices sous forme de QCM. Des scripts Javascript permettront de calculer le nombre de bonnes réponses et une note sur 20. Ce site sera publié en local sur le réseau du projet de consolidation 1 ;

Le troisième module, intitulé *Programmation dynamique de pages Web* permet de compléter le module précédent en ajoutant du dynamisme aux pages Web. Sa structure est la suivante :

- Le *Chapitre 9 : Introduction aux bases de données*, aborde les notions de stockage d'informations dans des collections organisées de données ;
- Le *Chapitre 10 : Le langage SQL*, présente les techniques de création et de gestion des bases de données

- Le *Chapitre 11 : Introduction à PHP*, aborde les concepts de base de création de scripts exécutables coté serveur ;
- Le *Chapitre 12 : Accès aux données avec PHP*, présente les techniques de base pour interfaçer une page Web à une base de données ;
- Le *Projet de consolidation 3*, est un protocole à suivre par les élèves pour ajouter au site Web du projet de consolidation 2, trois pages qui permettront d'enregistrer des élèves comme membre du site, et de réserver uniquement aux membres enregistrés l'accès aux pages de QCM

Les articulations de l'ouvrage sont ponctués d'exemples concrets qui sont facilement transposables aux activités des projets de consolidation. Chaque chapitre se termine par une série d'exercices (théoriques et/ou pratiques) qui permettent d'affiner son savoir et son savoir-faire, afin d'aborder au mieux les projets de consolidation. La réalisation des projets de consolidation va permettre d'atteindre progressivement l'objectif de création d'un site Web interactif et dynamique, publié localement sur un réseau filaire et/ou sans fil.

Typographie de l'ouvrage

La police générale de l'ouvrage est "Garamond Premier Pro". Toutefois, pour accentuer les textes de codes, la police "courier" a été utilisée et une bande rose a été ajoutée à la gauche.

```
function addition(a,b) {
c=a+b;
return c;
}
```

Les majuscules sont employées pour les directives SQL, et les minuscules pour les autres éléments. Les noms des tables et des champs sont précédés d'une majuscule (exemple : la table Eleve contient la colonne Matricule).

Dans une instruction SQL, le symbole « | » un choix et les symboles « [» et «] » précisent le caractère optionnel d'une directive au sein d'une commande (exemple :

```
SELECT [ALL]|[DISTINCT] [attributs]*
FROM liste_table
[WHERE] Condition1
[GROUP BY] liste_champ
[HAVING] condition2
[ORDER BY] liste_champ [ASC]|[DESC];
```

).

Plusieurs pictogrammes sont repérables pour attirer l'attention du lecteur :



Ils introduisent les définitions de mots clés qu'il faut retenir.



Ils introduisent des remarques pour préciser des abus de langage, des erreurs courantes sur les notions et des commentaires sur des éléments de syntaxe.



Ils introduisent des résumés des chapitres ;



Ils introduisent des exercices théoriques (restitution organisée des connaissances, analyse d'un problème);



Ils introduisent des exercices pratiques (qui peuvent être directement exécutables dans l'ordinateur) ;

Ressources complémentaires

L'ouvrage est accompagné d'un CD-ROM contenant les codes sources des pages Web présentées dans les activités des chapitres et les scripts SQL de création et de mise à jour de la base de données "lycee" utilisé dans le chapitre 10.

Remerciements

Nous tenons à remercier tous ceux qui nous ont aidés à rédiger ce livre, aux familles, proches et amis pour leur soutien et leur patience pendant ces longues semaines de réflexion et de rédaction.

A nos encadreurs, aux cotés desquels nous avons affiné notre connaissance des contenus nécessaires au développement des divers chapitres, nous exprimons notre gratitude.

A M. GOUIFFE A ABOUEM Franck, pour la formation et la conception graphique des maquettes.

Aux enseignants et élèves, qui se sont prêtés au jeu du test de cet ouvrage, et dont les retours ont permis d'améliorer l'aspect fonctionnel du document

Et à tous ceux que nous avons simplement oubliés dans la précipitation juste avant l'impression de cette page.

Merci à tous, car sans vous ce livre n'aurait peut-être pas vu le jour.

Table des matières

Avant - propos _____	i	<i>c. Les organes de sortie</i> _____	10
Objectifs de l'ouvrage _____	i	<i>i. L'imprimante</i> _____	10
Structure de l'ouvrage _____	i	<i>ii. L'écran</i> _____	10
Typographie de l'ouvrage _____	iii	<i>iii. Les autres organes de sortie</i> _____	11
Ressources complémentaires _____	iv	<i>d. Les organes de stockage</i> _____	11
Remerciements _____	iv	<i>i. Le disque dur</i> _____	11
		<i>ii. Les organes de sauvegarde amovibles</i>	11
			11
Module 0 : Rappels _____	2	<i>e. Les organes d'entrée et sortie</i> _____	12
		<i>f. Les organes de communication</i> _____	12
Chapitre 1 : Environnement informatique _____	3	<i>i. Le bus</i> _____	12
Activité _____	4	<i>ii. Le chipset</i> _____	12
I. Présentation Générale _____	5	<i>g. Les cartes d'extension</i> _____	12
1. Ordinateur et Informatique _____	5	2. Le software _____	13
2. Utilisation des ordinateurs _____	5	<i>a. L'exécution d'un programme</i> _____	13
II. Fonctionnement d'un ordinateur _____	6	<i>b. Les programmes et les fichiers</i> _____	13
1. Le hardware _____	6	3. Notions de base sur les logiciels _____	14
<i>a. L'unité centrale</i> _____	7	<i>a. Les logiciels systèmes</i> _____	14
<i>i. La carte mère</i> _____	7	<i>i. Fonctions d'un système d'exploita-</i>	15
<i>ii. Le processeur</i> _____	8	<i>tion</i> _____	15
<i>iii. La mémoire centrale</i> _____	8	<i>ii. Types de systèmes d'exploitation</i> _____	15
<i>b. Les organes d'entrée</i> _____	9	<i>b. Les logiciels d'application</i> _____	16
<i>i. Le clavier</i> _____	9	<i>c. Les utilitaires</i> _____	17
<i>ii. La souris</i> _____	9	<i>i. Les antivirus</i> _____	18
<i>iii. Les autres périphériques d'entrée</i> _____	9	<i>ii. Les autres utilitaires</i> _____	18

Chapitre 2 : Le langage HTML	21
I. Notion de site Web	23
II. Contenu d'une page Web	23
1. Notion de balise	23
2. Notion d'attribut de balise	24
3. Balises HTML usuelles	24
III. Exemple de page web	26
RESUME	28
Exercices	29

Module 1 : Les réseaux informatiques __ 30

Chapitre 3 : Création d'un réseau local	31
Activité	32
Introduction	33
I. Intérêt et impact de l'interconnexion en réseau	33
1. Intérêts de l'interconnexion en réseau	33
2. Impact de l'interconnexion en réseau	33
II. Les éléments d'un réseau	34
1. Les supports de transmission	34
a. <i>Les connexions câblées</i>	34
b. <i>Les connexions sans fil</i>	35
2. Les messages	35
3. Les périphériques du réseau	35
a. <i>Les périphériques finaux</i>	35
b. <i>Les périphériques intermédiaires</i>	36
4. Le protocole de communication	37
5. La méthode d'accès	37
a. <i>Les topologies physiques</i>	38

b. <i>Les topologies logiques</i>	41
III. Classification des réseaux	42
1. La taille de la couverture géographique	42
2. Nature de la liaison	43
a. <i>Les réseaux filaires</i>	43
b. <i>Les réseaux sans fil</i>	43
IV. Les architectures réseaux	44
1. L'architecture d'égal à égal	44
2. L'architecture client/serveur	45
3. L'architecture infrastructure	45
RESUME	46
Exercices	47

Chapitre 4 : Configuration d'un réseau local 49

Activité	50
Introduction	51
I. Le protocole de communication	51
1. La couche Application	52
2. La couche Transport	52
3. La couche Interréseau	52
4. La couche d'Accès au réseau	53
5. L'encapsulation des données	53
II. L'adressage des ordinateurs	54
1. L'adresse IP	55
a. <i>Les adresses publiques</i>	55
b. <i>Les adresses privées</i>	55
2. Le masque de sous-réseau	56
3. Le mode d'adressage	56
a. <i>Adressage statique</i>	56
b. <i>Adressage dynamique</i>	57
III. Test du réseau	57

RESUME _____	60	Activité _____	80
Exercices _____	61	I. Historique de JavaScript _____	81
Chapitre 5 : Montage d'un réseau local _____	63	II. Importance de JavaScript _____	82
Activité _____	64	III. Limites de JavaScript _____	82
I. Introduction _____	65	IV. Sécurité de JavaScript _____	83
II. Inventaire des éléments du réseau _____	65	V. Notion d'objet _____	84
III. Fabrication des câbles à paires torsadées _____	65	RESUME _____	85
1. Matériel utile à la fabrication du câble _____	65	Exercices _____	86
2. Processus de fabrication d'un câble à paires torsadées _____	66	Chapitre 7 : Le langage JavaScript _____	89
a. <i>Dénuder le câble</i> _____	66	Activité _____	90
b. <i>Agencer les fils</i> _____	67	I. Déclaration des variables _____	91
<i>Agencement pour un câble droit</i> _____	68	1. Déclaration sans affectation _____	91
<i>Agencement pour un câble croisé</i> _____	68	2. Déclaration avec affectation _____	92
c. <i>Sertissage de la prise RJ45</i> _____	69	3. Utilisation des variables _____	93
IV. L'interconnexion des périphériques du réseau _____	70	4. Opérations sur les variables _____	93
1. Interconnexion filaire _____	70	a. <i>Opérations arithmétiques</i> _____	93
2. Interconnexion sans fil _____	70	b. <i>Opérations de comparaison</i> _____	94
V. Installation du protocole de communication _____	71	c. <i>Opération sur chaînes de caractères</i> _____	95
VI. Adressage du réseau _____	71	d. <i>Opération d'affectation</i> _____	95
RESUME _____	72	II. Utilisation des structures de contrôle _____	95
Exercices _____	73	1. Les blocs d'instructions _____	95
Projet de consolidation 1 _____	77	2. Les structures alternatives _____	95
Module 2: Programmation interactive de pages Web _____	78	a. <i>L'alternative réduite</i> _____	96
Chapitre 6 : Introduction à JavaScript _____	79	b. <i>L'alternative complète</i> _____	96
		3. Les structures itératives _____	97
		a. <i>La structure While</i> _____	97
		b. <i>La structure Do...While</i> _____	98
		c. <i>La structure For</i> _____	99
		III. Utilisation des tableaux _____	100
		IV. Utilisation des fonctions _____	102

1. Notion de fonction et de procédure	102
2. Notion de méthode	103
V. Eléments d'interaction	104
1. Les fenêtres de dialogue	104
<i>a. Demande d'information</i>	104
<i>b. Alertes</i>	105
2. Les événements	106
3. Ajout de contrôles	107
4. Les cookies	108
5. Accès aux éléments de la page	108
<i>a. Accès par l'attribut «name»</i>	109
<i>b. Accès par l'attribut «id»</i>	110
RESUME	111
Exercices	112

Chapitre 8 : Intégration de JavaScript à HTML 123

Activité	124
I. Intégration directe au code HTML	126
1. Les balises <script>	126
2. Les pseudo-URL	127
3. Les gestionnaires d'événements	128
II. Intégration à un fichier externe	129
III. Code de la page d'activité avec scripts JavaScript des exemples	130
IV. Code de la fonction «corrige» de l'activité	132
V. Commentaires de la fonction «corrige»	133
RESUME	134
Exercices	135

Projet de consolidation 2 137

Module 3 : Programmation dynamique de pages Web 138

Chapitre 9 : Introduction aux bases de données 139

Activité	140
I. Terminologie des bases de données	141
1. Définition des concepts	141
2. Différences entre un fichier et une base de données	142
3. Type de base de données	142
II. Notion de Système de Gestion de Bases de Données (SGBD)	142
1. Fonctions d'un SGBD	143
2. Types de SGBD	144
RESUME	145
Exercices	146

Chapitre 10 : Le langage SQL 149

Activité	150
I. Définition de données	151
1. Définition de bases de données	151
2. Définition de tables	151
<i>a. Création de tables</i>	151
<i>b. Modification des tables</i>	154
<i>c. Suppression de tables</i>	155
II. Manipulations des données	155
1. Insertion d'un enregistrement	155
2. Modification d'un enregistrement	156
3. Suppression d'un enregistrement	156
III. Interrogation des données	157
1. Projection	158

2. Restriction _____	159	IV. Récupération du résultat d'un SELECT	187
3. Regroupement _____	160	V. Fermeture de la connexion _____	189
<i>a. Les fonctions de groupe</i> _____	160	VI. Accès aux éléments d'un formulaire HTML	190
<i>b. La clause HAVING</i> _____	161	1. Rappel sur les formulaires HTML_	190
4. Tri _____	162	2. Les éléments de formulaire _____	190
5. Jointure _____	162	3. Récupération des éléments du formulaire	avec PHP _____
RESUME 1 _____	64	191	
Exercices _____	165	RESUME _____	194
Chapitre 11 : Introduction à PHP _____	169	Exercices _____	194
Activité _____	170	<i>Projet de consolidation 3 _____</i>	197
I. Qu'est-ce que PHP ? _____	171	<i>Bibliographie _____</i>	199
1. Origines de PHP _____	171	<i>Index _____</i>	201
2. Intérêt de PHP _____	171		
3. Fonctionnement de PHP _____	172		
II. Syntaxe de PHP _____	173		
1. Les variables en PHP _____	173		
2. Opérations sur les variables _____	174		
3. Utilisation des structures de contrôle	175		
4. Utilisation des tableaux _____	176		
5. Utilisation des fonctions _____	177		
6. L'instruction echo _____	177		
III. Intégration de PHP à HTML _____	179		
RESUME _____	179		
Exercices _____	180		
Chapitre 12 : Accès aux données avec PHP _____	181		
Activité _____	182		
I. Connexion au serveur Web _____	183		
II. Choix de la base de données _____	184		
III. Envoi de requêtes SQL _____	185		

Listes des références

Liste des figures

Figure 1.1 : Schéma fonctionnel d'un ordinateur	6	Figure 3.10 : Réseau en anneau	40
Figure 1.2 : Unité centrale d'un ordinateur	7	Figure 3.11: Type de réseau en fonction de l'étendue géographique	42
Figure 1.3 : Carte mère standard	7	Figure 3.12: Type de réseau en fonction de l'étendue géographique	44
Figure 1.4 : Processeur	8	Figure 4.1 : Pile de protocole TCP/IP et protocoles utilisés	51
Figure 1.5 : Barrette mémoire	9	Figure 4.2 : Encapsulation des données dans TCP/IP54	
Figure 1.6 : Clavier	9	Figure 5.1 : Matériel nécessaire au sertissage des câbles à paires torsadées	66
Figure 1.7 : Souris	9	Figure 5.2 : Etapes de dénudage d'un câble à paires torsadées	67
Figure 1.8 : Imprimantes (jet d'encre à gauche et Laser à droite)	10	Figure 5.3 : Fils de câble regroupés par paires	67
Figure 1.9 : Ecrans (plat à gauche et CRT à droite)	10	Figure 5.4 : Agencement de fils pour câble droit	68
Figure 1.10 : Organes de stockage	11	Figure 5.5 : Agencement de fils pour câble croisé	69
Figure 2.1 : Code HTML minimum	26	Figure 5.6 : Connectique d'une carte réseau	70
Figure 2.2 : Page web sans balisage	27	Figure 5.7 : Plan d'adressage du réseau	72
Figure 2.3 : Page web avec balisage	27	Figure 7.1 : Algorithme de l'alternative réduite	96
Figure 3.1 : câble coaxial	34	Figure 7.2 : Algorithme de l'alternative complète	96
Figure 3.2 : paire torsadée	34	Figure 7.3 : Algorithme de la structure While	97
Figure 3.3 : fibre optique	34	Figure 7.4 : Algorithme de la structure Do ... While	98
Figure 3.4: Ordinateur de bureau	35	Figure 7.5 : Algorithme de la structure For	99
Figure 3.5: Ordinateur portable	35	Figure 7.6: Exécution de la fonction «prompt»	104
Figure 3.6 : Serveur	35	Figure 7.7: Exécution de la fonction «prompt»	105
Figure 3.7 : réseau en bus	38	Figure 7.8: Exécution de la fonction «alert»	106
Figure 3.8 : réseau en étoile	39		
Figure 3.9: réseau en anneau	40		

Figure 7.9 : Eléments d'une page Web _____	109
Figure 8.1 : Page "corrige.html" _____	128
Figure 8.2 : Exemple en mode exécution _____	129
Figure 9.1 : Structure de la table «Matière» _____	141
Figure 11.1 : Fonctionnement de PHP _____	173
Figure 12.1: Document HTML après l'exécution du script de l'Exemple 12.9 _____	189

Liste des exemples

Exemple 4.1 : Test du réseau avec ping _____	58
Exemple 4.2 : Résultat de ping sur Linux : _____	59
Exemple 4.3: Résultat de ping sur Windows : _____	59
Exemple 7.1 : Déclarations sans affectation en JavaScript _____	92
Exemple 7.2 : Déclarations avec affectation en JavaScript _____	92
Exemple 7.3 : Utilisation des variables en JavaScript _____	93
Exemple 7.4 : Utilisation d'un bloc d'instructions en JavaScript _____	95
Exemple 7.5 : Alternative simple en JavaScript _____	96
Exemple 7.6 : Alternative complète en JavaScript _____	97
Exemple 7.7 : Structure While en JavaScript _____	98
Exemple 7.8 : Structure Do ... While en JavaScript _____	99
Exemple 7.9 : Structure For en JavaScript _____	100
Exemple 7.10 : Utilisation des tableaux JavaScript _____	101
Exemple 7.11 : Création de fonction et de procédure _____	102
Exemple 7.12 : Appel de fonction en JavaScript _____	103
Exemple 7.13 : Utilisation de <i>document.write</i> _____	103
Exemple 7.14 : Utilisation de la fonction prompt en JavaScript _____	104
Exemple 7.15 : Conversion de type d'une fonction prompt _____	105

Exemple 7.16 : Utilisation de la fonction confirm en JavaScript _____	105
Exemple 7.17 : Utilisation de la fonction alert en JavaScript _____	106
Exemple 7.18 : Ajout d'une image à une page Web _____	107
Exemple 7.19 : Ajout d'une zone de saisie à une page Web _____	108
Exemple 7.20 : Ajout d'un bouton à une page Web _____	108
Exemple 7.21 : Accès à des éléments HTML par "name" _____	109
Exemple 8.1 : Intégration de JavaScript par la balise <script> _____	127
Exemple 10.1 : Création d'une base de données _____	151
Exemple 10.2 : Suppression d'une base de données _____	151
Exemple 10.3 : Création de la table «Matiere» _____	152
Exemple 10.4 : Création de la table Eleve _____	153
Exemple 10.5 : Création de la table Note _____	153
Exemple 10.6 : Script SQL de modification de la table Eleve _____	154
Exemple 10.7: Suppression de la table Eleve _____	155
Exemple 10.8 : Insertion d'un enregistrement à la table «Eleve» _____	155
Exemple 10.9 : Modification du Quartier de l'élève de matricule «E0009» _____	156
Exemple 10.10 : Modification de toutes les informations de l'élève de matricule «E0010» _____	156
Exemple 10.11 : Suppression d'enregistrements _____	156
Exemple 10.12 : Matricules et noms de tous les élèves _____	158
Exemple 10.13 : Code des matières sur lesquelles on a composé _____	158
Exemple 10.14 : Code des matières sur lesquelles on a composé (sans doublons) _____	158
Exemple 10.15 : Noms des élèves qui habitent Mendong _____	159

Exemple 10.16 : Noms des élèves dont le nom commence par B _____	159
Exemple 10.17 : Noms et date de naissance des élèves nés entre 1992 et 1994 _____	159
Exemple 10.18 : Moyenne des notes obtenues par matière _____	160
Exemple 10.19 : Meilleure note par matière _____	161
Exemple 10.20 : Nombre d'élèves par quartier _____	161
Exemple 10.21 : Code des matières pour lesquelles au moins deux élèves ont composé _____	161
Exemple 10.22 : Noms et quartier des élèves triés par ordre croissant des quartiers et décroissant des noms _____	162
Exemple 10.23 : Noms, matière et notes des élèves	162
Exemple 11.1 : Utilisation de variables en PHP _____	173
Exemple 11.2 : Utilisation d'une structure alternative en PHP _____	175
Exemple 11.3 : Utilisation d'une boucle While en PHP	175
Exemple 11.4 : Utilisation d'une boucle Do ...While en PHP _____	175
Exemple 11.5 : Utilisation d'une boucle For en PHP	175
Exemple 11.6 : Déclaration de tableau en PHP _____	176
Exemple 11.7 : Accès aux éléments d'un tableau PHP	176
Exemple 11.8 : Déclaration et appel de fonctions en PHP _____	177
Exemple 11.9 : Utilisation de l'instruction "echo"	178
Exemple 12.1 : Connexion à un serveur Web _____	184
Exemple 12.2 : Test de connexion à un serveur Web	184
Exemple 12.3 : Test de connexion à une base de données	185
Exemple 12.4 : Requête PHP d'insertion d'enregistrement à la base _____	186
Exemple 12.5 : Requête PHP de suppression d'enregistrement à la base _____	186
Exemple 12.6 : Requête PHP de sélection d'enregistre-	

ment à la base _____	186
Exemple 12.7 : Accès à un recordset par un tableau	187
Exemple 12.8 : Accès à un recordset par un tableau associatif _____	188
Exemple 12.9 : Accès à un recordset par un objet _____	188
Exemple 12.10 : Fermeture de la connexion au serveur Web _____	189
Exemple 12.11 : Formulaire de connexion à un site	191

Liste des tableaux

Tableau 1.1 : Quelques extensions de fichier _____	13
Tableau 1.2 : Systèmes d'exploitation par type _____	16
Tableau 2.1 : Liste des balises usuelles _____	24
Tableau 5.1 : Plan d'adressage du réseau _____	71
Tableau 7.1 : Opération arithmétiques en JavaScript	94
Tableau 7.2 : Opération arithmétiques en JavaScript	94
Tableau 7.3 : Objet et gestionnaires d'évènements en JavaScript _____	107
Tableau 10.1 Opérations en PHP _____	174
Tableau 10.1 : Types de données en SQL _____	152
Tableau 10.2 : Structures de contrôle en PHP _____	175
Tableau 10.2 : Types de modification de table _____	154
Tableau 10.3 : Fonctions de groupe dans SQL _____	160
Tableau 10.3 : Variables prédéfinies en PHP _____	176
Tableau 12.1 : Syntaxe des éléments de formulaire HTML _____	190

Module 0

RAPPELS

L'informatique est une discipline transversale, en ce sens qu'elle est applicable à presque tous les domaines d'activité d'aujourd'hui. Pour l'opérabilité de cette discipline, un matériel principal est adapté à savoir l'ordinateur. Si son utilisation est préférée à l'homme c'est parce qu'elle est non seulement plus fiable mais plus rapide, plusieurs équipements lui sont associés pour remplir pleinement son rôle.

L'ordinateur, plaque tournante des applications, permet la réalisation des sites web. Pour mieux comprendre le langage JavaScript, il est judicieux de maîtriser le HTML.

Ce module est donc un cours introductif qui permet de mieux appréhender les modules suivants.

LOGICIELS D'APPLICATION ET UTILITAIRES

- Logiciels de traitement de texte
- Tableurs
- Navigateurs
- Antivirus

LOGICIELS SYSTÈMES

- Systèmes d'exploitation
- BIOS
- Pilotes

MATERIEL

- Ordinateurs de bureau
- Ordinateurs portables
- Imprimantes
- Appareils photos numériques

SOFTWARE

HARDWARE

1

ENVIRONNEMENT INFORMATIQUE

SOMMAIRE

- ▶ Présentation générale
- ▶ Fonctionnement d'un ordinateur
- ▶ Notions de base sur les logiciels

MOTS-CLES

- Hardware
- Software
- Programmes
- Fichier
- Système d'exploitation



Ordinateur de bureau



Ordinateur portable



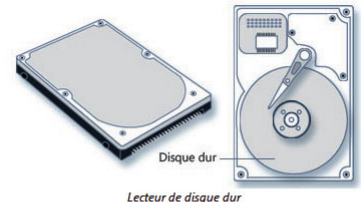
Tablette PC



Ipad



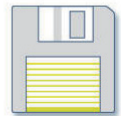
Imprimante



Disque dur
Lecteur de disque dur



CD



Disquette

OBJECTIFS

- Présenter les notions d'ordinateur et d'informatique
- Décrire le fonctionnement d'un ordinateur
- Différencier les familles de logiciels

L'histoire de l'ordinateur nous provient du fait que l'homme, par nature paresseux, a toujours cherché à améliorer sa façon de calculer, afin de limiter ses erreurs et économiser son temps.

L'être humain dans ses échanges avec son milieu extérieur, est capable d'acquérir et de conserver des informations, d'effectuer des traitements et de restituer les informations stockées.

- L'acquisition des informations se fait grâce aux organes d'entrée
- La restitution des informations grâce aux organes de sortie
- La conservation des informations grâce aux organes de stockage
- Le traitement des informations grâce à l'unité de calcul

Donner les organes de l'être humain qui permettent :

1. L'acquisition des informations
2. Le traitement des informations
3. La conservation des informations de manière durable
4. La conservation des informations de manière temporaire
5. La restitution des informations

Ces organes ont besoin des ordres, venant de l'unité de calcul, sans lesquels ils ne sauraient fonctionner.

6. Comment appelle t'on cet ensemble d'ordres ?

Parmi ces ordres, il y en a qui permettent de remplir les fonctions vitales de l'être humain (faire battre le coeur). Tandis que d'autres utilisent les fonctions vitales pour effectuer une série d'actions selon le besoin (marcher).

6. Citer des fonctions vitales de l'être humain
7. Citer des actions combinant les fonctions vitales pour satisfaire un besoin
8. Les organes de l'être humain peuvent t-ils fonctionner sans ses fonctions vitales ?

I. Présentation Générale

1. Ordinateur et Informatique

Le terme anglais «computer» signifiait au départ : calculateur numérique électronique. En effet, les premières machines étaient utilisées pour effectuer des suites d'opérations. L'ordinateur se définit maintenant comme une machine de traitement de l'information. Il capable d'acquérir et de conserver des informations, d'effectuer des traitements et de restituer les informations stockées. Un ordinateur peut traiter divers types d'informations (valeurs numériques, textes, images) mais de manière interne, toutes ces informations sont converties sous forme numérique (série de bits). On peut donc utiliser l'ordinateur pour traiter des textes, des dessins ou des images, etc.

L'Informatique (en anglais computer sciences) est un terme issu de la contraction des mots « information » et « automatique ». C'est la science du traitement de l'information à l'aide d'ordinateurs. Ce traitement consiste à effectuer des opérations sur des données (informations représentées sous forme codée et compréhensible par l'ordinateur).

2. Utilisation des ordinateurs

L'utilisation des ordinateurs sous-entend l'exécution d'un programme; c'est pourquoi toute application (réservation de billets de voyage, gestion des notes des élèves ou simplement un traitement de texte) nécessite du matériel, en l'occurrence un ordinateur, et du logiciel qui est exécuté par cet ordinateur. Le logiciel se décompose en deux familles : *les programmes systèmes et les programmes d'applications*.

Les utilisateurs se servent des programmes pour effectuer différentes tâches. Les fonctions de base d'un ordinateur sont réalisées au moyen de programmes système, qui sont intégrés après la

Informatique :

Le mot informatique a été inventé en 1962 par Philippe DREYFUS. Sa définition première est proche de son synonyme anglais (computer sciences) qui le restreint aux sciences de l'ordinateur, donc du calcul. L'informatique aujourd'hui va bien au-delà du calcul numérique.



REMARQUE



DEFINITION

Programme :

Ensemble d'instructions exécutables par un ordinateur dans le but de réaliser une tâche précise.

construction de la machine. Les programmes d'application sont développés généralement par des entreprises du logiciel ou par les utilisateurs eux-mêmes. Ce sont soit des programmes généraux tels que des logiciels de traitement de texte ou des logiciels spécifiques aux besoins d'un utilisateur.

Le système d'exploitation, le plus important des programmes systèmes, s'occupe de gérer les différentes ressources de la machine, il est généralement spécifique à une famille d'ordinateurs compatibles. Il gère les travaux des utilisateurs, c'est-à-dire qu'il enchaîne les différentes étapes nécessaires à l'exécution d'un programme.

II. Fonctionnement d'un ordinateur

Un ordinateur est composée de deux parties : une partie matérielle appelée Hardware et une partie logicielle appelée Software. Ces deux parties sont indispensables voir même indissociable pour le bon fonctionnement de l'ordinateur.

1. Le hardware

Ce sont tous les composants matériels, qu'ils soient extérieurs ou contenus à l'intérieur de l'unité centrale.

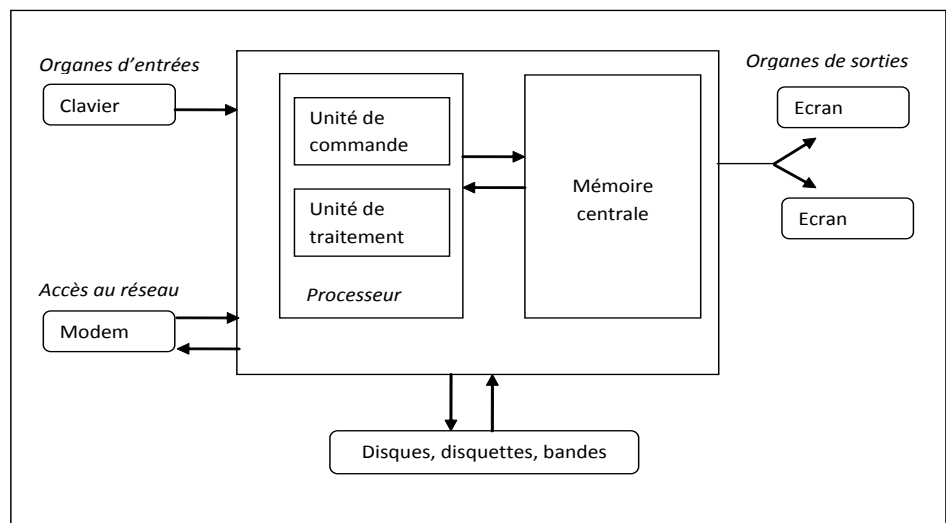


Figure 1.1 : Schéma fonctionnel d'un ordinateur

a. L'unité centrale

C'est un boîtier contenant les principaux éléments suivants :

i. La carte mère

C'est la partie électronique principale. Elle réunit tous les composants matériels internes de l'unité centrale. Ceux-ci sont fixés :

- Directement : processeur, mémoire centrale, carte d'extension ;
- Ou par branchement d'une nappe : disque dur, lecteur de disquette, lecteur de CD-ROM.



Figure 1.2 : Unité centrale d'un ordinateur

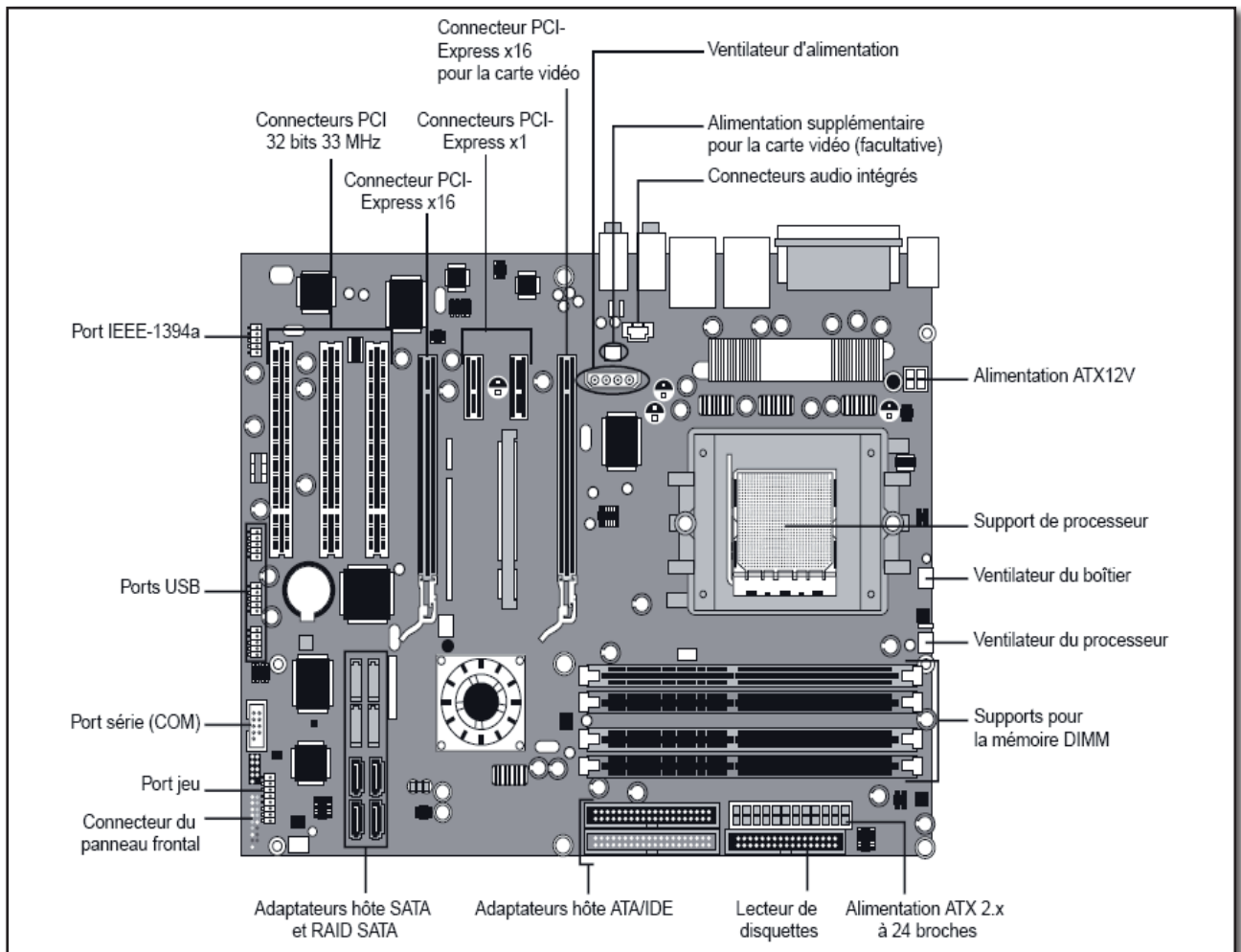


Figure 1.3 : Carte mère standard

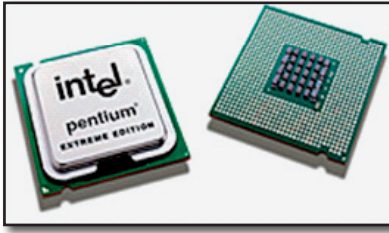
ii. Le processeur

Figure 1.4 : Processeur

Le mot « processeur » vient du verbe anglais « process » qui veut dire « exécuter ». Encore appelé Unité Centrale de Traitement (en anglais CPU : Central Processing Unit), le processeur a pour rôle l'interprétation et l'exécution des instructions des programmes en mémoire. Le processeur est composé de deux principaux éléments :

- D'une unité de traitement : contenant l'Unité Arithmétique et Logique (UAL) ou unité de calcul ;
- D'une unité de commande ou unité d'instruction ;

La puissance d'un processeur est définie en fonction de sa fréquence. La fréquence d'un processeur est le nombre d'opérations simples qu'il est capable d'effectuer en une seconde. Elle est mesurée en Hertz (Hz).

iii. La mémoire centrale

De manière générale, les mémoires sont des dispositifs de stockage de l'information. Elles sont caractérisées par :

- Une capacité : qui représente le volume global d'informations que la mémoire peut stocker ;
- Un temps d'accès : correspondant à l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée.

La mémoire centrale est la mémoire de travail de l'ordinateur. Elle a pour rôle de stocker les instructions et données des programmes en cours d'exécution.

La mémoire centrale est appelé RAM (Random Access Memory) par opposition à la ROM (Read Only Memory) qui ne peut être réécrite. Elle est également appelée mémoire vive car elle perd les informations qu'elle contient en absence d'alimentation électrique : on dit qu'elle est volatile.

La mémoire centrale se présente sous forme de barrettes à enficher sur la carte mère. Sa capacité est mesurée en octet.

b. Les organes d'entrée

i. Le clavier

Le clavier est un dispositif qui permet de saisir des caractères et d'envoyer des commandes vers l'unité centrale.

Les claviers peuvent être connectés sur le port clavier (généralement violet) ou sur un port USB. Certains claviers utilisent des liaisons WiFi (sans fil). Les claviers des ordinateurs portables sont directement intégrés sur le volet inférieur de ceux-ci.

ii. La souris

La souris est un dispositif de pointage. Elle sert à pointer et à sélectionner des éléments sur l'écran, à déplacer ou à manipuler des objets.

Les souris peuvent être connectées sur le port souris (généralement vert) ou sur un port USB. Certaines souris utilisent des liaisons WiFi (sans fil). Les souris des ordinateurs portables sont directement intégrées sur le volet inférieur de ceux-ci. Elles sont composées d'un rectangle tactile représentant l'écran et de deux boutons.

iii. Les autres périphériques d'entrée

Plusieurs autres périphériques permettent d'envoyer des informations vers l'unité centrale. Il s'agit :

- **Du scanner ou numériseur** : il permet de transformer une image réelle en image numérique qui pourra directement être traitées par l'ordinateur ;
- **De la Webcam** : elle permet d'envoyer des vidéos vers l'unité centrale ;
- **Du microphone** : il permet d'envoyer des sons vers l'unité centrale.

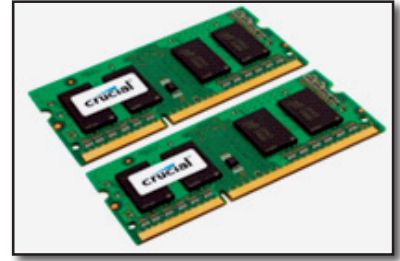


Figure 1.5 : Barrette mémoire



Figure 1.6 : Clavier

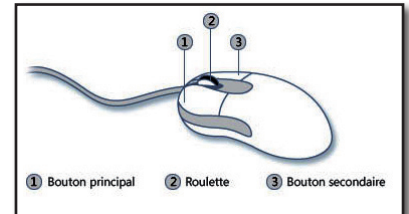


Figure 1.7 : Souris

c. Les organes de sortie*i. L'imprimante*

L'imprimante est une machine électromécanique qui permet de produire une sortie gravée des données de l'ordinateur sur un support généralement papier. On en rencontre trois types :

- Les imprimantes matricielles ;
- Les imprimantes à jet d'encre ;
- Les imprimantes LASER.

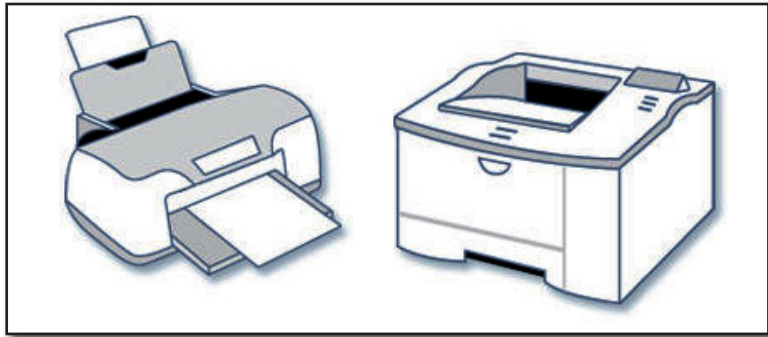


Figure 1.8 : Imprimantes (jet d'encre à gauche et Laser à droite)

ii. L'écran

L'écran ou moniteur est le périphérique de sortie le plus répandu. Il est utilisé pour afficher du texte ou des graphismes. On distingue deux types d'écran :

- Les écrans à tubes cathodiques ou CRT (Cathode Ray Tube)
- Les écrans matriciels ou écrans plats.

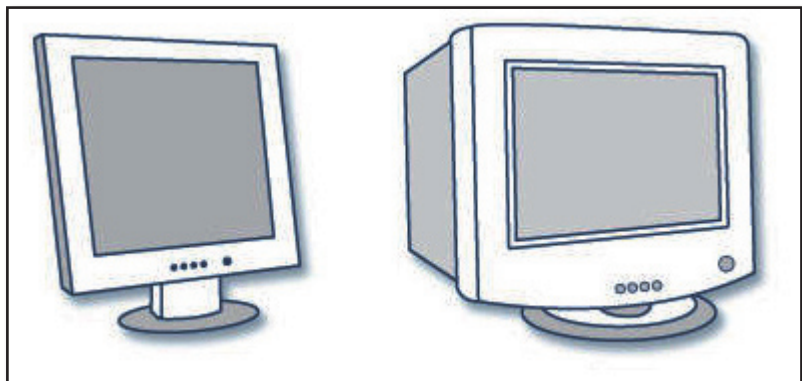


Figure 1.9 : Ecrans (plat à gauche et CRT à droite)

iii. Les autres organes de sortie

Citons les baffles ou hauts-parleurs et le vidéoprojecteur.

d. Les organes de stockage

Les organes de stockage ou de sauvegarde sont encore appelés mémoires de masse ou mémoire auxiliaires car ils sauvegardent les informations de manière permanente.

i. Le disque dur

Le disque dur est un ensemble de disques réunis autour d'un plateau circulaire fait de métal, de céramique ou de verre et recouvert d'une mince couche d'oxyde de fer magnétisable.

La capacité d'un disque dur se mesure en octet. Sur le marché, les disques durs ont des capacités qui varient entre 20 et 80 Go. Toutefois, il en existe qui ont de bien plus grandes capacités.

ii. Les organes de sauvegarde amovibles

Ce sont des périphériques de stockage qui sont connectés à l'ordinateur à travers un lecteur spécifique. Ils sont dits amovibles car facilement transportables d'un ordinateur à un autre. Ce sont :

- La disquette ;
- Le CD (Compact Disc) ;
- Le DVD (Digital Versatile Disc) ;
- Les clés USB (Universal Serial Bus) ;
- Les cartes mémoires.

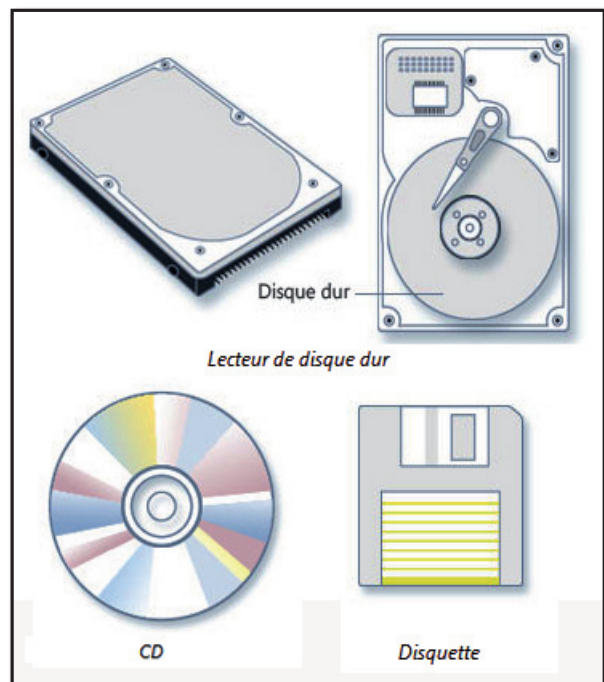


Figure 1.10 : Organes de stockage

e. Les organes d'entrée et sortie

Ce sont des périphériques qui permettent à la fois de lire et d'écrire des informations sur un support. Ce sont principalement :

- Les lecteurs (lecteur de disque, de disquette) ;
- Les graveurs (CD, DVD, ...) ;
- Le modem (modulateur/démodulateur).

Avec l'arrivée des jeux multimédia, on a développé des manettes qui renvoient des vibrations.

f. Les organes de communication

i. *Le bus*

Un bus est un ensemble de fils électriques (cuivre) qui transporte les informations binaires. Chaque fil transporte un bit (0 ou 1).

On distingue :

- Le bus de données ;
- Le bus de commandes ;
- Le bus d'adresses.

ii. *Le chipset*

On appelle chipset (en français jeu de composants) l'élément chargé d'aiguiller les informations entre les différents bus de l'ordinateur afin de permettre à tous les éléments constitutifs de l'ordinateur de communiquer entre eux.

g. Les cartes d'extension

Les premiers ordinateurs traitaient les informations de type texte. Avec l'arrivée du multimédia (images, sons, vidéo), on a dû étendre les fonctionnalités des ordinateurs pour qu'ils puissent gérer les nouveaux périphériques. On distingue :

- La carte graphique ;
- La carte son ;
- La carte réseau.

2. Le software

Ce sont les programmes qui servent à utiliser l'ordinateur.

- Certains sont indispensables à son fonctionnement (comme le système d'exploitation),
- D'autres sont ajoutés pour étendre ses possibilités d'utilisation (Internet Explorer et Outlook Express, un traitement de texte, un tableur, un jeu, etc).

a. L'exécution d'un programme

Au lancement d'un programme, Le processeur va chercher les données du programme sur le disque dur, il les place dans la mémoire vive (RAM), afin de pouvoir les avoir à portée de main, il s'en sert pour effectuer des calculs, et transmet les résultats à la carte graphique, qui les affiche à l'écran.

b. Les programmes et les fichiers

Les programmes manipulent de fichiers. Généralement, les fichiers créés par un programme ne peut être manipulés que par celui-ci. Toutefois, certains fichiers (texte, vidéo, son, ...) peuvent être manipulés par divers programmes. Quelque soit les cas, un fichier d'un extension. Le tableau ci-après présente quelques extensions courantes :

Tableau 1.1 : Quelques extensions de fichier

Nom du fichier	Commentaire
Ms_Word.exe	Fichier exécutable qui permet de lancer un programme
vidéo1.avi, vidéo2.mp4	Fichiers vidéo
son1.mp3, son2.wma	Fichiers son
Mapage.html	Page Web
Historique.txt	Document texte
Document.docx	Document Word
monscript.js	Fichier de script JavaScript
script.sql	Fichier de script SQL



DEFINITION

Fichier :

Ensemble d'informations de même nature regroupées en unité indépendante de traitement.



DEFINITION

Extension de fichier

Ensemble de trois ou quatre caractères qui définissent la nature d'un fichier ou le programme avec lequel il peut être ouvert

**Dossier :**

Division logique de l'espace mémoire qui permet de rapidement retrouver des informations.

Un ordinateur contient un très grand nombre de fichiers. Pour les retrouver rapidement, on les stocke dans des *dossiers ou répertoires*. Un dossier peut être assimilé à un album photo et les photos qui y sont collés sont des fichiers.

Un dossier peut contenir un ou plusieurs dossiers. Ceux-ci sont alors appelés *sous-dossiers*. Le dossier dans lequel se trouve un sous-dossier est son *dossier parent*.

3. Notions de base sur les logiciels

Un logiciel est un ensemble de programmes permettant la réalisation d'une fonction précise. (Exemple : un logiciel de traitement de texte permet de saisir du texte et de le mettre en forme grâce à des programmes pour le gras, l'italique, le souligné, ...).

On distingue deux grands types de logiciels :

- *Les logiciels systèmes* : qui permettent de contrôler la machine ;
- *Les logiciels d'application* : qui permettent de réaliser des tâches précises des utilisateurs.

a. Les logiciels systèmes

Les principaux programmes système sont les systèmes d'exploitation et le BIOS

Le BIOS (Basic Input Output System) permet de gérer les accès entre le processeur et les périphériques. Il contient un ensemble de fonctions (routines) qui sont lancées au démarrage de l'ordinateur et réalise un test appelé POST (Power On Self Test). Le POST permet de vérifier que les composants matériels nécessaires au fonctionnement de l'ordinateur sont présents et fonctionnels

Une partie du BIOS est contenue dans la mémoire morte ROM

qui est alimenté par une pile CMOS (Complementary Metal-Oxide Semiconductor)

Les systèmes d'exploitation (SE) sont des logiciels chargés d'assurer le bon fonctionnement de l'ordinateur et de présenter à l'utilisateur une machine virtuelle qui masque la complexité de la machine physique.

i. Fonctions d'un système d'exploitation

Un système d'exploitation a pour principales fonctions :

- **La Gestion du processeur** : le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes grâce à un programme d'ordonnancement.
- **La Gestion de la mémoire vive** : le système d'exploitation est chargé de gérer l'espace mémoire alloué à chaque application et, le cas échéant, à chaque utilisateur.
- **La Gestion des entrées/sorties** : le système d'exploitation permet de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes (appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie).
- **La Gestion de l'exécution des applications** : le système d'exploitation est chargé de la bonne exécution des applications en leur affectant les ressources (fichiers, programmes, matériels nécessaires à leur bon fonctionnement).
- **La Gestion des fichiers** : le système d'exploitation gère la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.

ii. Types de systèmes d'exploitation

On distingue les systèmes d'exploitation suivant :

- **Le nombre de tâches** qu'on peut exécuter à la fois : on parle de systèmes mono-tâches et de systèmes multi-tâches ;
- **Le nombre de postes** qu'on peut connecter : on parle de systèmes monoposte et de systèmes multipostes ou réseau

Mémoire virtuelle :

C'est une zone mémoire créée sur le disque dur qui permet de faire fonctionner les applications nécessitant plus de mémoire vive qu'il n'y en a de disponible sur le système.



REMARQUE

- *Le nombre d'utilisateurs* qui peuvent utiliser le système : on parle de systèmes mono-utilisateurs ou multi-utilisateurs. Dans un système est multi-utilisateurs, chaque utilisateur dispose d'un compte (ensemble de fichiers, dossiers et préférences auxquels il peut accéder).
- *La largeur du bus d'adresse* : on parle de système 16, 32 ou 64 bits

Le tableau suivant regroupe les principaux systèmes d'exploitation classés par type.

Tableau 1.2 : Systèmes d'exploitation par type

Système	Longueur du bus	Mono utilisateur	Multi utilisateur	Mono tâche	Multi tâche
DOS	16 bits	X		X	
Windows 3.1	16/32 bits	X			X
Windows 95/98 /Me	32 bits	X			X
Windows NT/2000	32 bits		X		X
WindowsXP	32/64 bits		X		X
Windows Vista	32 bits		X		X
Windows 7	32/64 bits		X		X
Windows Server	32 bits		X		X
Unix	32 bits		X		X
Linux	32/64 bits		X		X
Mac OS	32 bits		X		X

b. Les logiciels d'application

Il existe une pléthore de logiciels d'application ou programmes. Ils peuvent être regroupés en famille de la manière suivante :

- *Les logiciels de traitement de texte* : Microsoft Office Word, WordPerfect, OpenWriter ;
- *Les tableurs* : Microsoft Office Excel, Lotus 1-2-3, OpenCalc ;

- *Les Systèmes de Gestion de Bases de Données (SGDB)* : Microsoft Office Access, Oracle, SQLServer, MySQL ;
- *Les logiciels de présentation ou de PréAO* : Microsoft Office PowerPoint ;
- *Les navigateurs* : Internet Explorer, Netscape Navigator, Mozilla Firefox, Safari, Opéra ;
- *Les logiciels de publication ou de PAO* : Microsoft Office Publisher, Xpress, PageMaker, Indesign ;
- *Les logiciels de retouche photo* : Adobe Photoshop, Photo-Filtre, Gimp ;
- *Les logiciels de développement* : Visual Basic, Delphi, Windev ;
- *Les logiciels de développement Web* : Microsoft Office Front-Page, Macromedia Dreamweaver, Mambo, Joomla! ;
- *Les logiciels multimédia* : RealPlayer, VLC, Windows Media, VirtualDj, Magix ;
- *Les logiciels de design graphique (infographie)* : InDesign, Illustrator, Corel Draw ;
- *Les suites logicielles* : ce sont des ensembles de logiciels réunis en package. On peut citer :
 - La suite bureautique Microsoft Office (avec les logiciels Word, Excel, Access, Outlook, ...). Il existe d'autres suites bureautiques (OpenOffice, Sun StarOffice) ;
 - La suite graphique Adobe Creative Suite (Flash, InDesign, Photoshop, Dreamweaver, Illustrator, ...)
 - La suite de développement Visual Studio (Visual Basic, Visual C#, Visual C++, ...)

c. Les utilitaires

Ce sont des logiciels qui assistent ou qui aident les usagers ou les développeurs dans leurs tâches quotidiennes sur les ordinateurs :

Bureautique :

Ensemble d'outils logiciels qui permettent d'automatiser des tâches de bureau. Il comprend généralement des logiciels de traitement de texte, tableur, messagerie et PAO.



REMARQUE



Virus informatique :

Un virus informatique ou « Code Auto-propageable » est un petit programme qui va s'insérer dans un autre et exécuter une série d'instructions à l'insu de son programme hôte.

i. Les antivirus

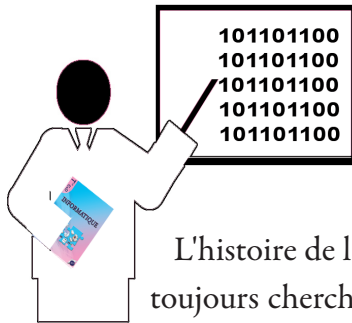
Les antivirus sont des logiciels capables de détecter la présence d'un virus et de nettoyer l'ordinateur afin de prévenir tout risque. Le fichier infecté est alors réparé, mis en quarantaine (dans une zone à partir de laquelle il est inaccessible) ou supprimé.

Les antivirus les plus courants sont : Mc Affée, Avira, Avast, Norton, Panda

ii. Les autres utilitaires

Citons entre autres :

- Les logiciels de compression (Winzip, Winrar, ...)
- Les logiciels de gravure (Easy CD Creator, Nero Burning Rom, ...)
- Les outils d'accessibilité (Loupe, clavier visuel, explorer, ...)



RESUME

L'histoire de l'ordinateur nous provient du fait que l'homme, par nature paresseux, a toujours cherché à améliorer sa façon de calculer, afin de limiter ses erreurs et économiser de son temps. L'idée était donc de concevoir une machine permettant d'effectuer les tâches de l'homme plus rapidement. A l'instar de l'ordinateur qui, pour réaliser les opérations a besoin simultanément de la partie matérielle (hardware) et de la partie logiciel (Software).

Le hardware est composé de l'unité centrale autour duquel sont connectés différents organes : clavier, souris, écran, imprimante, etc.

Le software est composé des logiciels systèmes (BIOS et Système d'exploitation), des logiciels d'applications (traitement de texte, tableur, etc.) et des utilitaires (antivirus, etc.).

EXERCICE
THEORIQUE**Exercice 1 : Répondre aux questions suivantes.**

1. Définir les termes : ordinateur, programmes, fichier, système d'exploitation
2. Quels sont les parties d'un ordinateur
3. Qu'est ce qu'un logiciel d'application ? Citez en 3 exemples
4. Qu'est ce qu'un logiciel système ? Citez en 3 exemples
5. Donner le logiciel de base qui permet de faire fonctionner l'ordinateur.

EXERCICE
THEORIQUE**Exercices 2 : A l'aide des mots : identificateur, dossier, information, ordinateur, fichier, extension, nature, fichiers, complétez la phrase suivante :**

Un _____ est un ensemble _____ de même _____ regroupées en une unité indépendante de traitement.

Un _____ ou répertoire est une division logique de l'espace mémoire de l'_____ permettant de conserver, et de retrouver nos _____.

Le nom d'un fichier est constitué de son _____ proprement dit, et d'une _____.

EXERCICE
THEORIQUE**Exercices 3 : A l'aide des mots : traitement de texte, programmes systèmes, utilisateur, programmes, système d'exploitation, complétez la phrase suivante :**

Les utilisateurs se servent des _____ pour effectuer différentes tâches sur un ordinateur. Les fonctions de base d'un ordinateur sont réalisées au moyen de _____, développé tel que le _____.

Les programmes d'applications sont des programmes développés généralement des entreprises du logiciel ou par les utilisateurs eux-mêmes. Ce sont soit des programmes généraux tels que des logiciels de _____ soit des logiciels spécifiques aux besoins d'un _____.

EXERCICE
THEORIQUE**Exercices 4 : Répondre par vrai ou faux**

1. Un ordinateur est composé de deux parties le hardware et le software ;
2. Le mot informatique est issu de la contraction des mots *information* et *automate* ;
3. Les mots computer et informatique ont la même signification ;
4. Le hardware suffit pour faire fonctionner un ordinateur ;
5. En termes de comparaison le cerveau humain est plus rapide que le cerveau processeur.



Exercices 5 : Identification du matériel informatique

Observer les figures 1 et 2. Donnez un nom à ces figures et donnez le nom et le rôle des éléments numérotés.

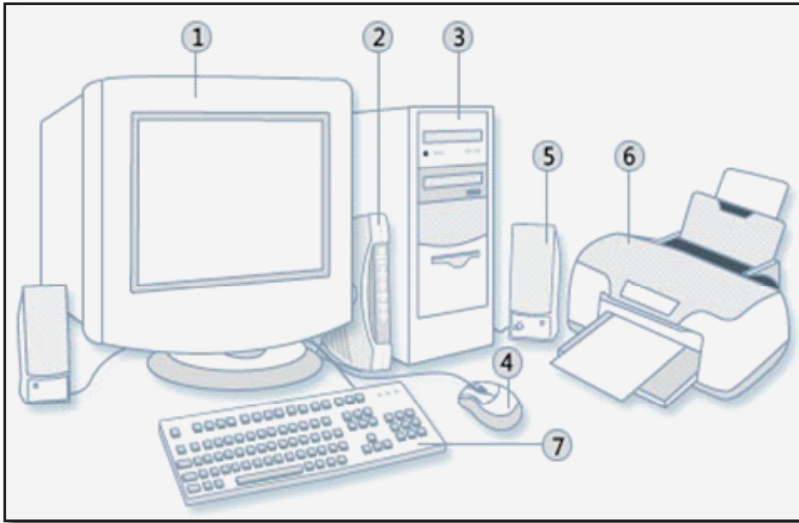


Figure 1

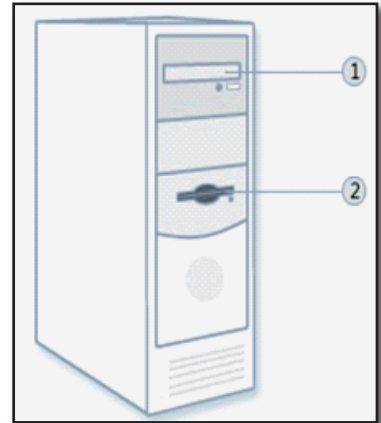


Figure 2

Module 0 : Rappels

2

LE LANGAGE HTML

SOMMAIRE

- ▶ Notion de site web
- ▶ Contenu d'une page web
- ▶ Exemple de page web

MOTS-CLES

- Page web
- Navigateur
- Balise

PRE-REQUIS

- Environnement informatique



OBJECTIFS

- Définir : site web, page web, navigateur, balise.
- Utiliser les balises usuelles du HTML

La construction d'une maison est un chef d'œuvre qui se planifie, depuis l'idée jusqu'à sa réalisation. Cela passe par plusieurs étapes à savoir :

- L'organisation de sa maison : nombre de chambres, de salons, de toilettes, de cuisine, piscine, véranda, etc.
 - La conception de la maquette de la maison
 - Puis la construction de la maison proprement dite
 - Et l'ameublement et la décoration de la maison
1. En quoi consiste l'organisation de la maison ?
 2. En quoi consiste la conception de la maquette de la maison ?
 3. En quoi consiste la construction de la maison ?
 4. En quoi consiste l'embellissement de la maison ?

La maison est donc et déjà habitable.

5. Quel élément permet de passer d'un compartiment de la maison à un autre ?

I. Notion de site Web

Un site Web est un ensemble de pages Web reliées entre elles par des liens hypertextes. On distingue deux types de sites Web :

- *Les sites statiques, ou de présentation ou site vitrine* : qui permettent d'afficher des contenus statiques (qui ne changeront pas d'un internaute à un autre);
- *Les sites dynamiques* : qui permettent d'afficher un contenu en fonction des demandes des internautes.

Les pages Web sont des documents informatiques créées dans des éditeurs de texte à l'aide des langages HTML et CSS, et visualisés par les internautes grâce à un navigateur Web.

HTML (Hypertext Markup Language) est un langage de description de page Web. C'est le langage du Web.

CSS (Cascade Style Sheets) est un langage qui permet de réaliser la mise en page et le graphisme de la page Web. Si HTML s'occupe de créer le fond ou le dessin au crayon, CSS s'occupe de la forme ou du coloriage du dessin.

Les documents HTML sont des documents texte. Pour pouvoir être affichés, on a besoin d'un navigateur Web ou browser qui analyse le code HTML d'une page et produit un résultat visuel (Mozilla Firefox ; Netscape Navigator, Internet Explorer, Safari).

II. Contenu d'une page Web

1. Notion de balise

Une page HTML est rédigée en HTML. HTML repose sur l'utilisation des balises. Une balise est un élément de texte délimité par des chevrons ouvrant et fermant (< et >), qui qualifie des portions de texte, des paragraphes, des tableaux, etc. Sauf exception, les balises fonctionnent par paires, et à chaque balise ouvrante correspond sa balise fermante, débutant par les caractères </>. Ces paires



Script

Un script est un petit programme qui va s'insérer dans un autre programme

délimitent la portion de texte à laquelle elles s'appliquent. (Exemple : `<html>` et `</html>`).

Parfois, une balise est refermée immédiatement. Une telle « balise vide » dite unique, ne délimite alors qu'un point, et non une zone du document. (exemple : `
` ``)

Un document HTML minimum comporte la structure suivante :

```
<html>
  <head>
    .....
  </head>

  <body>
    .....
  </body>

</html>
```

2. Notion d'attribut de balise

Que la balise soit de type « paire » ou « unique », elle peut prendre en plus ce qu'on appelle des attributs. Le rôle des attributs est de compléter une balise pour donner des informations supplémentaires. (exemple : `<image nom="mon_image.jpg">`).

3. Balises HTML usuelles

Les balises HTML les plus couramment utilisés sont résumées dans le tableau ci-après.

Tableau 2.1 : Liste des balises usuelles

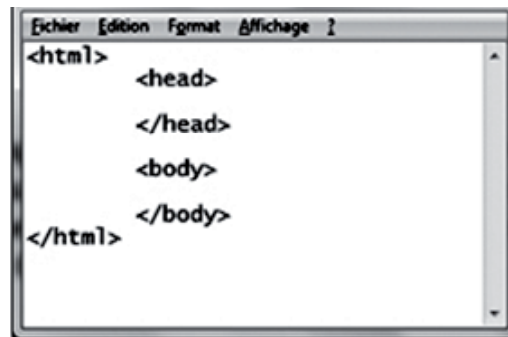
Balise	Attributs utiles	Commentaires et utilisations
<code><title></code>		Texte de la barre de titre dans la fenêtre du navigateur <code><title>Album officiel des élèves du LBM</title></code>

Balise	Attributs utiles	Commentaires et utilisations
<body>	Bgcolor, text, link, vlink, alink	<body bgcolor="green" text="black">
<link>	Rel,type,href	Liaison à une feuille de style <link rel="stylesheet" type="text/css" href="eleveslbn.css" />
<p>	Align (left, right, center, justify)	Balise désignant un paragraphe <p align="center">Un paragraphe de texte</p>
<h1>, <h2>.. <h6>	Align (left, right, center, justify)	HTML prévoit six niveaux de titres, hiérarchiquement placés sous le titre principal (<h1>). <h1>Un titre principal</h1>
 		Saut de ligne
<hr>	Size, width, Align (left, right, center)	Ligne horizontale (<hr size="3" width="50"/>)
<pre>		Affiche le texte tel qu'il est tapé
	Type (A,a,1,i,I), start	Liste ordonnée
		Liste non ordonnée
		Élément de liste
		Mise en gras
<u>		Souligné
		Renforcé
<sub>		Mise en indice
<sup>		Mise en exposant
<blockquote>		Indentation de texte
<var>		Mise en italique
	Face, size, color	
<a>	Href, title	Insertion d'un lien hypertexte (accueil)
	Src,alt, width, height, Align (left, right)	Insertion d'une image ()
<table>	Width, border, Align (left, right)	Insertion d'un tableau (<table width="60%" border="1">)
<tr>		Ligne de tableau
<th>	Width, height, colspan,	Titre de tableau
<td>	rowspan, align(left, right, middle)	Cellule de tableau

III. Exemple de page web

Nous allons créer une page de présentation du Lycée Bilingue du Cameroun. Pour le faire, on utilise un éditeur de texte ("bloc-notes", Wordpad, Notepad).

Après avoir tapé le code de notre document minimum, on enregistre le fichier (menu Fichier ... Enregistrer). On tape le nom « accueil.html ». On obtient la figure 6 ci-contre :



```

Fichier  Edition  Format  Affichage  ?
<html>
    <head>
</head>
    <body>
</body>
</html>
  
```

Figure 2.1 : Code HTML minimum

Après avoir enregistré le document, on double-clique sur notre document html et la page web s'affiche dans notre navigateur.

On notera l'indentation qui est faite après la balise ouvrante <html>. L'écriture du code avec indentation permet de bien structurer les différents blocs, et ainsi de repérer aisément les éventuelles erreurs, notamment pour ce qui concerne les balises fermantes.

Entre les éléments <body> et </body>, on insère le texte de l'encadré ci-après :

Bienvenue sur le site des élèves du Lycée Bilingue du Cameroun
 Présentation générale
 Le Lycée Bilingue du Cameroun est un établissement scolaire d'enseignement général secondaire bilingue. Il est composé :

- D'une section francophone : comprenant des classes de sixième ;
- D'une section anglophone comprenant des classes de Form I et de Form II.

On obtient la page web suivante dans notre navigateur :

Bienvenue sur le site des élèves du Lycée Bilingue du Cameroun
Présentation générale Le Lycée Bilingue du Cameroun est un établissement scolaire d'enseignement général secondaire bilingue. Il est composé : D'une section francophone : comprenant des classes de sixième ; D'une section anglophone comprenant des classes de Form I et de Form II.

Figure 2.2 : Page web sans balisage

On tape maintenant le texte de l'encadré ci-après entre les balises <body> et </html>:

```
<p><h1>Bienvenue sur le site des élèves du Lycée Bilingue du Cameroun</h1></p>
<p><h3>Présentation générale</h3></p>
<p>Le Lycée Bilingue du Cameroun est un établissement scolaire d'enseignement
général secondaire bilingue. Il est composé :
  <ul>
    <li>D'une section francophone : comprenant des classes de sixième ;</li>
    <li>D'une section anglophone comprenant des classes de Form I et de Form II.</li>
  </ul>
</p>
```

On obtient la page web suivante dans notre navigateur :

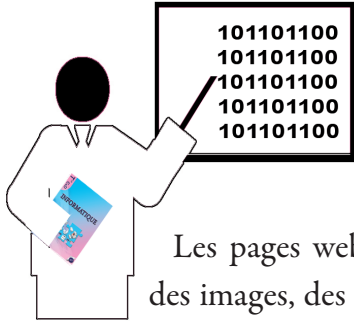
Bienvenue sur le site des élèves du Lycée Bilingue du Cameroun

Présentation générale

Le Lycée Bilingue du Cameroun est un établissement scolaire d'enseignement général secondaire bilingue. Il est composé :

- D'une section francophone : comprenant des classes de sixième ;
- D'une section anglophone comprenant des classes de Form I et de Form II.

Figure 2.3 : Page web avec balisage



RESUME

Les pages web sont des documents informatiques qui peuvent contenir du texte, des images, des tableaux, des formulaires et d'autres éléments multimédias. Elles sont créées à l'aide des langages HTML et CSS, et sont visualisées par les internautes grâce à un navigateur Web comme Mozilla Firefox ; Netscape Navigator ou Internet Explorer. Le navigateur ou browser est un logiciel qui a pour rôle d'analyser le code HTML d'une page et d'en produire un résultat visuel, facile à lire.

La rédaction du contenu des pages Web en HTML repose sur l'utilisation des balises paires ou uniques pouvant contenir des attributs. Le rôle des attributs est de compléter une balise pour donner des informations supplémentaires.

EXERCICE
THEORIQUE**Exercice 1 : Répondre aux questions suivantes.**

1. Définir les termes suivants : page web, site web, navigateur, balise
2. Citer les éléments contenus dans une page web en HTML
3. Lister trois exemples de navigateurs
4. Donnez les balises qui permettent de créer des tableaux, de formater un texte et d'insérer une image.
5. Donner la différence entre un site web statique et un site web dynamique

EXERCICE
THEORIQUE**Exercice 2 : Répondre par vrai ou faux**

1. Le HTML est un langage qui permet de réaliser la mise en page et le graphisme de la page web
2. Le CSS est un langage de description de page web
3. Les pages web sont des documents textes
4. Toutes les balises html fonctionnent par paires
5. La balise permet d'insérer un tableau dans une page web

EXERCICE
THEORIQUE**Exercice 3 : A l'aide des mots : Uniques, logiciel, Pages web, attributs, browser,**

1. Saisir les balises de la page web
2. Exécuter le fichier à l'aide d'un navigateur
3. Ouvrir un éditeur de texte
4. Enregistrer le fichier au format HTML

EXERCICE
PRATIQUE**Exercice 4 : Utilisation des balises du langage html**

On désire réaliser une page web d'un établissement contenant les différents cycles, les classes contenues dans chaque cycle et une description de l'établissement.

1. Donner le squelette d'une page web
2. Insérer les balises qui permettent de créer des listes non ordonnées pour les différents cycles
3. Insérer les balises qui permettent de créer une sous liste comportant les différentes classes contenues dans chaque cycle
4. Insérer la balise qui permet de créer un paragraphe dans lequel vous décririez votre établissement

Exécuter le fichier afin de visualiser le résultat.

Module 1

LES RÉSEAUX INFORMATIQUES

Le développement de l'informatique ne se résume plus à des calculs de plus en plus élaborés et sophistiqués, ni à la mise à disposition d'un ordinateur pour le plus grand nombre, mais plus à la mise en commun des connaissances et des expériences de tous les hommes dans un vaste village planétaire. Les réseaux informatiques sont l'outil incontournable pour la réalisation de cet objectif.

Qu'ils soient locaux ou étendus, filaires ou sans fil, privés ou d'entreprises, les réseaux informatiques obéissent tous à la même logique qu'il est indispensable de comprendre :

- Un ensemble d'ordinateurs et d'équipements à interconnecter ;
- Un protocole de communication à installer ;
- Une architecture pour lier le tout.

Imprimante
192.168.0.9



Desktop1
192.168.0.2



Desktop2
192.168.0.3



Desktop3
192.168.0.4



Switch



Protocole de communication



Point d'accès
192.168.0.1



Desktop 4
192.168.0.5



Laptop
192.168.0.6



Tablet PC
192.168.0.8



Smartphone
192.168.0.7

3

CREATION D'UN RESEAU LOCAL

SOMMAIRE

- ▶ Introduction
- ▶ Intérêt et impact de l'interconnexion en réseau
- ▶ Eléments d'un réseau
- ▶ Classification des réseaux
- ▶ Architecture des réseaux

MOTS-CLES

- Réseau filaire
- Réseau sans fil
- Protocoles
- PAN, LAN, MAN, WAN
- Client/Serveur

Imprimante



Desktop1



Desktop2



Desktop3



Câble



Switch



Protocole de communication



Point d'accès



Desktop4



Laptop



Tablet PC



Smartphone

OBJECTIFS

- Définir la notion de réseau informatique
- Citer les types de réseaux et leur caractéristiques
- Lister des topologies réseau
- Citer les équipements réseaux et donner leur rôle

De tout temps, les hommes ont cherché à améliorer leur mobilité et leurs échanges (transport des matières, des personnes et des informations). Plusieurs canaux ont été utilisé à cette fin. Ce sont :

- Les voies terrestres ;
 - Les voies fluviales ;
 - Les voies aériennes.
1. Lister les différents inconvénients de chacun de ces modes de transport ;
 2. Dire en quoi chaque mode de transport a permis de résoudre les difficultés que posaient les autres ;
 3. Lister les véhicules correspondants à chacun des modes de transport ;
 4. Les véhicules permettent de relier des points appelés noeuds. Lister les noeuds correspondants à chaque mode.

Pour échanger des informations, l'homme a également utilisé plusieurs moyens :

5. Lister les types de moyens utilisés pour transmettre les informations ;
6. Prenons le cas de la poste :
 - Comment appelle-t-on celui qui écrit la lettre ?
 - Comment appelle-t-on celui qui reçoit la lettre ?
 - Où dépose-t-on la lettre ?
 - Quel chemin suit la lettre ?
 - Quels sont les endroits par lesquels la lettre passe ?
 - Où est-ce que celui qui reçoit la lettre va-t-il la chercher ?
 - Quels sont les inconvénients de la poste ?

Pour communiquer par la voix, on utilise le téléphone.

7. Quels les types de téléphones que vous connaissez ?
8. Quels moyens de transmissions sont utilisés par les types de téléphones ?
9. Comment appelle-t-on un ensemble de noeuds reliés pour assurer une communication ?

De nos jours, tous les moyens de communication énoncés sont plus ou moins encore utilisés.

10. Quel est le moyen de communication le plus utilisé de nos jours ?
11. Quels sont les éléments qui permettent de mettre en place ce moyen de communication

Introduction

Le besoin de communication entre individus et organisations, de même que le besoin d'informations et d'amélioration de la mobilité ont favorisé la création des réseaux informatiques. Ainsi, ces réseaux ont eu un impact significatif dans nos vies, car ils ont significativement changé notre façon de vivre et de travailler à travers l'utilisation d'Internet et des différents services qu'il offre.

I. Intérêt et impact de l'interconnexion en réseau

1. Intérêts de l'interconnexion en réseau

La mise en réseau des équipements a pour but de :

- *Partager les ressources* : Un utilisateur peut partager un fichier ou un programme avec les autres utilisateurs afin qu'ils s'en servent si la nécessité se présente.
- *Améliorer la communication humaine et l'échange d'informations* : En permettant entre autres une communication directe entre les différentes personnes (téléenseignement, commerce électronique, etc.)

2. Impact de l'interconnexion en réseau

Les réseaux ont fortement impacté les communications entre personnes par :

- *La diminution des coûts* : en éliminant les distances, les réseaux ont permis la mise à disposition des informations à moindre coût (exemple : livre numérique acheté au dixième de sa valeur, y compris le frais d'acheminement) ;
- *Le gain de temps* : les réseaux permettent la mise à jour en temps réel (au fur et à mesure) des informations ;
- *La modification des habitudes sociales et culturelles* : les réseaux ont permis le développement des attitudes colla-



DEFINITION

Internet :

Interconnexion à l'échelle planétaire de réseaux hétérogènes autour du protocole TCP/IP



DEFINITION

Réseau informatique :

Ensemble d'ordinateurs et d'équipements informatiques interconnectés permettant de faire circuler les données informatiques dans le but d'échanger du texte, des images, de la vidéo ou du son entre ces équipements selon des règles bien définies



DEFINITION

Ressources :

Éléments matériels et logiciels partageables dans un réseau. Ce sont généralement des fichiers, des programmes, ou des équipements (imprimantes, disque, ...)

boratives (travail en groupe) et l'éclosion, notamment avec Internet, d'une génération à la culture pluridimensionnelle.

II. Les éléments d'un réseau

La création d'un réseau informatique nécessite cinq principaux éléments :

- *Les messages* : ils transitent d'un périphérique à un autre ;
- *Les supports de transmission* : ils permettent de transporter les messages d'un périphérique à un autre ;
- *Les périphériques du réseau* : ils échangent des informations entre eux ;
- *Le protocole de communication* : il définit les règles de communication ;
- *La méthode d'accès au support de transmission* : elle définit le procédé qu'utilisent les ordinateurs pour transférer leur message sur le support.

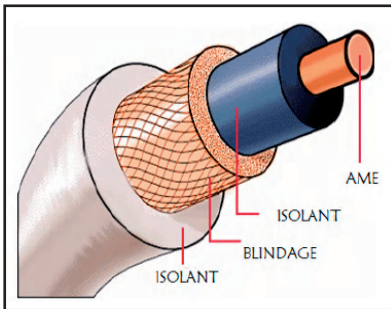


Figure 3.1 : câble coaxial

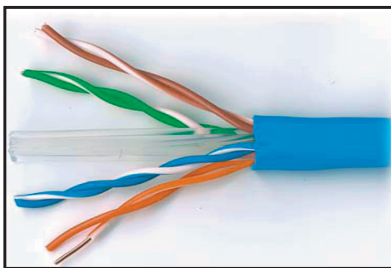


Figure 3.2 : paire torsadée

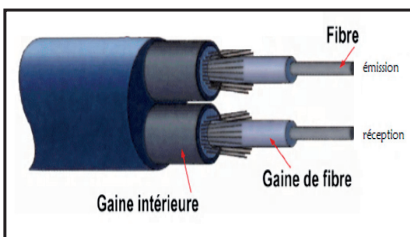


Figure 3.3 : fibre optique

1. Les supports de transmission

L'interconnexion d'un réseau dépend du support utilisé pour connecter les équipements du réseau. Ainsi, on distingue les connexions câblées et les connexions sans fil.

a. Les connexions câblées

Ici, on distingue deux types de support :

- *Les supports métalliques* : c'est généralement le cuivre, qui transporte les signaux électriques. Au nombre des supports métalliques, on cite les câbles à paires torsadées et les câbles coaxiaux. Les câbles sont similaires aux câbles utilisés dans la télévision. Ils utilisent des connecteurs BNC (British Naval Connector). Les câbles coaxiaux quant à eux utilisent des connecteurs RJ45 (semblables à ceux du téléphone) ;
- *Les supports de verre* : c'est la fibre optique, qui transporte les signaux lumineux.

b. Les connexions sans fil

Dans les connexions sans fil, le support de transmission est l'atmosphère terrestre où les signaux transmis sont des hyperfréquences.

2. Les messages

Lors de la transmission d'un message d'un ordinateur vers un autre, ce message est converti dans un format pouvant être transmis sur le réseau. Tous les messages doivent être convertis en bits, c'est-à-dire en signaux numériques codés en binaire, avant d'être envoyés vers leur destination, ceci quel que soit le format d'origine du message (texte, vidéo, audio ou données informatiques). Une fois le message converti en bits, il est prêt à être envoyé sur le réseau, jusqu'à son destinataire.

3. Les périphériques du réseau

Ce sont des périphériques indispensables pour l'émission, la transmission et la réception des messages sur le réseau. On en distingue deux :

a. Les périphériques finaux

- *Les ordinateurs de bureau* : Ce sont des ordinateurs à usage personnel, familial ou professionnel (dans un lieu de travail).
- *Les ordinateurs portables* : Ce sont des ordinateurs possédant une batterie qui emmagasine le courant électrique. Sa particularité est qu'il peut être facilement déplacé et utilisé dans tout lieu puisqu'il ne nécessite d'être alimenté que lorsque la batterie intégrée est faible ;
- *Les serveurs* : Les serveurs sont des ordinateurs qui jouent un rôle particulier dans un réseau. Ils permettent de répondre aux demandes (requêtes) formulées par les autres ordinateurs (clients). Ses caractéristiques ainsi que ses performances sont bien meilleures qu'un ordinateur portable ou de bureau.



Figure 3.4: Ordinateur de bureau



Figure 3.5: Ordinateur portable

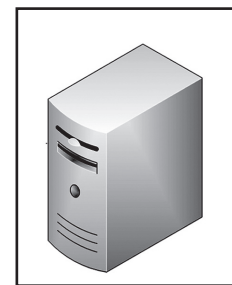


Figure 3.6 : Serveur

Noms anglo-saxons :

Desktop ou ordinateur de bureau
Laptop ou ordinateur portable
Bridge ou pont
Hub ou Concentrateur
Switch ou Commutateur
Firewall ou Pare-feu



REMARQUE



REMARQUE

Client et serveur :

Dans un réseau informatique, les termes « serveur » et « client » désignent à la fois des ordinateurs, mais aussi les applications qui forment des requêtes (applications clientes) et celle qui répondent aux requêtes (serveur Web, serveur d'applications, serveur de données)



REMARQUE

Concentrateur vs commutateur :

Un commutateur permet de ne diffuser les paquets qu'aux ordinateurs concernés par le message, alors que le concentrateur envoie systématiquement les paquets à tous les ordinateurs connectés

b. Les périphériques intermédiaires

- **La carte réseau** : elle sert d'interface physique entre l'ordinateur et le câble réseau. Elle transfère et contrôle le flux de données entre l'ordinateur et le câble. Elle dispose d'un numéro d'identification unique appelé *Media Access Control (MAC)*. Elle est encore appelée *adaptateur réseau* (Network Interface Card, NIC) ;
- **Le répéteur** : il s'agit d'un périphérique non intelligent, qui répète automatiquement les signaux qui lui arrivent et transmettent d'un support vers un autre support. Dans le même temps, le répéteur régénère les signaux, ce qui permet de prolonger le support physique vers un nouveau support physique ;
- **Le pont** : il permet de passer d'un réseau vers un autre réseau de même type. Il filtre les messages et laisse passer ceux destinés au réseau raccordé. En d'autres termes, un pont ne retransmet que les messages dont l'adresse correspond à une machine située sur le réseau raccordé ;
- **Le concentrateur** : il permet, comme son nom l'indique, de concentrer le trafic provenant de différents équipements terminaux. Cela peut se réaliser par une concentration du câblage en un point donné ou par une concentration des données qui arrivent simultanément par plusieurs lignes de communication ;
- **Le commutateur** : il permet de connecter plusieurs autres périphériques. C'est un périphérique intelligent car il est capable de connaître à quel ordinateur est destiné le message pour l'envoyer uniquement à celui-ci ;
- **Le pare-feu** : il permet d'empêcher l'intrusion dans un réseau. C'est un périphérique qui assure la sécurité dans un réseau ;
- **La passerelle** : c'est un système matériel et logiciel qui permet de relier des réseaux qui utilisent des protocoles diffé-

rents. En l'absence d'un tel équipement, un des ordinateurs peut servir de passerelle. Il devra alors rester en permanence allumé ;

- **Le routeur** : il permet d'interconnecter plusieurs réseaux. Il contribue à orienter les messages d'un réseau à un autre. Il va déterminer le plus court chemin entre l'ordinateur émetteur et l'ordinateur destinataire. il intègre des fonctions de passerelle et de pare-feu ;
- **Les points d'accès ou Access Point** : ils permettent de relier des équipements sans fil à un réseau filaire. Ce sont généralement des routeurs sans fil.

4. Le protocole de communication

Les individus désirent souvent envoyer et recevoir des messages divers et variés à l'aide d'applications informatiques, qui ont besoin que le réseau leur fournisse certains services (World Wide Web, la messagerie électronique, la messagerie instantanée, téléchargement). Les périphériques connectés les uns aux autres dans le but de fournir ces services sont régis par des règles, ou protocoles.

Aujourd'hui, la norme en matière de réseaux est un ensemble de protocoles appelé TCP/IP (Transmission Control Protocol/Internet Protocol). TCP/IP est non seulement utilisé dans les réseaux privés et professionnels, mais c'est le principal protocole d'Internet. C'est lui qui définit les mécanismes de formatage, d'adressage et de routage utilisés pour veiller à ce que nos messages soient livrés aux destinataires appropriés.

5. La méthode d'accès

Elle décrit la manière avec laquelle les informations circulent dans le réseau (topologie logique). Une méthode d'accès dépend étroitement de la disposition spatiale des équipements dans le réseau (la topologie physique).



REMARQUE

Équipement actif vs équipement passif :

Certains équipements, dits *actifs*, permettent de générer le signal échanger (ordinateur, commutateur, routeur). D'autres, dits *passifs*, diffuse simplement le signal (répéteur, ponts, concentrateur)



DEFINITION

Protocole :

Ensemble de règles et de formats de données à respecter pour échanger des données dans de bonnes conditions entre deux équipements ou deux programmes.

a. Les topologies physiques

• *La topologie en bus*

Dans cette topologie, les ordinateurs sont disposés et reliés de part et d'autre d'un câble principal appelé *bus*. Le support de transmission utilisé dans ce cas est le câble coaxial. Lorsqu'un ordinateur envoie un message, chaque ordinateur situé sur le chemin intermédiaire reçoit le message et l'ouvre s'il est le destinataire. Dans la figure 1.7 ci-dessous, si l'ordinateur nommé A veut envoyer un message à l'ordinateur E, les ordinateurs B, C et D vont recevoir et retransmettre le même message, mais seul l'ordinateur E va l'ouvrir.

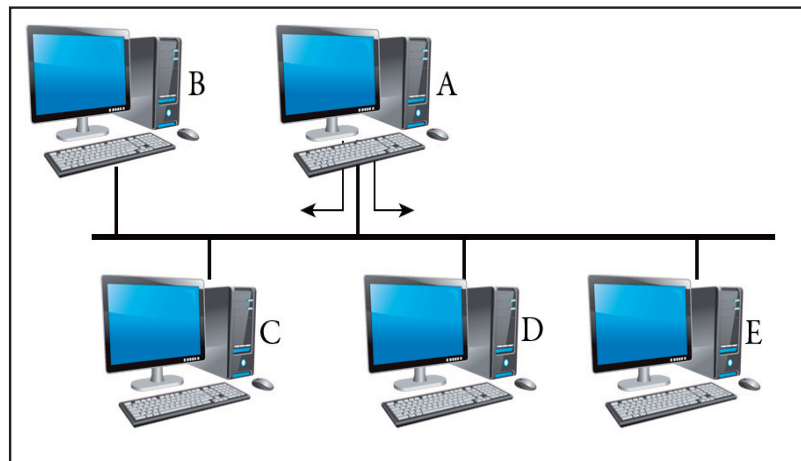


Figure 3.7 : réseau en bus

L'inconvénient de cette topologie est qu'elle perd en performance lorsque le nombre d'ordinateurs augmente. De plus, si le bus se rompt, la communication entre les ordinateurs est impossible.

• *La topologie en étoile*

Dans cette topologie, les ordinateurs sont tous reliés à un équipement central. Ce dernier joue le rôle d'intermédiaire, car tous les messages passent d'abord par lui. C'est lui qui dirige les messages à destination. Le support de transmission dans ce cas est le câble à paires torsadées. Dans la figure 1.8 ci-dessous, Lorsque l'ordinateur A envoie un message à l'ordinateur D, l'équipement central reçoit



DEFINITION

Topologie physique :

Elle désigne le mode d'interconnexion physique des différents éléments du réseau (structure des chemins de câbles, type de raccordement...)



REMARQUE

Le vendeur ambulancier à la criée :

Dans un village qui a une seule route principale, un vendeur ambulancier crie pour présenter ses produits. Les habitants écoutent ces appels et ouvrent leurs portes s'ils sont intéressés. Analogiquement à la figure 1.7 ci-contre, la route est le bus, les habitants les ordinateurs et le vendeur le message.

le message et le retransmet à l'ordinateur D (si c'est un Switch) et aux ordinateurs B, C et D (si c'est un Hub). C'est pourquoi son principal inconvénient réside au niveau de l'équipement central, car s'il tombe en panne, les communications sont impossibles.

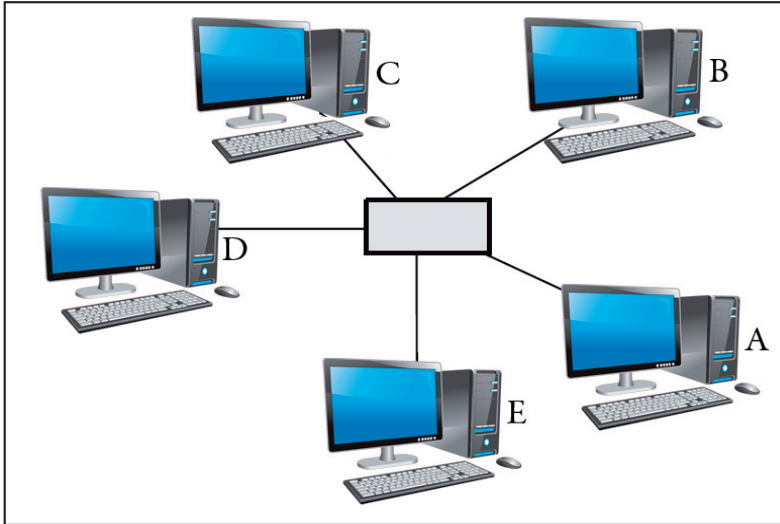


Figure 3.8 : réseau en étoile

- *La topologie en anneau*

Dans cette topologie, les ordinateurs sont connectés sur un câble en forme de boucle ou anneau. C'est cet anneau qui achemine les messages suivant un sens bien défini. Le support de transmission dans ce cas est le câble coaxial. A travers le schéma 1.9 ci-dessous, si l'on suppose que l'ordinateur A veuille envoyer un message à l'ordinateur C, et que le sens de déplacement des messages est celui des aiguilles d'une montre, l'ordinateur D va d'abord recevoir puis va redéposer le message sur l'anneau car celui-ci ne lui est pas destiné. Le message continue donc son bout de chemin jusqu'à ce que l'ordinateur C le reçoive et l'ouvre.

Avec cette topologie, lorsqu'un ordinateur tombe en panne, le réseau est paralysé car les ordinateurs en panne sur l'anneau ne pourront pas retransmettre les messages se trouvant sur leur chemin.

Le messager du Chef:



Dans un village, les habitants présentent leurs litiges au Chef. Celui-ci envoie son messager appeler les autres habitants concernés par le litige pour confrontation. Analogiquement à la figure 1.8 ci-contre, le Chef et son messager constituent l'équipement central et les habitants les ordinateurs.



REMARQUE

Le vendeur ambul- lant porte-à-porte :

Dans un village un vendeur ambulant se présente à chaque porte de maison pour présenter ses produits. Les habitants qui sont intéressés le reçoivent. Ceux qui ne le sont pas le laisse continuer son chemin. Analogiquement à la figure 1.9 ci-contre, les habitants sont les ordinateurs et le vendeur le message.

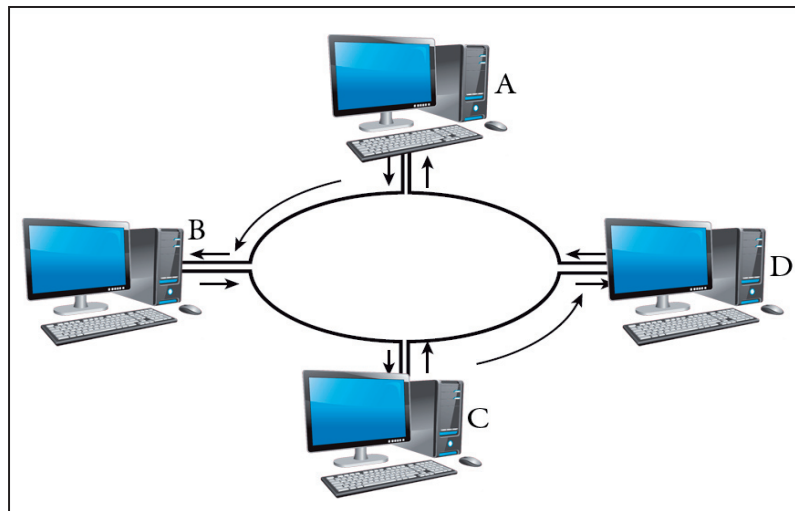


Figure 3.9: réseau en anneau

- *La topologie maillée*

Dans cette topologie, chaque ordinateur est relié aux autres par un câble. Lorsqu'un ordinateur veut envoyer un message, ce dernier transite directement à travers le câble qui mène au destinataire. A travers la figure 1.10 ci-dessous, si l'ordinateur A veut envoyer un message à destination de l'ordinateur C, le message va transiter sur le câble qui lie les deux machines jusqu'à destination.



REMARQUE

Tous se connaissent :

Dans un petit village, tous les habitants se connaissent. Lorsqu'un habitant souhaite passer une informations à un autre, il va directement à la maison de ce-ci le trouver. Analogiquement à la figure 1.10 ci-contre, les habitants sont les ordinateurs.

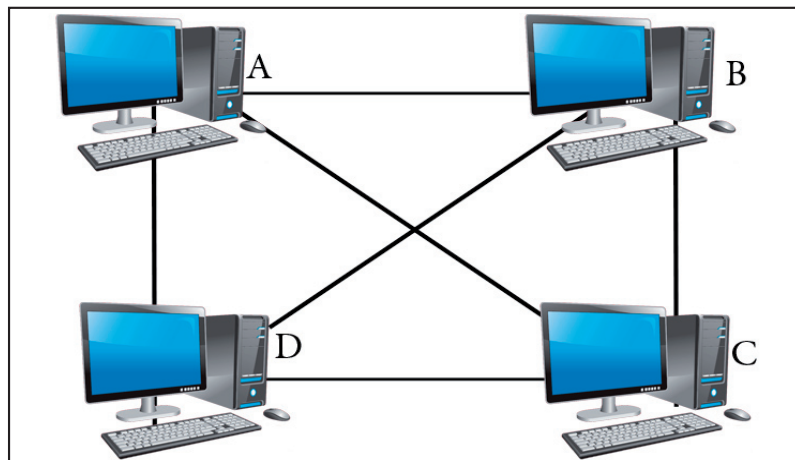


Figure 3.10 : réseau en maillé

Le principal inconvénient de cette topologie de réseau est lié au nombre important de câbles à déployer pour un réseau de plusieurs machines. Très vite ces câbles deviennent encombrants.

b. Les topologies logiques

Une fois la topologie physique construite, il est nécessaire de définir le procédé qu'utilisent les ordinateurs pour transmettre les informations sur le support de transmission ainsi que la manière avec laquelle ces informations vont circuler sur ce support. Pour ce faire, il existe plusieurs topologies logiques :

- *Ethernet*

Lorsqu'un ordinateur veut émettre un message, il vérifie d'abord qu'aucun message ne circule sur le réseau. Si c'est le cas, il envoie le message sur le support. Dans le cas contraire, c'est un autre ordinateur qui est entrain d'émettre : on dit qu'il ya *collision*. Cette topologie logique est utilisée dans les topologies physiques en bus et en étoile.

- *FDDI*

La topologie logique FDDI (Fiber Distributed Data Interface) est une topologie qui utilise la fibre optique comme support de transmission pour transmettre les informations. Elle est basée sur la topologie physique en anneau. Pour émettre un message sur l'anneau, chaque ordinateur a besoin d'un jeton qui donne le droit d'envoyer les données sur le support. Le jeton fait le tour de l'anneau. Lorsqu'un ordinateur le reçoit, il le garde jusqu'à ce qu'il ait fini de transmettre son message. S'il n'a pas de message à transmettre, il laisse passer le jeton qui continue son bout de chemin sur l'anneau.

- *Token ring*

Token Ring (en français anneau à jeton) est une topologie logique qui fonctionne également sur la base d'un jeton pour l'accès au support. Elle est utilisée dans la topologie physique en étoile l'envoi des messages.



DEFINITION

Topologie logique :

Encore appelé topologie d'échange, la topologie logique désigne le mode de circulation des données sur le support de transmission.



REMARQUE

Collision :

On parle de collision lorsque deux stations émettent simultanément, leurs signaux se superposent et chaque émetteur ne reconnaît plus son message sur le support

III. Classification des réseaux

Plusieurs critères permettent de classer les réseaux. On peut citer comme critères : la taille de la couverture géographique et la nature de la liaison existante entre les périphériques.

1. La taille de la couverture géographique

C'est la classification traditionnelle (voir figure 1.11), fondée sur la notion d'étendue géographique. Elle correspond à un ensemble de contraintes à prendre en compte lors de la réalisation de son réseau. Généralement, on adopte la terminologie suivante:

- **PAN** (*Personal Local Area Network en français réseau personnel*) : réseau qui permet de connecter des périphériques sur quelques mètres. Il est encore appelé *réseau domestique*.
- **LAN** (*Local Area Network en français réseau local*) : c'est un réseau local d'étendue limitée à une aire géographique réduite (salle, bâtiment, ...). On l'appelle encore *réseau d'entreprise*.
- **MAN** (*Metropolitan Area Network en français réseau métropolitain*) : d'une étendue de l'ordre d'une centaine de kilomètres. Ils sont généralement utilisés pour fédérer des réseaux locaux (réseau de campus, le réseau des réseaux des agences de MTN ou d'Orange, ...).
- **WAN** (*Wide Area Network*) : ces réseaux assurent généralement le transport d'information sur de grandes distances.

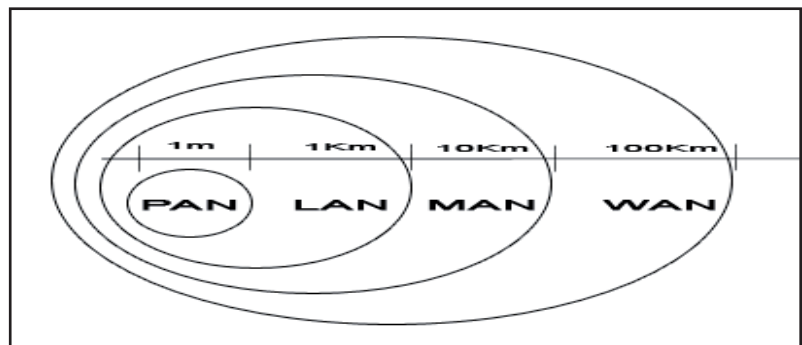


Figure 3.11: Type de réseau en fonction de l'étendue géographique

2. Nature de la liaison

Les périphériques peuvent être connectés de deux manières, soit par câble ou sans câble. Ainsi, on distingue les réseaux filaires des réseaux sans fils.

a. Les réseaux filaires

Les périphériques sont interconnectés à l'aide d'un support de transmission câblé : câble coaxial, câble à paires torsadées ou fibre optique.

b. Les réseaux sans fil :

Les périphériques sont interconnectés à l'aide d'un support de transmission non câblé ou sans fil : ondes radioélectriques.

Les réseaux sans fil sont tributaires de la propagation des ondes qui sont généralement freinées par le milieu qu'elles traversent. Ainsi, la puissance du signal dans un réseau sans fil va connaître un affaiblissement :

- **Faible** : lorsque les ondes doivent traverser des portes ou des cloisons en bois ou en plastique ;
- **Moyen** : lorsque les ondes doivent traverser des portes ou des cloisons en verre, en brique ou en plâtre ;
- **Elevé** : lorsque les ondes doivent traverser des portes ou des cloisons en béton, en vitre blindée ou en métal.

De plus, les technologies de transmission des ondes dépendent de la distance à couvrir (voir figure 1.12) :

- **Les WPAN (Wireless PAN)** : on utilise le *Bluetooth*, *l'infrarouge* ou le *HomeRF (Home Radio Frequency)* ;
- **Les WLAN (Wireless LAN)** : on utilise le *WIFI (Wireless Fidelity)* ou *l'Hiperlan2 (High Performance Radio LAN 2)*
- **Les WMAN (Wireless MAN)** : on utilise le *Wimax (World-wide Interoperability for Microwave Access)* ;

- Les **WWAN (Wireless WAN)** : on utilise le **GSM (Global System for Mobile Communication** ou en français **Groupe Spécial Mobile**), le **GPRS (General Packet Radio Service)** ou **UMTS (Universal Mobile Telecommunication System)**

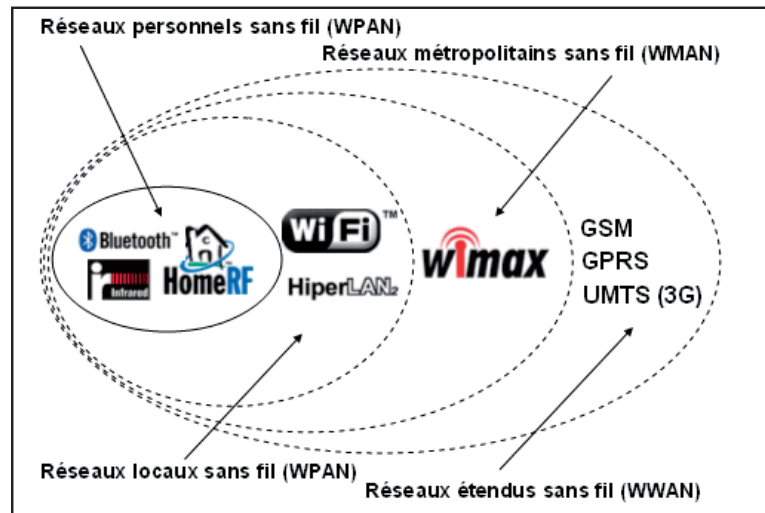


Figure 3.12: Type de réseau en fonction de l'étendue géographique

IV. Les architectures réseaux

Une architecture réseau définit le rôle que jouent les ordinateurs dans le réseau. Chaque ordinateur peut jouer deux rôles :

- **Client** : c'est l'ordinateur qui initie les requêtes puis attend les réponses du serveur
- **Serveur** : c'est l'ordinateur qui est à l'écoute, en attente des requêtes des clients. Dès qu'une requête arrive, il traite et renvoie la réponse au client.

Il existe plusieurs architectures réseaux : les architectures d'égal à égal, infrastructure et client-serveur.

1. L'architecture d'égal à égal

L'architecture d'égal à égal (en anglais *peer-to-peer*) est un type d'architecture réseau dans lequel chaque ordinateur est à la fois client et serveur. Autrement dit, chaque ordinateur possède des

ressources qu'il met à la disposition des autres. De plus, Chaque ordinateur peut solliciter des ressources des autres ordinateurs au travers d'une requête bien formulée. Dans le langage des réseaux sans fil, cette architecture est appelé *ad hoc*.

Cette architecture a pour inconvénients :

- Une sécurité faible
- Une architecture non centralisée rendant difficile l'administration

2. L'architecture client/serveur

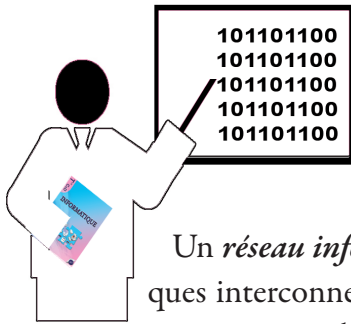
L'architecture client-serveur est un mode de communication dans lequel on distingue un serveur parmi plusieurs clients. Chaque logiciel client peut formuler des requêtes à destination du serveur. Ce dernier peut être spécialisé en serveur d'applications, de fichiers, de données ou de messagerie électronique.

Les avantages de cette architecture sont :

- La centralisation des ressources au niveau du serveur : il peut gérer des ressources communes à plusieurs ordinateurs, par exemple une base de données centralisée ou encore des pages Web d'un site Internet.
- Une meilleure sécurité
- Le caractère évolutif du réseau : l'ajout ou la suppression des clients n'affecte pas le fonctionnement du réseau.

3. L'architecture infrastructure

Elle est spécifique aux réseaux sans fil. Chaque ordinateur ou station se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations dans sa zone de couverture est appelé *ensemble de services de base (BBS, Basic Service Set)*.



RESUME

Un *réseau informatique* est un ensemble d'ordinateurs et d'équipements informatiques interconnectés, permettant de faire circuler les données informatiques entre ces équipements selon des règles bien définies. L'objectif de la création des réseaux informatiques est de partager les ressources (fichiers, des programmes, ou équipements), d'améliorer la communication humaine et l'échange d'informations. La communication dans un réseau nécessite cinq principaux éléments : des messages (suite binaire), des supports de transmission (câble coaxial, câble à paires torsadées, fibre optique, hyperfréquences), des périphériques (desktop, laptop, répéteur, hub, Switch, routeur, ...), un protocole de communication (TCP/IP ...) et une méthode d'accès au support de transmission (Ethernet, FDDI, Token ring).

On peut classer les réseaux suivant la taille de la couverture géographique (PAN, LAN, MAN, WAN) ou la nature de la liaison existante entre les périphériques (réseau filaire et sans fil).

Dans un réseau informatique, chaque ordinateur peut jouer le rôle de client ou de serveur ou les deux à la fois. Le réseau dans lequel les ordinateurs jouent à la fois le rôle de client et de serveur est appelé architecture d'égal à égal. Tandis que l'architecture client/serveur est un mode de communication dans lequel on distingue un serveur parmi plusieurs clients.

EXERCICE
THEORIQUE**Exercice 1 : Répondre aux questions suivantes**

- 1) Définir les termes : réseaux informatiques, protocole, internet, client, serveur, support de transmission, architecture réseau ;
- 2) Donnez l'intérêt des réseaux informatiques ;
- 3) Quelles sont les ressources partageables dans un réseau ?
- 4) Quels sont les éléments constitutifs d'un réseau ?
- 5) Montrer la différence existante entre une topologie physique et une topologie logique ;
- 6) Citer 3 exemples de topologie physique et 3 exemples de topologie logique
- 7) Donner les critères de classification des réseaux ;
- 8) Citer les équipements d'interconnexion d'un réseau et donner pour chacun son rôle ;
- 9) Citer des exemples de support d'interconnexion dans un réseau informatique ;
- 10) Donner la différence entre un réseau filaire et un réseau sans fil ;
- 11) Citer les catégories de réseau sans fil et donner pour chacune un exemple de technologie sans fil ;
- 12) Citer les architectures réseau. Préciser les avantages et les inconvénients de chaque architecture.

EXERCICE
THEORIQUE**Exercice 2 : Répondre par vrai ou faux**

1. L'interconnexion en réseau a permis la diminution des coûts et le gain de temps entre autres ;
2. Les câbles à paires torsadées utilisent des connecteurs RJ45 ;
3. Lors de la transmission d'un message d'un ordinateur vers un autre, le message est converti en bits ;
4. Un ordinateur de bureau est encore appelé «Laptop» ;
5. Une méthode d'accès dépend étroitement de la disposition spatiale des équipements dans le réseau ;
6. La topologie logique Ethernet est utilisée dans les topologies physiques en étoile ;
7. la topologie maillée utilise très peu de câbles pour un réseau de plusieurs machines ;
8. Le réseau d'un bâtiment est appelé réseau domestique ;
9. L'architecture client/serveur est un mode de communication dans lequel on distingue un client parmi plusieurs serveurs.

EXERCICE
THEORIQUE

Exercice 3 : Compléter les phrases à trous par la liste de mots suivants : TCP/IP, sans fil, réseau, périphériques finaux, support de transmission, câble, ondes radioélectriques, intermédiaires, finaux.

1. Les périphériques _____ sont à la fois ceux qui initient les messages et ceux à qui les messages sont destinés.
2. Les périphériques _____ sont ceux qui transmettent les messages de l'émetteur vers le destinataire.
3. La carte _____ sert d'interface physique entre l'ordinateur et le _____ réseau.
4. La norme en matière de réseaux est un ensemble de protocoles appelé _____
5. Avec le Wifi, les _____ sont interconnectés à l'aide d'un _____ non câblé ou _____ appelés : _____

4

CONFIGURATION D'UN RESEAU LOCAL

SOMMAIRE

- ▶ Le protocole de communication
- ▶ L'adressage des ordinateurs

MOTS-CLES

- TCP/IP
- Segment
- Datagramme
- Paquet
- Trame
- Adresse IP
- Masque de sous-réseau
- Point d'accès

PRE - REQUIS

- Eléments d'interconnexion d'un réseau

Imprimante
192.168.0.9



Desktop1
192.168.0.2



Desktop2
192.168.0.3



Desktop3
192.168.0.4



Switch



Protocole de communication



Point d'accès
192.168.0.1

Desktop 4
192.168.0.5



Laptop
192.168.0.6



Tablet PC
192.168.0.8



Smartphone
192.168.0.7

OBJECTIFS

- Définir TCP/IP
- Identifier les protocoles utiles dans une communication inter-réseau
- Identifier les adresses privées
- Distinguer l'adressage statique de l'adressage dynamique

Lorsqu'on souhaite expédier un message à une personne par le biais de la poste, on passe par un certain nombre d'étapes :

- On couche notre message sur un support papier à l'aide d'un stylo à bille ;
- On met le papier portant notre message dans une enveloppe ;
- On indique le nom et l'adresse postale du destinataire ;
- On colle un timbre fiscal et on dépose l'enveloppe à la poste.

1. A quelle application peut-on comparer le papier et stylo à bille ?
2. A quoi sert l'enveloppe ?
3. A quoi sert l'adresse postale ?

On souhaite maintenant passer le même message par le biais d'un téléphone portable.

4. A quoi correspond le papier et le stylo à bille ?
5. A quoi correspond l'enveloppe ?
6. A quoi correspond l'adresse du destinataire ?

On souhaite enfin passer le même message par le biais d'Internet.

7. A quoi correspond le papier et le stylo à bille ?
8. A quoi correspond l'enveloppe ?
9. A quoi correspond l'adresse du destinataire ?

De manière générale, peut-on envoyer un message :

- Sans papier ni stylo à bille ?
- Sans enveloppe ?
- Sans adresse de destinataire ?
- Sans système de transport de l'enveloppe ?

Comment appelle-t-on l'ensemble techniques de communication qui permettent d'échanger des messages ?

Introduction

La création d'un réseau s'effectue en deux phases principales : l'interconnexion des périphériques réseaux et leur configuration. L'interconnexion des périphériques consiste à définir une topologie physique et logique utilisant un support qui relie les différents périphériques, tandis que la configuration du réseau revient à définir non seulement le protocole de communication entre les différents périphériques mais aussi les adresses des ordinateurs du réseau, de sorte qu'ils soient identifiables de manière unique dans les communications.

I. Le protocole de communication

Les ordinateurs utilisent un ensemble de protocoles bien hiérarchisés (ou pile de protocoles) qui définit les règles de communication entre les différents ordinateurs. La plupart des réseaux utilisent la pile de protocole TCP/IP (Transmission Control Protocol / Internet Protocol). Cette pile de protocole est constituée d'un ensemble de couches qui résolvent chacune un problème spécifique lié à la transmission des données. De plus, chaque couche fournit des services aux couches supérieures. Les couches inférieures mettent en forme les données afin qu'elles puissent être transmises sur le support. La pile TCP/IP comporte essentiellement 4 couches comme présentée sur la figure suivante 2.1 ci-après:

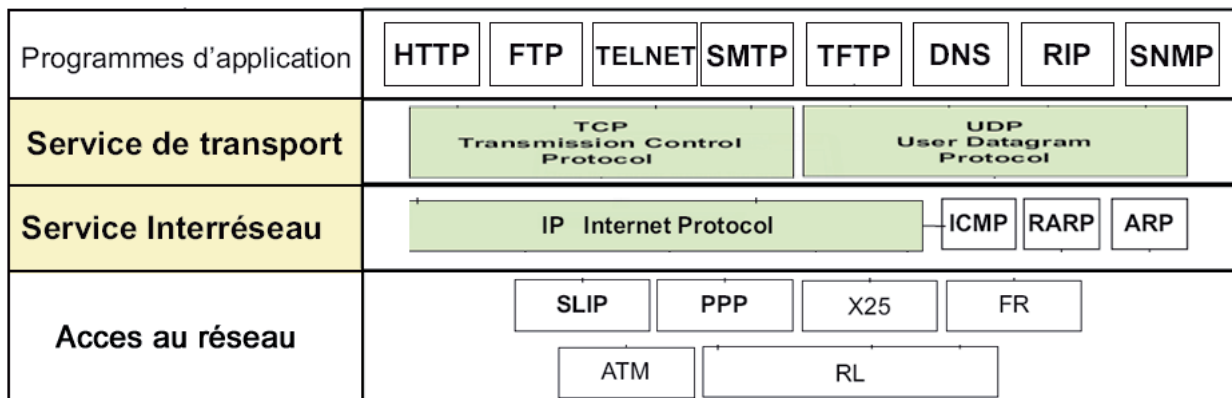


Figure 4.1 : Pile de protocole TCP/IP et protocoles utilisés

1. La couche Application

C'est dans cette couche que se situe la plupart des programmes réseau. Ces programmes utilisent des protocoles pour gérer les communications. Les protocoles de la couche application les plus connus sont ceux permettant l'échange d'informations entre les utilisateurs :

- *HTTP (HyperText Transfer Protocol)* : il est utilisé pour transférer les fichiers qui constituent les pages du Web ;
- *SMTP (Simple Mail Transfer Protocol)* : il est utilisé pour transférer les courriels et les pièces jointes ;
- *FTP (File Transfer Protocol)* : il est utilisé pour le transfert interactif de fichiers entre les ordinateurs.

2. La couche Transport

Les protocoles de la couche transport déterminent à quelle application est destinée les messages envoyés. On distingue :

- *TCP (Transfer Control Protocol)* : il a pour rôle d'envoyer les messages en séquence et dans le bon ordre, en se rassurant qu'ils sont bien arrivés (mode connecté) ;
- *UDP (User Datagram Protocol)* : C'est un protocole qui envoie les messages sans se rassurer qu'ils sont bien arrivés (mode non connecté). De plus il ne garantit pas leur arrivée dans le bon ordre.

3. La couche Interréseau

Cette couche résout le problème d'acheminement des paquets à travers l'adressage des périphériques finaux. Les protocoles mis en œuvre dans la couche réseau qui transportent des données utilisateur comprennent :

- *Le Protocole IP* : Il cherche un chemin pour transférer les données d'un équipement émetteur à un équipement destinataire, identifiés chacune par leur adresse IP ;



REMARQUE

Notion de paquets :

A l'émission, un message est découpé en fragments de petite taille appelé paquets. Chaque paquet est acheminé indépendamment des autres. Les paquets sont ensuite réassemblés à la réception pour reconstituer le message.

- *Le Protocole ICMP* : C'est un mécanisme de contrôle des erreurs au niveau IP. Il permet aux routeurs d'informer les utilisateurs des erreurs de transmission par le protocole IP ;
- *Le Protocole ARP* : Il établit une correspondance dynamique entre adresses physiques (MAC) et adresses logiques (IP). Il permet donc à une machine de trouver l'adresse physique d'une machine cible située sur le même réseau local, à partir de sa seule adresse IP

4. La couche d'Accès au réseau

Le rôle de cette couche est de coder les chiffres binaires contenus dans le message de la couche réseau en signaux, de transmettre et recevoir ces signaux sur le support physique (fils de cuivre, fibre optique et sans fil) reliant des périphériques réseau.

5. L'encapsulation des données

Lorsque les données d'application descendent la pile de protocoles en vue de leur transmission sur le support réseau, différents protocoles ajoutent des informations à chaque niveau : Il s'agit du processus d'*encapsulation*. De même, lors de la réception du message chez le destinataire, les mêmes protocoles retirent les informations à chaque niveau jusqu'à la couche application : il s'agit du processus de *décapsulation*.

La forme qu'emprunte une donnée sur n'importe quelle couche est appelée *unité de données* de protocole. Au cours de l'encapsulation, chaque couche encapsule l'unité de données de protocole qu'elle reçoit de la couche supérieure en respectant le protocole en cours d'utilisation. À chaque étape du processus, une unité de données de protocole possède un nom différent qui reflète sa nouvelle apparence. Les unités de données de protocoles sont nommées en fonction des protocoles de la suite TCP/IP :

La lettre et l'enveloppe :



REMARQUE

Il n'est pas possible de déposer une feuille de papier comme message à la poste. Il faut la plier et l'insérer dans une enveloppe pour qu'elle puisse être traitée. Le changement de forme du message ainsi opéré est appelé *encapsulation*.

- *Message* : terme général pour les unités de données de protocole utilisées au niveau de la couche application ;
- *Segment* : unité de données de protocole de la couche transport ;
- *Datagramme* : unité de données de protocole de la couche réseau ;
- *Trame* : unité de données de protocole de la couche d'accès au réseau ;
- *Bits* : unité de données de protocole utilisée lors de la transmission physique de données à travers le support.

L'encapsulation et la décapsulation des données dans la pile de protocoles TCP/IP sont résumées par la Figure 3.1 ci-après :

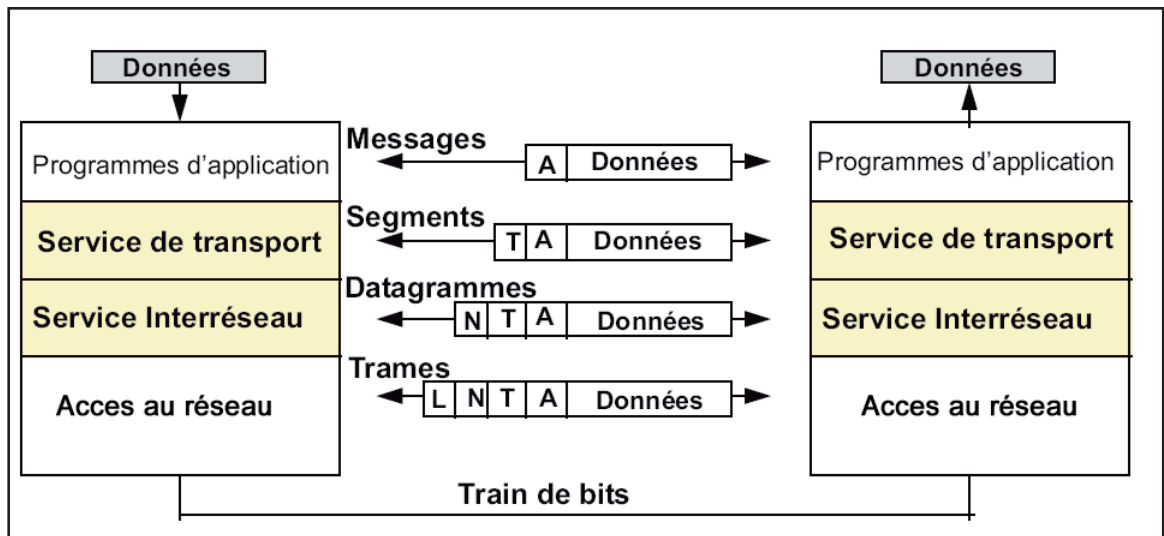


Figure 4.2 : Encapsulation des données dans TCP/IP

II. L'adressage des ordinateurs

L'adressage est une fonction de la couche réseau de la pile de protocole TCP/IP. Il permet de mettre en œuvre la transmission de données entre des périphériques situés sur un même réseau ou sur des réseaux différents. Le protocole IP permet un adressage hiérarchique des paquets qui transportent les données. Pour adresser un

ordinateur, il faut préciser :

- L'adresse IP de l'ordinateur ;
- Le masque de sous-réseau ;
- Le mode d'adressage : statique ou dynamique.

1. L'adresse IP

Tous les périphériques appartenant à un réseau doivent être identifiés de manière unique. Au niveau de la couche réseau, les paquets de communication doivent être identifiés par les adresses source et de destination des périphériques, à l'aide d'une adresse IP. On distingue les adresses publiques des adresses privées.

a. Les adresses publiques

Ce sont des adresses utilisées par les ordinateurs des réseaux qui sont directement connectés à Internet. On recense un grand nombre d'adresse publiques pour les utilisateurs à travers le monde. Elles sont distribuées par l'ICANN (*Internet Corporation for Assigned Names and Numbers*).

Pour retenir des adresses IP publiques, on les associe à des noms en langage courants appelés noms de domaine. Le système qui permet de d'associer une adresse IP à un nom de domaine est appelé *DNS (Domain Name System)*.

b. Les adresses privées

Les ordinateurs ne nécessitant pas l'accès à Internet utilisent des adresses privées appartenant aux trois plages suivantes :

- De 10.0.0.1 à 10.255.255.254 ;
- De 172.16.0.1 à 172.31.255.254 ;
- De 192.168.0.1 à 192.168.255.254 ;



DEFINITION

Adresse IP :

Suite de quatre nombres allant de 0 à 255, séparés par des points permettant d'identifier de manière unique chaque équipement actif dans un réseau.



REMARQUE

Nom de domaine et URL :

Un nom de domaine représente une adresse IP, c'est-à-dire un ordinateur du réseau. Une URL (Uniform Resource Locator) représente une ressource disponible sur cet ordinateur.



REMARQUE

Adresse IP utilisées :

Les adresses IP vont de 0.0.0.0 à 255.255.255.255. Toutefois, seules les adresses de la plage 0.0.0.0 à 223.255.255.255 peuvent être utilisées.

2. Le masque de sous-réseau

C'est une adresse IP qui renseigne sur le réseau dans lequel se trouve un ordinateur. Chaque ordinateur se base sur cette adresse pour savoir si le destinataire fait partie du même réseau que lui. A chaque plage d'adresse IP privée correspond un masque de sous réseau :

- 255.0.0.0 pour la plage commençant par 10 ;
- 255.255.0.0 pour la plage commençant par 172 ;
- 255.255.255.0 pour la plage commençant par 192.

3. Le mode d'adressage

L'attribution d'une adresse IP à un ordinateur peut se faire de deux manières :

- *Manuellement* : on parle d'adressage statique ;
- *Dynamiquement* : on parle d'adressage dynamique.

a. Adressage statique

L'adressage statique consiste à définir manuellement une adresse IP statique ou fixe à chaque ordinateur. Chaque fois qu'un ordinateur se connecte au réseau, il aura la même adresse IP. Lorsque deux ordinateurs ont la même adresse IP, on parle de *conflit d'adresse*.

Les adresses statiques présentent certains avantages. Elles conviennent pour les imprimantes réseau, les serveurs et d'autres périphériques réseau, qui doivent être accessibles pour les clients d'un réseau. En effet, si les hôtes ont l'habitude d'accéder à un serveur à travers un ordinateur particulier (proxy), la modification de l'adresse de celui-ci peut poser des problèmes.. De plus, l'attribution statique des informations d'adressage permet de mieux contrôler les ressources réseau.

Le principal inconvénient de ce mode d'adressage tient du fait



REMARQUE

Notion de proxy :

Un proxy est un serveur intermédiaire entre Internet et le reste des ordinateurs du réseau local. Dans la plupart des cybercafés, l'ordinateur du gérant est un proxy. Les autres ordinateurs doivent connaître l'adresse IP ce de proxy pour pouvoir accéder à Internet.

qu'il n'est pas possible d'utiliser momentanément les adresses IP des ordinateurs qui ne sont pas connectés., la configuration IP sur chaque ordinateur prend du temps.

b. Adressage dynamique

En raison des difficultés associées à la gestion des adresses statiques, les périphériques des utilisateurs se voient parfois attribuer leur adresse de manière dynamique, à l'aide du protocole DHCP (Dynamic Host Configuration Protocol).

Le protocole DHCP permet l'attribution automatique des informations d'adressage, telles que l'adresse IP, le masque de sous-réseau et d'autres paramètres. La configuration du serveur DHCP nécessite qu'un bloc d'adresses, appelé *pool d'adresses*, soit défini de manière à être attribué aux clients DHCP d'un réseau.

Le protocole DHCP est généralement la méthode d'attribution d'adresses IP privilégiée pour les réseaux de grande taille, car le personnel de support du réseau est dégagé de cette tâche et le risque d'erreur de saisie, donc de conflit d'adresse est quasiment éliminé.

L'autre avantage de l'attribution dynamique réside dans le fait que les adresses ne sont pas permanentes pour les hôtes, elles sont uniquement « louées » pour une certaine durée. Si l'hôte est mis sous tension ou retiré du réseau, son adresse est renvoyée au pool et sera réutilisée. Cela est particulièrement intéressant pour les utilisateurs mobiles qui se connectent et se déconnectent d'un réseau.

III. Test du réseau

La configuration d'un réseau se termine par le test. Il s'agit de vérifier que l'ordinateur sont connectés et échangent bien. On utilise pour cela la commande ping.

Ping est un utilitaire qui permet de vérifier si une machine distante est accessible par le réseau. A intervalles réguliers (par défaut chaque seconde), la machine source (celle sur laquelle la commande ping est exécutée) envoie une commande "echo request" à la machine cible. Dès réception du paquet "echo reply", la machine source affiche une ligne contenant un certain nombre d'informations.

Pour exécuter l'utilitaire ping utilise l'une des trois syntaxes suivantes :

```
ping adresse_IP  
ping nom_ordinateur  
ping nom_domaine
```

Exemple 4.1 : Test du réseau avec ping

```
ping 127.0.0.1  
ping 192.168.0.123  
ping desktop_siriki  
ping www.lbc.com
```

- ping sur l'adresse 127.0.0.1 teste l'installation de la pile TCP/IP sur la machine source ;
- ping sur l'adresse 192.168.0.123 permet de vérifier que la station à laquelle correspond à cette adresse IP est correctement configurée. Si cette station est une passerelle par défaut, ping permet de contrôler la validité du masque de sous-réseau et la configuration de la passerelle par défaut ;
- ping sur desktop_siriki et ping sur www.lbc.com permettent de vérifier que les stations auxquelles correspondent le nom d'ordinateur et le nom de domaine sont correctement configurées ;

Suivant le système d'exploitation, l'affichage de la sortie d'une commande ping pourra être légèrement différent.

Exemple 4.2 : Résultat de ping sur Linux :

```
ping www.lbc.com
ping www.lbc.com (163.5.255.85): 56 data bytes
64 bytes from 163.5.255.85: icmp_seq=0 ttl=56 time=7.7 ms
64 bytes from 163.5.255.85: icmp_seq=1 ttl=56 time=6.0 ms
64 bytes from 163.5.255.85: icmp_seq=2 ttl=56 time=5.5 ms
64 bytes from 163.5.255.85: icmp_seq=3 ttl=56 time=6.0 ms
64 bytes from 163.5.255.85: icmp_seq=4 ttl=56 time=5.3 ms
64 bytes from 163.5.255.85: icmp_seq=5 ttl=56 time=5.6 ms
64 bytes from 163.5.255.85: icmp_seq=6 ttl=56 time=7.0 ms
64 bytes from 163.5.255.85: icmp_seq=7 ttl=56 time=6.0 ms
--- www.commentcamarche.fr ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 5.3/6.1/7.7 ms
```

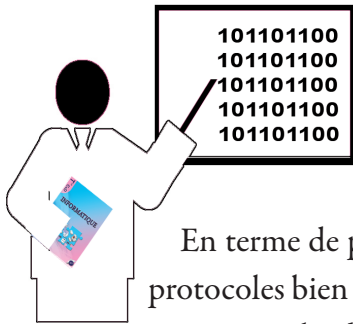
Exemple 4.3: Résultat de ping sur Windows :

```
ping www.commentcamarche.fr
Envoi d'une requête 'ping' sur www.commentcamarche.fr
[163.5.255.85] avec 32 octets de données :
Réponse de 163.5.255.85 : octets=32 temps=34 ms TTL=54
Réponse de 163.5.255.85 : octets=32 temps=37 ms TTL=54
Réponse de 163.5.255.85 : octets=32 temps=32 ms TTL=54
Réponse de 163.5.255.85 : octets=32 temps=33 ms TTL=54
Statistiques Ping pour 163.5.255.85 :
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 32ms, Maximum = 37ms, Moyenne =
34ms
```

La sortie de la commande ping permet ainsi de connaître :

- L'adresse IP correspondant au nom de la machine distante ;
- Le numéro de séquence ICMP ;

- La durée de vie du paquet (TTL, Time To Live). C'est le nombre de routeurs traversés par le paquet lors de l'échange entre les deux machines.
- Le temps de propagation en boucle (round-trip delay) correspondant à la durée en millisecondes d'un aller-retour entre la machine source et la machine cible. Un paquet doit en règle générale posséder un temps de propagation inférieur à 200 ms ;
- Le nombre de paquets perdus.



RESUME

En terme de protocole de communication, les ordinateurs utilisent un ensemble de protocoles bien hiérarchisés (ou pile de protocoles) qui définit les règles de communication entre les différents ordinateurs appelés pile de protocole TCP/IP. Chacune de ses quatre couches définissent des protocoles et des services : la couche application (HTTP, FTP, SMTP), la couche transport (TCP, UDP), la couche d'interresseau (IP, ICMP, ARP), la couche d'accès aux données. Lorsque les données d'application descendent la pile de protocoles en vue de leur transmission sur le support réseau, différents protocoles ajoutent des informations à chaque niveau : Il s'agit du processus d'encapsulation. De même, lors de la réception du message chez le destinataire, les mêmes protocoles retirent les informations à chaque niveau jusqu'à la couche application : il s'agit du processus de décapsulation.

L'adressage permet de mettre en œuvre la transmission de données entre des périphériques situés sur un même réseau ou sur des réseaux différents. Pour adresser un ordinateur, il faut préciser : l'adresse IP de l'ordinateur, le masque de sous-réseau et le mode d'adressage (statique ou dynamique). L'adressage statique est conseillé pour des ordinateurs serveurs et les réseaux de petite taille. Tandis que l'adressage dynamique est utile pour des réseaux ayant un grand nombre d'ordinateurs.

La configuration se termine par le test qui est généralement réalisé par l'utilisateur ping.

EXERCICE
THEORIQUE*Exercice 1 : Répondre aux questions suivantes*

1. Définir les termes suivants : pile de protocoles TCP/IP, adresse IP, adresse privée, segment, trame
2. Donner les quatre couches de la pile de protocoles TCP/IP et leur rôle.
3. Citer les protocoles de communication au niveau de chaque couche
4. Différencier le processus d'encapsulation du processus de décapsulation
5. Quels sont les informations à préciser lors de l'adressage d'un réseau
6. Quel organisme se charge d'attribuer les adresses publiques aux utilisateurs ?
7. Donner les plages d'adresses privées utilisables dans les réseaux locaux
8. Qu'appelle-t-on conflit d'adresse ?
9. Quels sont les avantages et inconvénients de l'adressage statique ?
10. Quels sont les avantages et les inconvénients de l'adressage dynamique ?

EXERCICE
THEORIQUE*Exercice 2 : Répondre par vrai ou faux*

1. La création d'un réseau s'effectue en deux phases principales : l'interconnexion des périphériques réseaux et leur configuration.
2. La pile de protocole TCP/IP est constituée d'un ensemble de couches qui résolvent chacune un problème spécifique lié à la transmission des données.
3. La couche application prend en charge la communication entre différents périphériques à travers divers réseaux.
5. La couche transport permet de représenter des données pour l'utilisateur, ainsi que le codage et de ces données et le contrôle du dialogue.
6. La couche d'interreseau contrôle les périphériques matériels et les supports qui constituent le réseau.
7. La couche d'accès aux données détermine le meilleur chemin à travers le réseau.
8. Le datagramme est l'unité de données de la couche interréseau.
9. Les données qui circulent sur le réseau sont sous forme de trame.
10. L'adressage dynamique consiste à donner une adresse fixe à des ordinateurs qui en font la demande

EXERCICE
THEORIQUE

Exercice 3 : Compléter les phrases à trous par les mots de la liste suivante : Support, serveur, décapsulation, DHCP, logique, pool d'adresse, ordinateurs, statique, encapsulation, périphériques, configuration, dynamique, protocole

1. Lors de l'envoi d'un message, l'ordinateur initial réalise le processus d'_____. Lors de la réception du message, l'ordinateur final réalise le processus de _____.
2. L'interconnexion des _____ consiste à définir une topologie physique et _____ utilisant un _____ qui relie les différents périphériques.
3. La _____ du réseau revient à définir non seulement le _____ de communication entre les différents périphériques mais aussi les adresses des _____ du réseau, de sorte qu'ils soient identifiables de manière unique dans les communications.
4. Un _____ est un ensemble d'adresses IP défini au niveau du serveur _____ pour les ordinateurs du réseau qui en feront la demande.
5. Une adresse _____ est fixe tandis qu'une adresse _____ est variable, donc peut changer suivant une durée de temps allouée par le _____ DHCP.

EXERCICE
PRATIQUE

Exercice 4 : Proposer une solution aux problèmes ci-après :

On désire configurer un réseau de 100 machines

1. Faire l'inventaire du matériel réseau nécessaire
2. Proposer les adresses IP et masque de sous-réseau de chaque ordinateur
3. Proposer un mode d'adressage efficace des machines du réseau

5

MONTAGE D'UN RESEAU LOCAL

SOMMAIRE

- ▶ Inventaire des éléments du réseau
- ▶ Fabrication des câbles
- ▶ Interconnexion des périphériques
- ▶ Installation du protocole de communication
- ▶ Adressage du réseau

MOTS-CLES

- Connecteurs RJ45
- Sertissage
- Câble droit
- Câble croisé
- Réseau ad-hoc

Imprimante
192.168.0.9



Desktop1
192.168.0.2



Desktop2
192.168.0.3



Desktop3
192.168.0.4



Switch



Protocole de communication



Point d'accès
192.168.0.1



Desktop 4
192.168.0.5



Laptop
192.168.0.6



Tablet PC
192.168.0.8

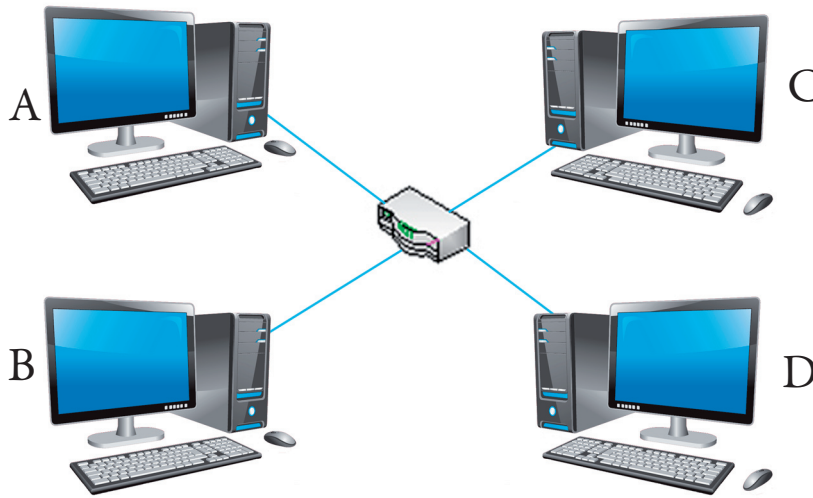


Smartphone
192.168.0.7

OBJECTIFS

- Lister les équipements nécessaires au montage d'un réseau
- Lister le matériel nécessaire à la fabrication d'un câble à paires torsadées
- Énoncer la procédure d'interconnexion des périphériques
- Énoncer la procédure d'installation du protocole TCP/IP
- Choisir les adresses IP et le type d'adressage du réseau
- Tester le fonctionnement du réseau

Soit le schéma réseau ci-dessous, réalisé dans le but de s'échanger des informations.



1. De quel type de réseau s'agit-il, filaire ou sans fil ?
2. Donner la topologie physique de ce réseau ?
3. Donner la topologie logique ?
4. Déterminer les équipements finaux et intermédiaires présents sur ce schéma
5. Quel type de câble est le plus approprié pour ce réseau ?
6. On suppose qu'on a choisit le câble de la question 5.
 - Combien de câbles faut-il ?
 - Que faut-il pour relier les câbles au ordinateur ?
7. Comment utiliser ces câbles pour interconnecter les différents périphériques ?
8. Proposer un protocole de communication et un moyen de le définir sur chaque machine du réseau.
9. On se propose d'adresser le réseau :
 - Combien d'adresses IP faut-il ?
 - Proposer les adresses IP à utiliser.

I. Introduction

L'interconnexion d'un réseau et sa configuration sont des étapes successives à mettre en œuvre pour la création d'un réseau fonctionnel. Mais avant, il est nécessaire de : connaître les éléments du réseau, fabriquer le câble pour connecter les ordinateurs, interconnecter les périphériques réseaux, installer le protocole de communication et adresser les ordinateurs pour qu'il y ait communication dans le réseau.

II. Inventaire des éléments du réseau

Pour la création d'un réseau filaire de quatre ordinateurs, il faut :

- *Les périphériques finaux* : quatre ordinateurs ayant chacun une carte réseau
- *Des périphériques intermédiaires* : un commutateur ou concentrateur pour relié les différents ordinateurs
- *Une topologie physique en étoile* : pour la disposition des périphériques réseaux
- *Une topologie logique Ethernet* : comme procédé d'émission sur le support de transmission
- *Du câble à paires torsadées* : support d'interconnexion utilisé dans les topologies en étoile. Il faut exactement quatre câbles à paires torsadées pour l'interconnexion

III. Fabrication des câbles à paires torsadées

1. Matériel utile à la fabrication du câble

Pour créer un câble à paires torsadées, il y a un certain nombre d'éléments à réunir :

- Une pince à sertir

- Une pince coupante
- 8 mètres de câble à paire torsadées à raison de 2 mètres par câble
- 8 connecteurs RJ45 à raison de deux par câble

La Figure 4.1 présente les éléments nécessaires pour sertir les câbles



Figure 5.1 : Matériel nécessaire au sertissage des câbles à paires torsadées

Le processus de fabrication du câble sera le même pour les autres.

2. Processus de fabrication d'un câble à paires torsadées

Muni des deux pinces, d'un câble de 2 mètres de long et de deux connecteurs RJ45, il faut :

a. Dénuder le câble

- Retirer la gaine du câble en utilisant la pince coupante, attention à ne pas endommager les fils. Certaines pinces à sertir proposent une partie permettant de dénuder le câble. Enlevez une partie plus grande que celle à entrer dans la prise RJ45 ;
- Une fois la gaine retirée, il en ressort alors la partie blindage (petite feuille d'aluminium). Utiliser la pince coupante pour l'enlever ;

- Enlever la partie protection en plastique transparent qui est au dessus, pour n'avoir que les paires torsadées ;
- Enfin Couper le petit fils en acier, ce dernier ne nous sert pas.

La Figure 4.2 présente ce qu'on obtient après cette étape.

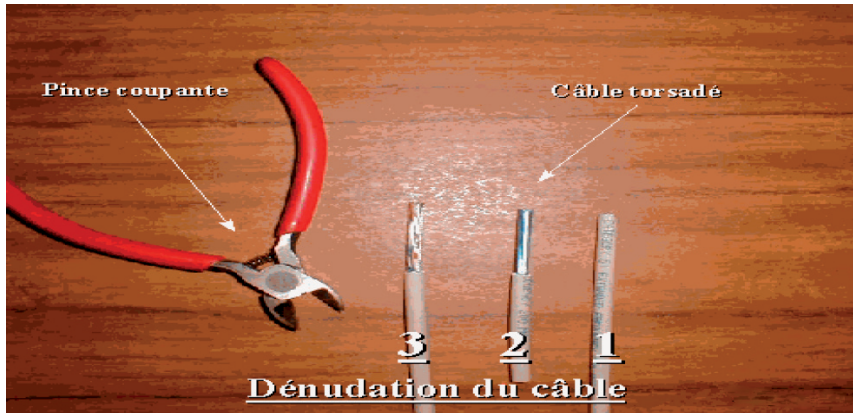


Figure 5.2 : Etapes de dénudage d'un câble à paires torsadées

b. Agencer les fils

Les fils vont par paire. 4 paires de fils, donc 8 fils. Les fils sont associés par couleur (voir Figure 4.3 ci-contre):

- Orange - blanc orange ;
- Bleue - blanc bleu ;
- Vert - blanc vert ;
- Marron - blanc marron.

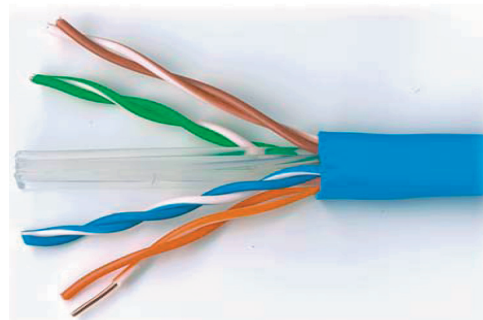


Figure 5.3 : Fils de câble regroupés par paires

L'agencement des fils sera différent selon qu'on crée un câble droit (pour relier un ordinateur à un périphérique réseau) ou un câble croisé (pour relier directement deux ordinateurs).

Agencement pour un câble droit

On peut utiliser le procédé mnémotechnique suivant :

Le vert coupe le vert

Le marron ne coupe pas

L'orange serre le bleu

La coupe ne coupe pas



Figure 5.4 : Agencement de fils pour câble droit

Ceci signifie qu'on place à gauche le fil vert-blanc (le vert coupé) puis le fil vert. Ensuite à droite on place le fil marron (qui ne coupe pas) et le fil blanc-marron. Puis on place les fils orange de part et d'autre des fils bleus qui sont au milieu (l'orange serre le bleu) en disposant les fils blanc-couleur à la suite des fils de couleur pleine (la coupe ne coupe pas).

Agencement pour un câble croisé

Ayant appliqué la méthode précédente pour un bout du câble, on utilise le procédé mnémotechnique suivant pour l'autre bout :

L'orange coupe l'orange

Le bleu ne coupe pas

Le vert serre le marron

La coupe ne coupe pas

Ceci signifie qu'on place à gauche le fil orange-blanc (l'orange coupe) puis le fil orange. Ensuite à droite on place le fil bleu (qui ne coupe pas) et le fil blanc-bleu. Puis on place les fils verts de part et d'autre des fils marron qui sont au milieu (l'orange serre le bleu) en disposant les fils blanc-couleur à la suite des fils de couleur pleine (la coupe ne coupe pas).

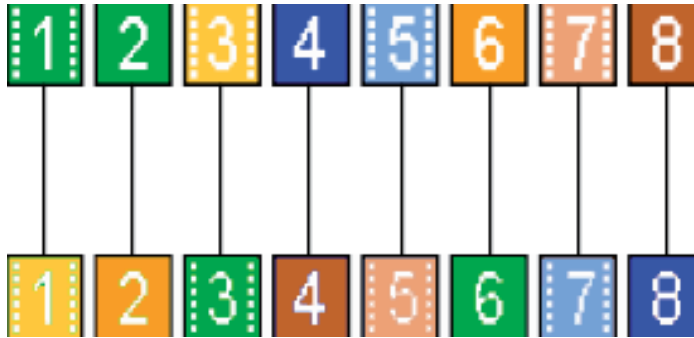


Figure 5.5 : Agencement de fils pour câble croisé

- Une fois l'ordre établi, coupez les fils afin de tous les mettre au même niveau, en gardant une taille supérieure à la longueur de la prise RJ45. En effet, une partie de la gaine du câble doit entrer dans la base de la prise RJ45 ;
- Faire entrer les fils, dans l'ordre préétabli, à l'intérieur de la prise RJ45. La prise RJ45 doit avoir la "tête" en bas (petite patte vers le bas) ;
- Vérifiez l'ordre à travers le plastique transparent de la prise RJ45.

Il reste alors à sertir le tout.

c. Sertissage de la prise RJ45

Lors du sertissage, les conducteurs de la prise RJ45 vont percer la gaine plastique de chaque fils pour entrer en contact avec le conducteur. La base de la prise va s'écraser sur la gaine de protection du câble. Tirer sur la prise pour vérifier que tout tient bien. Répéter la même opération sur les trois autres câbles droits.

IV. L'interconnexion des périphériques du réseau

L'interconnexion des périphériques du réseau peut se faire de deux manières :

1. Interconnexion filaire

- Connecter le premier bout de chacun des quatre câbles sur le commutateur ;
- Connecter le second bout de chacun des quatre câbles sur la sortie de la carte réseau de chaque ordinateur ;
- L'interconnexion a réussi lorsque deux voyants (vert et orange) sont allumés.

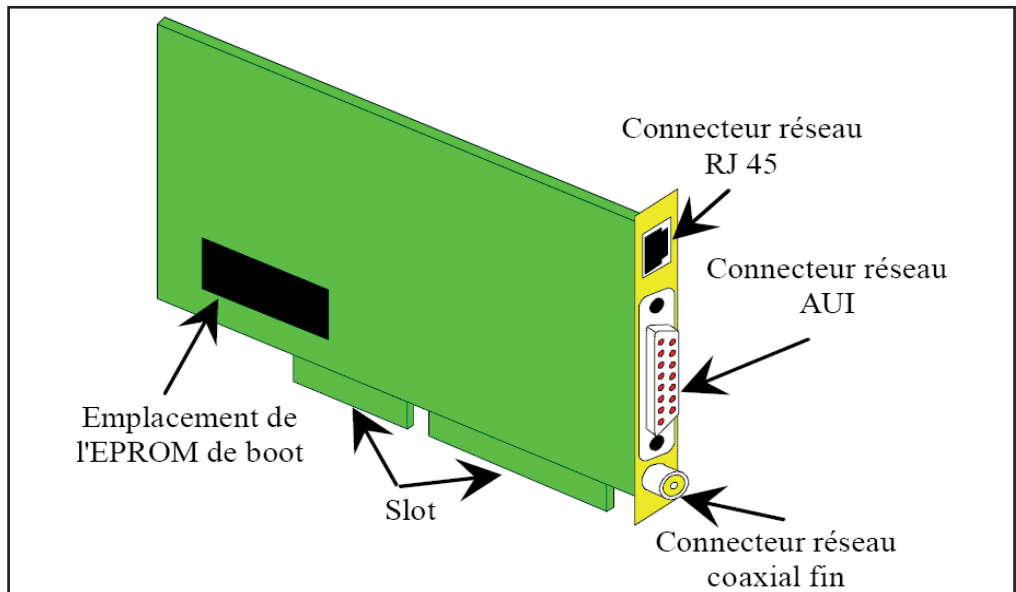


Figure 5.6 : Connectique d'une carte réseau

2. Interconnexion sans fil

La connexion des quatre ordinateurs sans câble peut se faire de deux manières :

- A partir d'un équipement central sans fil. Exemple : un point d'accès (routeur sans fil ou un commutateur sans fil) ;

- Sans équipement central : les équipements sont liés les uns aux autres sans équipement central : on parle de réseau ad-hoc.

V. Installation du protocole de communication

L'installation du protocole de communication varie d'un système d'exploitation à l'autre, selon le système qu'on utilise.

- Ouvrir dans les paramètres réseaux du système d'exploitation utilisé ;
- Ouvrir les propriétés de la connexion réseau en cours ;
- Choisir le protocole TCP/IP version 4 et l'installer.

VI. Adressage du réseau

L'adressage statique consistera à attribuer Des adresses fixes aux ordinateurs. Les quatre adresses sont choisies dans l'une des trois plages d'adresses privées, ici de 10.0.0.0 à 10.255.255.255

Tableau 5.1 : Plan d'adressage du réseau

Ordinateurs	Adresses IP	Masque de sous réseau
Ordinateur A	10.0.0.1	255.0.0.0
Ordinateur B	10.0.0.2	255.0.0.0
Ordinateur C	10.0.0.3	255.0.0.0
Ordinateur D	10.0.0.4	255.0.0.0

- Aller dans les paramètres réseaux;
- Ouvrir les propriétés de la connexion réseau en cours
- Entrer l'adresse IP et le masque de sous-réseaux de chacun des ordinateurs A, B, C, D ;
- Tester le réseau.

Le plan du réseau est présentée dans la Figure 4.7 ci-après.

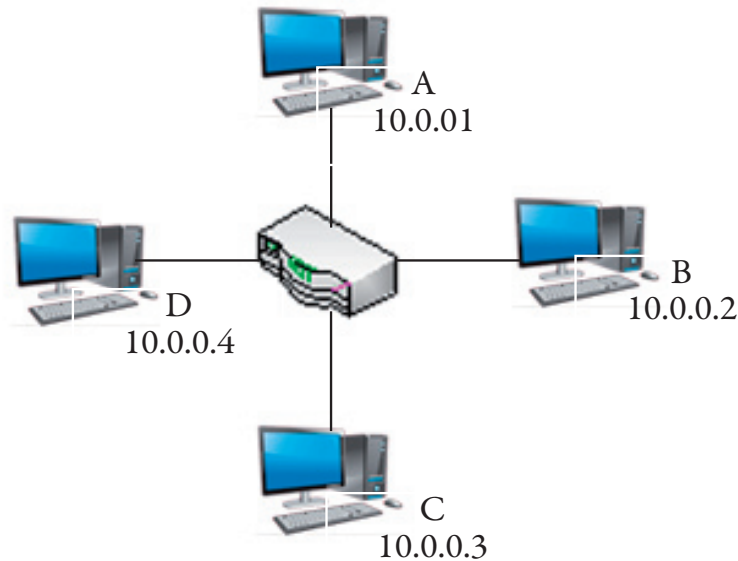
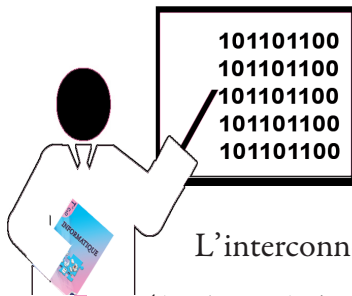


Figure 5.7 : Plan d'adressage du réseau



RESUME

L'interconnexion d'un réseau et sa configuration sont des étapes successives à mettre en œuvre pour la création d'un réseau fonctionnel. Mais avant, il est nécessaire de :

- Connaître les éléments du réseau (équipements, topologie, support de transmission) ;
- Fabriquer le câble pour connecter les ordinateurs ;
- Interconnecter les équipements réseaux ;
- Installer le protocole de communication et d'adresser les ordinateurs pour qu'il ait communication dans le réseau.

La connaissance de la topologie du réseau est utile pour faire l'inventaire des périphériques finaux et intermédiaires. La fabrication du câble dépend la nature des périphériques à connecter. Ainsi, pour connecter deux périphériques intermédiaires ou deux périphériques finaux, on utilise un câble droit. Dans le cas contraire, c'est un câble croisé pour connecter un périphérique final et un périphérique intermédiaire.

EXERCICE
THEORIQUE**Exercice 1 : Répondre aux questions suivantes :**

1. Quelle carte d'extension de l'ordinateur permet de gérer les communications dans un réseau ?
2. Quel type de câble est utilisé dans les topologies en étoile ?
3. Quel sont les couleurs du câble à paires torsadées ?
4. Citer le matériel nécessaire à la fabrication d'un câble à paires torsadées.
5. Donner le nom des deux types de câble à paires torsadées que l'on peut fabriquer.
6. Présenter le processus de fabrication d'un câble à paires torsadées.
7. Donner les deux manières distinctes d'interconnexion des périphériques réseaux.
8. Donner deux manières distinctes d'interconnexion sans fil.
9. Donner la démarche à suivre pour installer la pile de protocole TCP/IP sur un ordinateur.
10. Donner la démarche à suivre pour attribuer les adresses IP aux ordinateurs d'un réseau

EXERCICE
THEORIQUE**Exercice 2 : Répondre par vrai ou faux :**

1. Dénuder un câble consiste à retirer la gaine, la feuille métallique, la partie de protection et le fils en acier
2. L'agencement des fils sera différent selon qu'on veut créer un câble droit ou un câble croisé
3. Un câble droit permet d'interconnecter un périphérique final et un périphérique intermédiaire
4. L'installation du protocole de communication n'est pas obligatoire pour faire fonctionner le réseau
5. Seule l'adresse IP suffit pour l'adressage d'un ordinateur dans un réseau.

EXERCICE
THEORIQUE**Exercice 3 : Complétez les phrases à trous par la liste des mots suivants : Périphériques, système d'exploitation, fils, Configuration, Windows, ordinateurs, RJ45, câble, réseau, privées, protocole, installation, éléments, statique**

1. L'interconnexion d'un réseau et sa _____ sont des étapes successives à mettre en œuvre pour la création d'un _____ fonctionnel.

2. Mais avant, il est nécessaire de : Connaître les _____ du réseau, fabriquer le _____ pour connecter les ordinateurs, interconnecter les _____ réseaux, installer le _____ de communication et adresser les _____ pour qu'il y ait communication dans le réseau.
3. Lors du sertissage, les conducteurs de la prise _____ vont percer la gaine plastique de chaque _____ pour entrer en contact avec le conducteur.
4. L'_____ du protocole de communication varie d'un _____ à l'autre, selon que vous utilisez un système _____ ou Linux ou Macintosh.
5. L'adressage _____ consiste à attribuer les adresses fixes aux ordinateurs. Les adresses sont choisies dans l'une des trois plages d'adresses _____.



EXERCICE 4 : Mise en œuvre d'un réseau et navigation sur Internet

Dans le but de partager les ressources du réseau, de faciliter la communication et de garantir un enseignement de qualité, deux réseaux ont été mis sur pied au Lycée Bilingue du Cameroun, notamment un au niveau du bloc administratif (voir figure 1) et un autre dans la salle machine du second cycle (voir figure 2). Le bloc administratif comporte deux compartiments : le rez-de-chaussée, la salle des professeurs et le premier étage (Bureau des Censeurs et du Proviseur). Dans la salle des professeurs, les enseignants utilisent leurs ordinateurs portables pour travailler et sont connectés au même réseau que les censeurs et le Proviseur.

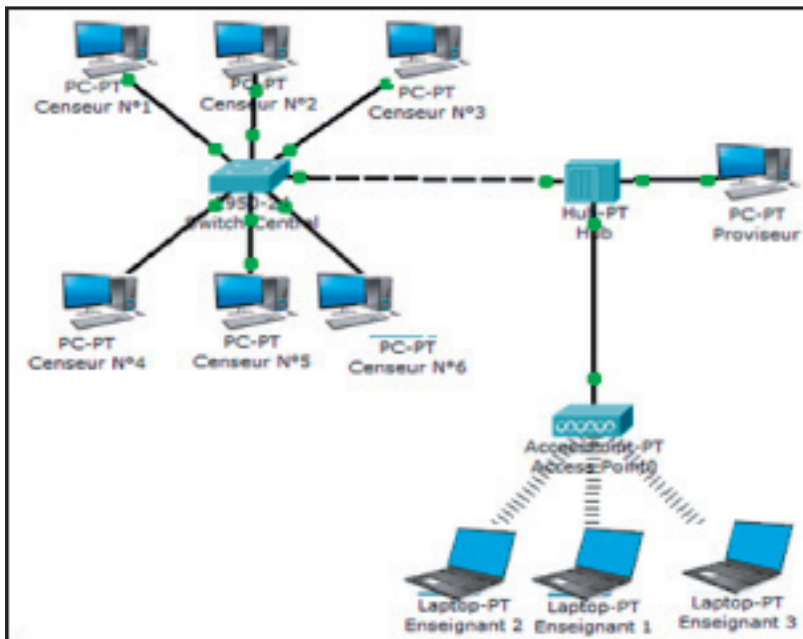


Figure 1 : Réseau du bloc administratif du Lycée Bilingue du Cameroun

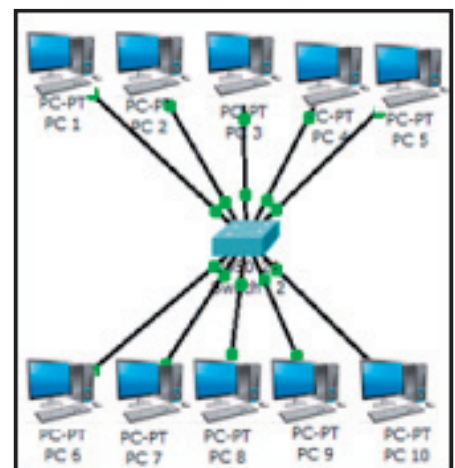


Figure 2 : Réseau de la salle machine

1. Examinez les deux figures 1 et 2 et répondez aux questions qui suivent :
 - a) Citer les ressources partageables dans un pareil réseau ?
 - b) Donner la topologie physique du réseau du bloc administratif et du réseau de la salle machine ?
 - c) Dire pour chaque réseau le (ou les) type (s) de réseaux conçu(s).
 - d) Lister les équipements d'interconnexion utilisés dans les deux réseaux
 - e) Lister les supports d'interconnexion utilisés dans les deux réseaux
 - f) Quel est le nombre d'adresses IP utilisés dans le réseau du bloc administratif ?
 - g) Le Proviseur n'a pas assez de moyens pour mettre une imprimante dans chaque bureau de censeur. Proposez une solution simple et peu coûteuse permettant à chaque censeur de faire des impressions.
 - h) Le proviseur veut envoyer un message au Censeur N°1, comment va circuler ce message sur le réseau ?
 - i) L'enseignant 2 a besoin du Censeur N°5 la liste des élèves de la classe de Terminale pour remplir les notes de ses élèves. Va-t-il se déplacer dans le bureau de ce censeur ? Comment gagner en temps à l'aide du réseau ?
2. Comment connecter le réseau du bloc administratif et de la salle machine ? quel équipement est indispensable ?
3. Le proviseur aimerait avoir une connexion internet dans son bureau.
 - a) Quel périphérique faut-il connecter à son ordinateur pour avoir la connexion internet ?
 - b) Que doit-il faire pour permettre aux enseignants et aux censeurs d'avoir également la connexion internet ?
4. Le proviseur s'est rendu compte que l'une des causes de l'échec scolaire des élèves est du au site internet dénommé : www.facebook.com lieu d'exhibition de photos et vidéos ainsi que de discussion instantanée.
 - a) Quel logiciel doit-elle installer pour ouvrir ce site internet ?
 - b) Le proviseur crée un compte facebook et charge sa photo de profil. Quel est le lieu où est stockée cette photo ? comment appelle-t-on cette opération ?
 - c) Le proviseur n'a pas jusqu' alors la philosophie du site internet facebook. Comment peut-elle avoir des informations sur l'utilité de ce site afin de mieux comprendre son fonctionnement et l'intérêt qu'il suscite chez les élèves de son lycée ? Proposer une solution.

PROJET DE CONSOLIDATION 1

L'objectif de ce projet est de mettre en place un réseau local comprenant deux parties :

- La première partie est composée de quatre desktops reliés à un switch ;
- La seconde partie est composée d'un certain nombre de périphériques mobiles reliés à un routeur sans fil par Wifi.

Les deux parties sont reliées entre elles par un câble allant du routeur sans fil au switch. On utilisera une topologie Ethernet avec connexion internet. On suppose qu'entre chaque ordinateur et le switch il y a une distance de cinq mètres. Entre le switch et le routeur sans fil il y a une distance de dix mètres.



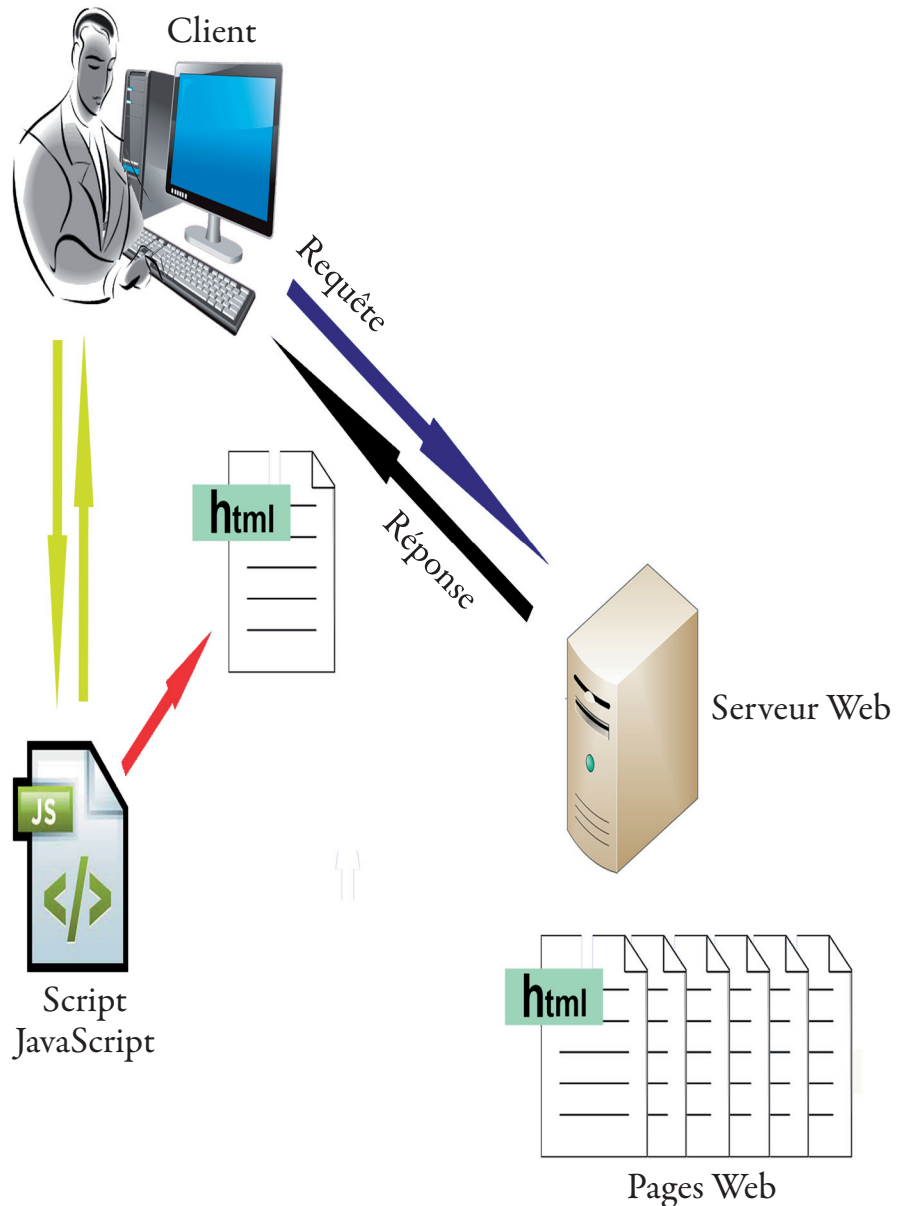
- Lister le matériel nécessaire pour la mise en place du réseau ;
- Proposer un adressage pour les équipements du réseau ;
- Quel équipement doit recevoir la connexion Internet en premier ;
- Sertir les câbles ;
- Connecter les équipements ;
- Réaliser l'adressage des équipements ;

Module 2

PROGRAMMATION INTERACTIVE DE PAGES Web

En classe de première, les notions de langage HTML ont été abordées pour la création de pages Web statiques. Le savoir-faire, qui était limité en la production d'un document figé, un peu comme une photo, est bien loin de pouvoir créer des pages semblables à celles présentes sur le Web. Une première solution sera abordée dans ce module à travers le langage JavaScript.

Après que le poste client ait reçu le code HTML de la page Web, son navigateur va exécuter les instructions pas à pas. Grâce à JavaScript, une conversation va pouvoir s'installer entre l'utilisateur et son poste (demande d'informations, réponse à un clic, réponse au survol de la souris, etc). Et l'interactivité sera.



6

INTRODUCTION A JAVASCRIPT

SOMMAIRE

- ▶ Rappels sur HTML
- ▶ Historique de JavaScript
- ▶ Importance de JavaScript
- ▶ Limites de JavaScript
- ▶ Sécurité de JavaScript
- ▶ Notion d'objet

MOTS-CLES

- Navigateur
- Script,
- Applet,
- Servlet
- Objet



OBJECTIFS

- Définir : Page Web, Script, Applet, Servlet, Objet
- Utiliser les balises usuelles du HTML
- Donner l'importance et les limites de JavaScript
- Critiquer la sécurité d'un script JavaScript
- Donner les objets gérés par JavaScript

Soit la page Web d'accueil du site des élèves du Lycée Bilingue du Cameroun présentée ci-dessous :

Module 2 : Programmation interactive de pages Web

élèveLBC.com
Site officiel des élèves du Lycée Bilingue du Cameroun

Section Francophone

- [Sixième](#)
- [Cinquième](#)
- [Quatrième](#)
- [Troisième](#)
- [Seconde](#)
- [Première](#)
- [Terminale](#)

Section Anglophone

- [Form I](#)
- [Form II](#)

Login

Mot de passe

Valider

Bienvenue sur le site des élèves du Lycée Bilingue du Cameroun

Présentation générale

Le lycée Bilingue du Cameroun est un établissement scolaire d'enseignement général secondaire bilingue. Il est composé :

- D'une section francophone : comprenant des classes de sixième ;
- D'une section anglophone comprenant des classes de Form I et de Form II.

[Accéder aux QCM](#)

1. Lister les différentes balises qui permettent de réaliser la page Web ci-dessus ;
2. Lister les éléments de style qui ont été appliqués ;
3. Que se passe-t-il si on clique sur le texte "Bienvenue sur le site des élèves du Lycée Bilingue du Cameroun" ;
4. Que se passe-t-il si on clique sur le texte "Sixième" ;
5. Peut-on donner une valeur à la page Web ?
6. La page Web peut-elle nous afficher un texte autre que ceux qui y sont déjà ?
7. Qu'appelle-t-on interactivité ?
8. Cette page Web est-elle interactive ?
9. Quelles sont les limites de HTML ?

I. Historique de JavaScript

JavaScript est un langage de programmation de script qui s'exécute coté client, c'est-à-dire dans le navigateur Web. JavaScript permet de réaliser des scripts qui vont s'insérer dans un document HTML.

JavaScript naît en 1995. Créé par Netscape, il s'interface avec la nouvelle version du navigateur Netscape Navigator. Il est alors appelé «*Livescript*». C'est suite à un accord commercial entre Netscape et Sun (le créateur de Java) que le langage sera baptisé JavaScript.

En 1996, Microsoft crée son propre langage de script appelé «*Jscript*» pris en charge dans Internet Explorer. Avec la guerre entre les navigateurs Internet Explorer et Netscape, on ne s'y retrouve plus en langage de script. Deux normes (ECMA 262 en 1997 et ECMA ISO en 1998) consacrent définitivement JavaScript comme langage de script coté client.

Aujourd'hui on en est à la version 1.6 de JavaScript qui est assez bien prise en charge par les navigateurs récents.

JavaScript ne doit pas être confondu avec Java qui est un langage de programmation plus évolué et plus complexe et qui permet de réaliser des programmes autonomes. Ainsi, un script JavaScript ne doit pas être confondu avec :

- Un applet : petit programme écrit en Java qui s'exécute sur le navigateur client ;
- Un servlet : petit programme écrit en Java qui s'exécute sur le serveur.

Toutefois, un script JavaScript peut accéder à un applet ou à un servlet, mais ceci dépasse largement le cadre de notre cours.

II. Importance de JavaScript

En classe de première, les pages Web que nous avons conçues étaient statiques, c'est-à-dire que leur contenu n'était pas modifiable, et ne réagissaient pas aux actions de l'utilisateur sauf pour ce qui est des clics sur les liens hypertextes. JavaScript va ajouter de l'interactivité. On va pouvoir :

- Répondre à des actions de l'utilisateur (clic sur un lien ou sur un bouton, survol d'une image avec la souris, etc) ;
- Récupérer des valeurs entrées par l'utilisateur pour exécuter des commandes (exemple de la validation d'un formulaire, calculs simples sur les valeurs entrées, gestion des dates et des heures) ;
- Créer et de lire des cookies ;
- Gérer la navigation (menu dynamique) ;
- Réaliser des animations graphiques ;
- Lire et écrire des éléments HTML.

III. Limites de JavaScript

JavaScript reste limité pour les raisons ci-après :

- JavaScript est très dépendant du navigateur (l'exécution d'un script JavaScript peut donner des résultats différents d'un navigateur à un autre) et un certain nombre (comme Lynx ou Internet Explorer) le supportent mal ;
- JavaScript ne permet pas d'écrire ou de lire le disque dur. Il n'est donc pas possible de créer des pages Web dynamiques (pages créées en fonction des besoins de l'utilisateur ou des entrées d'une base de données), de concevoir un forum de discussion, un script de sondage ou de vote. Pour cela, on va utiliser des langages de script comme PHP, CGI ou ASP ;
- JavaScript n'échange pas avec d'autres machines connectées. Impossible donc de créer une page de chat (discussion en direct).

IV. Sécurité de JavaScript

Le problème de la sécurité des langages de script tient du fait qu'ils sont transparents pour le visiteur du site. En effet, lorsqu'un internaute tape dans la barre d'adresse de son navigateur une adresse de site, son poste envoie une requête au serveur qui héberge le site. Le serveur lui renvoie ensuite un document HTML. Lorsque ce document arrive, l'internaute ne sait pas qu'il comporte des scripts, et ceux-ci s'exécutent généralement sans qu'il en ait connaissance.

JavaScript a été conçu pour limiter (ou presque) les risques pour le visiteur. En ce sens, on peut dire qu'exécuter JavaScript sur son navigateur ne comporte aucun risque pour les données ou le matériel. En effet, JavaScript ne permet pas :

- D'écrire ou de lire un fichier sur le disque dur du visiteur ou sur le serveur, sauf dans la zone réservée aux cookies. Ainsi, vous ne risquez pas de voir des données confidentielles récupérées à votre insu ;
- De lister les logiciels utilisés ou les pages Web visitées ;
- Détruire des fichiers indispensables ou d'introduire un virus sur le poste d'un visiteur ;
- De récupérer l'adresse IP d'un visiteur.

JavaScript est presque sans risque. Il est possible de coder un script qui ouvre des pop-up jusqu'à épuiser complètement les ressources système. Dans ce cas, la seule solution est de rebooter votre ordinateur. On peut aussi imaginer un script qui tourne en tâche de fond d'une page Web, occupant ainsi des ressources et réduisant les performances de votre poste.

Les performances de JavaScript en termes de sécurité pour les visiteurs font que JavaScript est accepté par (presque) tous. Le prix de cette sécurité se traduit par des limites techniques que nous avons énoncées.

V. Notion d'objet

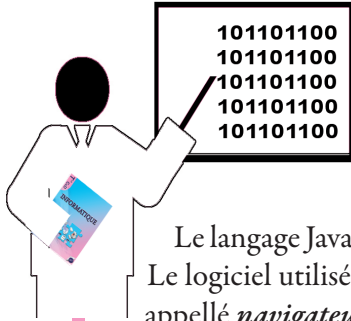
Un objet est la prise en compte de tout élément matériel ou immatériel utilisé par le programme. C'est un contenant qui permet de stocker des valeurs utilisées par le programme (opérandes) ou des résultats (exemples : un objet « somme », un objet « produit », ...).

Les objets pouvant contenir des valeurs diverses, elles sont appelées variables. Ces variables peuvent être regroupées en ensembles appelés types. (Exemple : les variables pouvant contenir des nombres font partie du type « nombre »).

JavaScript peut également accéder à des objets contenus dans la page Web. Ceux-ci sont appelés variables du navigateur. On distingue :

- L'objet Windows : il représente une fenêtre du navigateur ;
- L'objet Document : il représente une page Web affichée dans une fenêtre du navigateur. Il contient les objets :
 - Image ;
 - Anchor (lien) ;
 - Form (formulaire) : qui peut également contenir des éléments (text, button, etc.)
 - Etc.

L'accès aux objets se fait par une notation par points. Ainsi, pour accéder à une zone de texte d'un formulaire, on utilise la notation suivante : « windows.document.form.text »



RESUME

Le langage JavaScript permet d'écrire des scripts qui vont s'intégrer à un document HTML. Le logiciel utilisé pour interpréter et afficher le résultat des programmes écrits en HTML est appelé *navigateur*. Ce dernier est appelé client Web parce que c'est lui qui initie les requêtes de page Web à destination du serveur. A la réception du code de la page HTML contenant du JavaScript, le navigateur exécute les instructions des scripts. On dit que le code JavaScript s'exécute *côté client*.

Le JavaScript est important car il permet de répondre à des actions de l'utilisateur, de récupérer des valeurs entrées par l'utilisateur pour exécuter des commandes (exemple de la validation d'un formulaire etc.) Toutefois, JavaScript dépend des navigateurs et ne permet pas d'écrire sur le disque dur.

JavaScript est limité. En effet, il ne permet pas de mettre en place un certain nombre d'outils qui s'imposent dans les technologies Web (module de recherche, forums de discussion, etc.) Toutefois, cet aspect limité garantit la sécurité de JavaScript.

JavaScript est orienté-objet, c'est-à-dire qu'il manipule un certain nombre d'éléments, notamment du navigateur, en réponse à des événements.

EXERCICE
THEORIQUE**Exercice 1 : Répondre aux questions suivantes**

1. Définir les termes : site web, script, applet, Servlet et objet
2. Différencier le langage JavaScript du langage Java
3. Pourquoi dit-on que les scripts JavaScript apportent de l'interactivité aux pages Web ?
4. Quelles sont les limites du langage JavaScript? Donner quelques alternatives
5. Pourquoi dit-on que l'exécution des scripts JavaScript est sans risque ?
6. Citer les objets du langage JavaScript.
7. Pourquoi dit-on que les scripts en langage JavaScript s'exécute du côté client ?
8. Quel nom donne-t-on au logiciel permettant d'exécuter les programmes HTML et en JavaScript ?

EXERCICE
THEORIQUE**Exercice 2 : Répondre par vrai ou faux :**

1. Le langage JavaScript a été créé en 1996 par la société Netscape.
2. Les navigateurs lynx et internet explorer supportent bien le JavaScript.
3. L'exécution des scripts JavaScript est transparente aux utilisateurs de sites Web.
4. Exécuter un script JavaScript sur son navigateur ne comporte aucun risque pour le matériel et le réseau.
5. Les objets sont matérialisés dans des programmes par des variables de plusieurs types.

EXERCICE
THEORIQUE**Exercice 3 : Compléter les phrases à trous par les mots de la liste suivante :**

Internaute, Scripts, statiques, utilisateur, Navigateur, script, adresse, interactivité, JavaScript, site Web, HTML, client, serveur

1. Le langage de programmation de _____ appelé _____ s'exécute coté _____, c'est-à-dire dans le _____ Web.
2. JavaScript permet de réaliser des _____ qui vont s'insérer dans un document _____.
3. Le contenu des pages Web _____ ne sont pas modifiables, et ne réagissaient pas aux actions de l'_____ sauf pour ce qui est des clics sur les liens hypertextes. JavaScript va ajouter de l'_____.
4. Lorsqu'un internaute tape dans la barre d'_____ de son navigateur une adresse de _____

_____, son poste envoie une requête au serveur qui héberge le site. Le _____ lui renvoie ensuite un document HTML.

- Lorsque ce document arrive, l'_____ ne sait pas qu'il comporte des scripts, et ceux-ci s'exécutent généralement sans qu'il en ait connaissance.



Exercice 4 : Compléter les phrases à trous suivantes :

- Un _____ est la prise en compte de tout élément _____ ou immatériel utilisé par le _____.
- C'est un _____ qui permet de stocker des _____ utilisées par le programme ou des résultats.
- JavaScript peut également accéder à des _____ contenus dans la page _____.
- L'objet _____ représente une fenêtre du _____.
- L'accès aux objets se fait par une notation par _____.



Exercice 5 : Faire correspondre les éléments de la colonne de gauche aux éléments de la colonne de droite :

Langage
Document
Navigateur
Internet
Applet
Objet
Anchor
Page
Côté
Barre

d'adresse
client
statique
lien
Windows
java
explorer
Web
HTML
JavaScript

7

LE LANGAGE JAVASCRIPT

SOMMAIRE

- ▶ Déclaration des variables
- ▶ Utilisation des structures de contrôle
- ▶ Utilisation des tableaux
- ▶ Éléments d'interaction

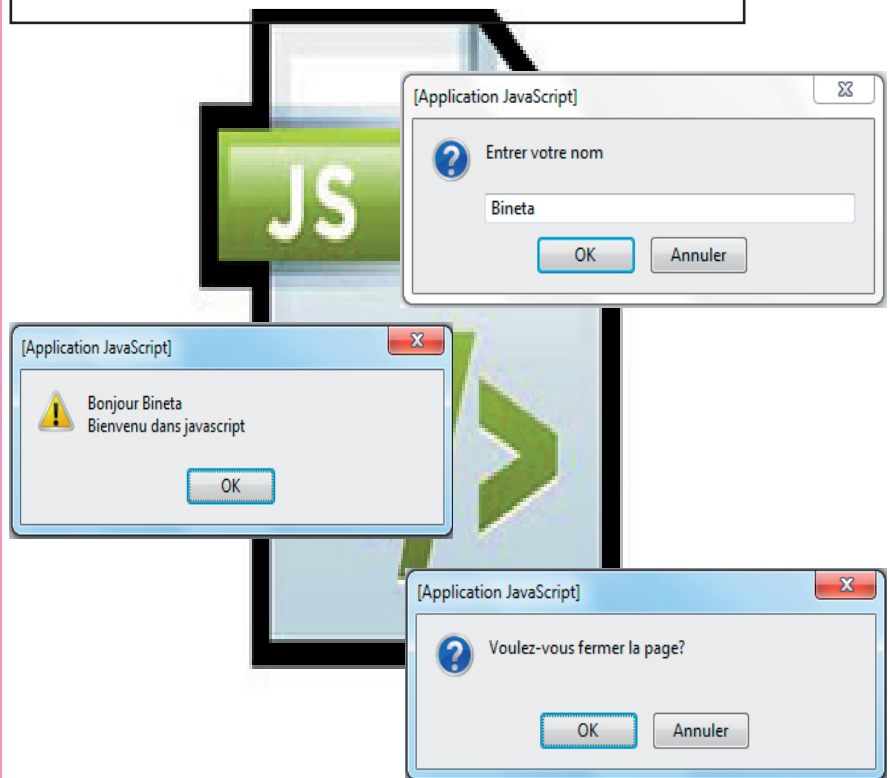
MOTS-CLES

- Variable
- Affectation
- Tableaux
- Fonctions
- Procédures

PRE - REQUIS

- Notion d'algorithme
- Instructions algorithmiques

```
var nom = new String();  
nom = prompt ("Entrer votre nom");  
alert ("Bonjour "+nom+"\nBienvenu dans javascript");  
confirm ("Voulez-vous fermer la page?");
```



OBJECTIFS

- Déclarer et utiliser des variables
- Utiliser des structures de contrôle
- Utiliser des tableaux
- Utiliser des fonctions et des procédures
- Utiliser des éléments d'interaction

Au cours d'un entretien, un conseiller d'orientation interroge l'élève Mamadou de Terminale scientifique afin de déterminer les choix les meilleurs qui s'offrent à lui suivant sa formation et ses aptitudes.

La conversation est reportée ci-après :

- *Le Conseiller d'orientation* : Bonjour Mamadou.

- *Mamadou* : Bonjour Madame.

- *Le Conseiller d'orientation* : Bientôt tu auras ton baccalauréat. Que penses-tu faire par la suite?

- *Mamadou* : Madame je voudrais être médecin ou pharmacien.

- *Le Conseiller d'orientation* : As-tu une bonne moyenne en Sciences de la Vie et de la Terre ?

- *Mamadou* : Oui Madame.

- *Le Conseiller d'orientation* : As-tu de bonnes notes dans le premier groupe ?

- *Mamadou* : Juste la moyenne.

- *Le Conseiller d'orientation* : Dans quelle école penses-tu faire tes études ?

- *Mamadou* : Je ne sais pas encore Madame.

- *Le Conseiller d'orientation* : Penses-tu à la Faculté de Médecine et des Sciences Biomédicales?

- *Mamadou* : Non Madame. Je voudrais aller hors de Yaoundé ;

- *Le Conseiller d'orientation* : Il y a également l'Université des montagnes. Qu'en penses-tu ?

- *Mamadou* : Oui, peut-être.

- *Le Conseiller d'orientation* : Tu pourrais aussi faire pharmacie à l'Université de Douala ;

- *Mamadou* : Non, je voudrais plus être médecin ;

- *Le Conseiller d'orientation* : Bien, je te propose alors la Faculté de Médecine de Yaoundé, d'Ebolowa, de l'Université des Montagnes ou d'aller à l'étranger. Dans tous les cas, soigne d'abord tes notes dans ton premier groupe, car elles seront déterminantes dans l'évaluation de ton dossier pour le concours.

Sur la base de cette conversation, proposer des réponses aux questions suivantes :

- Qu'appelle-t-on *interactivité* ?
- Ressortir dans le texte une structure alternative ? Quelle est la condition dans cette structure ?
- Ressortir une structure itérative ? Quelle est la condition d'entrée ou de sortie de cette structure ?
- Donner l'algogigramme de la conversation.

I. Déclaration des variables

Une variable est représentée dans le programme par un nom ou identificateur. Le nom d'une variable est un ensemble de caractères alphanumériques. (Exemple : Mavaleur1). Les noms de variables doivent cependant respecter les règles suivantes :

- Commencer par une lettre ;
- Ne pas contenir d'espace ;

Exemple : « 2variable » et « Ma variable » ne sont pas des noms de variables valides.

Pour qu'une variable soit utilisée par Javascript, il faut la déclarer, c'est-à-dire indiquer au navigateur de réserver un emplacement mémoire pour stocker les différentes valeurs de la variable. Les déclarations doivent être suivies du type de la variable (l'ensemble des valeurs que peut prendre la variable). JavaScript manipule principalement cinq types de variable :

- **Number** : ce sont des variables qui contiennent des nombres (entiers ou décimaux) ;
- **String** : ce sont des variables qui contiennent des chaînes de caractères (texte) ;
- **Boolean** : ce sont des variables qui ne prennent que deux valeurs (true ou false) ;
- **Array** : ce sont des variables qui stockent des tableaux.
- **Date** : ce sont des variables qui stockent des dates dans le format « AAAA, MM, JJ »
- Lorsqu'une variable ne contient aucun élément, sa valeur est « Null ».

1. Déclaration sans affectation

Avec une déclaration sans affectation, on déclare la variable sans lui donner une valeur initiale ; sa valeur est fonction de son type.



DEFINITION

Variable :

Une variable est un objet créée par le programmeur pour matérialiser tout élément utilisé dans le script.



REMARQUE

Syntaxe JavaScript :

JavaScript est très sensible à la casse. De ce fait, les noms « mavariable » et « Mavariable » désignent deux variables différentes, de même que les noms « élève » et « eleve ». Par conséquent, il est recommandé d'écrire les noms de variables en minuscule et sans accent.



REMARQUE

Mots-clés :

Les mots-clés du langage doivent impérativement être écrits en respectant la casse. Ainsi, « number » ne sera pas reconnu comme type de variable, mais plutôt « Number ».



REMARQUE

Valeur par défaut d'une variable :

La valeur par défaut d'un texte est une chaîne de caractères vide (" "), celle d'un nombre est 0, celle d'un booléen est false, celle d'un tableau est Null, celle d'une date est la date système.



REMARQUE

Evolution du type d'une variable :

Le type d'une variable change en fonction des valeurs qui y sont affectées. Ainsi une variable de type String évoluera en Number si on affecte la valeur «1» à cette variable.

La syntaxe d'une déclaration sans affectation est la suivante :

```
Var Mvariable = new typeVariable ( ) ;
```

Exemple 7.1 : Déclarations sans affectation en JavaScript

```
Var nombre1 = new Number( ) ;
Var tableau2 = new Array( ) ;
Var condition = new Boolean( ) ;
```

- Le mot-clé « Var » permet de spécifier qu'on commence la déclaration d'une variable. Ce mot-clé est facultatif ;
- Toutes les instructions JavaScript se terminent par un point-virgule.

2. Déclaration avec affectation

Avec ce type de déclaration, on donne à la variable déclarée une valeur initiale : on parle d'*assignement*. On peut alors faire la déclaration en précisant le type de variable ou en laissant au navigateur le soin de reconnaître le type de variable en fonction de la valeur d'assignement.

Exemple 7.2 : Déclarations avec affectation en JavaScript

```
Var nombre1 = new Number(3) ;
Var tableau2 = new Array(12,25,89) ;
Var condition = new Boolean(false) ;
Var chainel = new String("Bonjour" ) ;
nombre2 = 4 ;
texte1 = "Bonjour" ;
texte2 = "programmation javascript" ;
```

- «nombre1» a la valeur 3 ;
- «tableau2» contient les chiffres 12, 25 et 89 ;
- «condition» a pour valeur false ;
- «nombre2» a pour valeur 4 ;
- Le type de variable de «texte1» et «texte2» est String. Pour indiquer une chaîne de caractères, on l'entoure de guillemets ou d'apostrophes.

3. Utilisation des variables

Le nom d'une variable désigne à la fois un contenant (l'espace mémoire réservé à la variable) et un contenu (la valeur contenue dans ledit espace mémoire). Pour se le représenter, on peut considérer les termes « seau » et « seau d'eau » qui désignent respectivement le seau en tant que contenant, et le volume d'eau contenu dans le seau. De ce fait, le nom d'une variable peut être utilisé pour indiquer un emplacement mémoire mais aussi une valeur.

Exemple 7.3 : Utilisation des variables en JavaScript

```
Var nombre1 = new Number(17) ;  
nombre1 = (nombre1-8)*3 ;  
nombre1 = new Number( ) ;  
Var nombre2 = 6 ;  
nombre2 = 6/2 ;
```

- On déclare la variable «nombre1» et on lui assigne la valeur 17 ;
- On va chercher la valeur de «nombre1» et on réalise l'opération $(17-8)*3$ et on obtient 27. On affecte alors la valeur 27 à «nombre1» ;
- On déclare à nouveau «nombre1» et on lui affecte la variable 0 ;
- On déclare la variable «nombre2» et on lui affecte la valeur 6. Le navigateur donne à «nombre2» le type Number ;
- On affecte à «nombre2» la valeur 3 (résultat de $6/2$).

4. Opérations sur les variables

a. Opérations arithmétiques

On retrouve les opérations mathématiques de base. L'opération modulo permet de calculer le reste de division. Par exemple, 7 modulo 2 est le reste de division de 7 par 2, il vaut 1 (car $7 = 2 \times 3 + 1$).

Écrasement de valeur :

Lorsqu'on affecte une valeur à une variable qui en avait déjà une, la première valeur est remplacée par la seconde : on dit qu'elle est écrasée.



REMARQUE

Le tableau suivant présente les opérations arithmétiques de base :

Tableau 7.1 : Opération arithmétiques en JavaScript

Opération	Nom	Exemple	Observation
+	Addition	$x=7+2$	x vaut 9
-	Soustraction	$x=7-2$	x vaut 5
*	Multiplication	$x=7*2$	x vaut 14
/	Division	$x=7/2$	x vaut 3,5
^	Puissance	$x=2^3$	x vaut 8
%	Modulo	$x=7\%2$	x vaut 1
++	Incrémentation de 1	$x++$	x vaut 2
--	Décrémentation de 1	$x--$	x vaut 1

JavaScript dispose aussi de fonctions mathématiques. On peut citer :

- `Math.abs(x)`
- `Math.acos(x)`
- `Math.asin(x)`
- `Math.atan(x)`
- `Math.cos(x)`
- `Math.exp(x)`
- `Math.log(x)`
- `Math.max(x,y)`
- `Math.min(x,y)`
- `Math.pow(x,y)`
- `Math.random()`
- `Math.sin(x)`
- `Math.sqrt(x)`
- `Math.tan(x)`

b. Opérations de comparaison

Une opération de comparaison renvoie la valeur de vérité (true pour vrai et false pour faux) de la proposition correspondante. Elles sont utilisées pour exprimer des prédicats. Le tableau suivant présente les opérations de comparaison de base.

Tableau 7.2 : Opération arithmétiques en JavaScript

Opération	Nom	Exemple	Observation
==	Est égal à	$x=2==2$	x vaut true
!=	Différent de	$x=2!=3$	x vaut true
>	Supérieur à	$x=2>3$	x vaut false
<	Inférieur à	$x=2<3$	x vaut true
>=	Supérieur ou égal à	$x=2>=2$	x vaut true
<=	Inférieur ou égal à	$x=2<=3$	x vaut true
&&	Et	$x=(2!=3) \&\& (2>3)$	x vaut false
	Ou	$x=(2!=3) (2>3)$	x vaut true
!	Non (négation)	$x=! (2>=2)$	x vaut false

c. Opération sur chaînes de caractères

L'opérateur «+» permet aussi de concaténer deux chaînes de caractères. Par exemple «bon»+«jour» donne «bonjour». Remarquons que l'assignation `x=21+»pommes»` mettra la chaîne de caractères «21pommes» dans la variable `x` (elle transforme le nombre 21 en chaîne de caractères).

d. Opération d'affectation

Elle est réalisée par le symbole «=». Le symbole d'affectation est à différencier du symbole d'égalité qui est «==»

II. Utilisation des structures de controle

1. Les blocs d'instructions

Les blocs sont des ensembles d'instructions délimitées par des accolades (ouvrante { et fermante }). Elles permettent qu'un ensemble d'instructions soient exécutées ensemble.

Exemple 7.4 : Utilisation d'un bloc d'instructions en JavaScript

```
{
var nombre1 = new Number(17) ;
nombre1 = (nombre1-8)*3 ;
nombre1 = new Number() ;
var nombre2 = 6 ;
nombre2 = 6/2 ;
}
```

2. Les structures alternatives

Elles permettent d'exécuter un bloc d'instructions en fonction de la réalisation d'une condition. On en distingue deux :

- L'alternative réduite ou choix à une alternative ;
- L'alternative complète ou choix à deux alternatives.



Les commentaires :

On appelle commentaire un ensemble de caractères qui ne seront pas interpréter. Il permet de documenter un script. Les commentaires JavaScript peuvent être inscrit inline derrière un double slash (//)

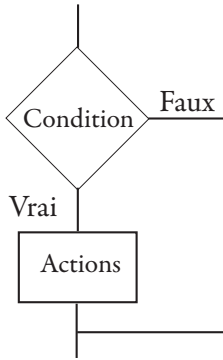


Figure 7.1 : Algorithme de l'alternative réduite



REMARQUE

Fonction alert :

La fonction «alert» qui sera étudiée en détail plus tard, permet d'afficher un texte et/ou dans une boîte de dialogue. Il correspond à l'instruction «Ecrire» étudiée en classe de première.

a. L'alternative réduite

Elle correspond à la structure algorithmique «Si ... Finsi» étudiée en classe de première. Sa syntaxe, présentée ci-après, est schématisée par l'algorithme de la Figure 7.1 ci-contre :

```

if (<condition>) {
    <action>;
    <action>;
    ...
}
  
```

Exemple 7.5 : Alternative simple en JavaScript

```

if ( sexe=="masculin" ) {
    alert ("Bonjour Monsieur");
}
  
```

- On compare le contenu de la variable «sexe» avec le texte «masculin». S'ils sont égaux, on exécute le bloc d'instructions. S'ils sont différents, on passe directement à l'instruction qui suit le bloc d'instructions ;
- Ici encore la casse est très importante car si le contenu de la variable «sexe» est «Masculin» alors le bloc d'instructions ne sera pas exécuté.

b. L'alternative complète

Elle correspond à la structure algorithmique «Si ... Sinon ... Finsi» étudiée en classe de première. Sa syntaxe, présentée ci-après, est schématisée par l'algorithme de la Figure 7.2 :

```

if (<condition>) {
    <action>;
    <action>;
    ...
} else {
    <action>
    <action>
    ...
}
  
```

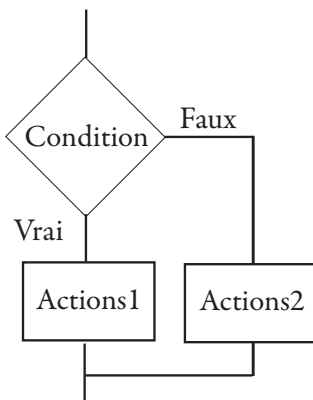


Figure 7.2 : Algorithme de l'alternative complète

Exemple 7.6 : Alternative complète en JavaScript

```
if ( sexe=="masculin" ) {
    alert ("Bonjour Monsieur") ;
}
else {
    alert ("Bonjour Madame") ;
}
```

- On compare le contenu de la variable «sexe» avec le texte «masculin». S'ils sont égaux, on exécute le premier bloc d'instructions.
- S'ils sont différents, on exécute le deuxième bloc d'instructions et on passe à l'instruction qui suit.

3. Les structures itératives

Les structures itératives permettent de répéter un bloc d'instructions suivant des critères précis :

- Tant qu'une condition reste vraie : structure While ;
- Jusqu'à ce qu'une condition devienne fausse : structure Do... While;
- Un nombre exact de fois : structure For.

a. La structure While

Elle correspond à la structure algorithmique «Tant Que» étudiée en classe de première. Elle permet d'exécuter les <actions> aussi longtemps que <condition> est vraie. Si la condition est fausse au début, aucune instruction n'est exécutée. Les actions doivent permettre d'incrémenter la condition. Sa syntaxe, présentée ci-après, est schématisée par l'algorithme de la Figure 7.3 ci-contre :

```
while ( <condition> ) {
    <actions>;
    ...
}
```

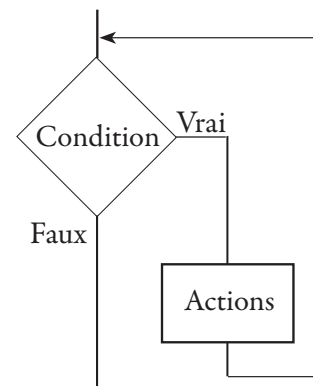


Figure 7.3 : Algorithme de la structure While

La clause else :

Else permet de prendre en compte la non-réalisation de la condition. Dans l'exemple ci-contre, «else» ne signifie pas que «sexe» a la valeur «féminin», mais juste que sa valeur n'est pas «masculin».





REMARQUE

While vs Do...While :

La boucle While doit être utilisée uniquement si on a une condition qui teste l'entrée dans la boucle, et la Do...While si la condition teste la sortie de la boucle.

Exemple 7.7 : Structure While en JavaScript

```
nbplaces=1 ;
while (nbplaces<=10){
    Alert("Bonjour Monsieur ! Bienvenu
    au séminaire") ;
    nbplaces=nbplaces+1 ;
}
```

- Le compteur (nbplaces) est initialisé à 1 ;
- A la première itération (premier passage dans la boucle) on teste la condition ($1 \leq 10$). Puisque la condition est vraie, on affiche le texte «Bonjour Monsieur ! Bienvenu au séminaire» et on incrémente le compteur qui a maintenant pour valeur 2 ;
- A la deuxième itération on teste encore la condition ($2 \leq 10$). Puisque la condition est vraie, on exécute encore le bloc d'instructions ;
- A la onzième itération, on teste à nouveau la condition ($11 \leq 10$). Puisque la condition est fausse, on passe à l'instruction qui suit le bloc d'instructions.

b. La structure Do...While

Elle correspond à la structure algorithmique «Répéter» étudiée en classe de première. Elle permet d'exécuter les <actions> aussi longtemps que <condition> est vraie. Si la condition est fausse au début, les instructions sont quand même exécutées une fois. Les actions doivent permettre d'incrémenter la condition. Sa syntaxe, présentée ci-après, est schématisée par l'algorithme de la Figure 7.4 :

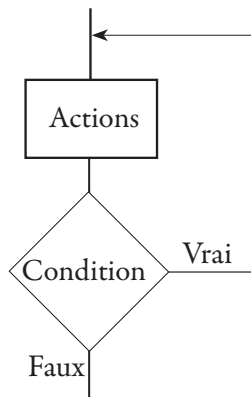


Figure 7.4 : Algorithme de la structure Do ... While

```
do {
    <actions>
    ...
} while ( <condition> )
```

Exemple 7.8 : Structure Do ... While en JavaScript

```

nbplaces=1 ;
do {
    Alert("Bonjour Monsieur ! Bienvenu
    au séminaire") ;
    nbplaces=nbplaces+1 ;
}
while (nbplaces<=10)

```

- Le compteur (nbplaces) est initialisé à 1 ;
- On affiche le texte «Bonjour Monsieur ! Bienvenu au séminaire» et on incrémente le compteur qui a maintenant pour valeur 2. On teste la condition ($2 \leq 10$). Puisque la condition est vraie, on passe à la deuxième itération ;
- On affiche le texte «Bonjour Monsieur ! Bienvenu au séminaire» et on incrémente le compteur qui a maintenant pour valeur 3. On teste la condition ($3 \leq 10$). Puisque la condition est vraie, on passe à la deuxième itération ;
- A la dixième itération, on teste à nouveau la condition ($11 \leq 10$). Puisque la condition est fausse, on passe à l'instruction qui suit le bloc d'instructions.

c. La structure For

Elle correspond à la structure algorithmique «Pour» étudiée en classe de première et dont l'algorithme est présenté dans la Figure 7.5 ci-contre. Son exécution se déroule comme suit:

- En partant d'une expression de départ (indice←valeur) on exécute le contenu des accolades;
- La variable indice est alors incrémentée (indice = indice + 1) et on exécute le contenu des accolades autant de fois que nécessaire;
- La boucle ne prend fin que lorsque la condition de continuation (indice←valeur2) devient fausse.

While vs For :

Les boucles While et Do...While seront préférées à la boucle For si on ne connaît pas d'avance le nombre d'itérations. Si ce nombre est connu d'avance, on utilisera la boucle For.



REMARQUE

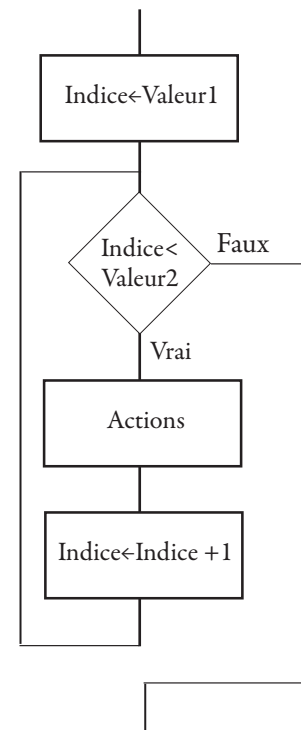


Figure 7.5 : Algorithme de la structure For



REMARQUE

Initialisation du compteur :

L'initialisation du compteur et la condition permettent de définir le nombre d'itérations. Cette initialisation est faite avant les boucles While et Do...While. Avec la boucle For, elle est faite dans la syntaxe.

Sa syntaxe est la suivante :

```
for ( <expression de depart> ; <condition de continuation> ; <incrémentation> ) {
  <action>
  ...
}
```

Exemple 7.9 : Structure For en JavaScript

```
for ( nbplace=1 ; nbplaces=10 ;
      nbplaces=nbplaces+1 ) {
  Alert("Bonjour Monsieur ! Bienvenu
  au séminaire") ;
  nbplaces=nbplaces+1 ;
}
```

III. Utilisation des tableaux

Les tableaux sont des variables qui permettent de stocker plusieurs données à la fois. Ils sont généralement utilisés pour stocker une liste d'éléments (nombres, textes, tableaux). De manière interne, les tableaux sont représentés de la manière suivante :

Indice 0	Indice 1	Indice 2	Indice 3	Indice n
Donnée 1	Donnée 2	Donnée 3	Donnée 4	Donnée n

La taille d'un tableau est dynamique. Elle augmente au fur et à mesure qu'on y ajoute des éléments. Elle est donnée par la propriété «length» (exemple : `tableau1.length`)

Pour créer un tableau on utilise les syntaxes ci-après :

```
tableau1 = new Array( ) ;
tableau2 = new Array(«donnée 1", «donnée 2", «donnée 3", «donnée 4") ;
tableau3 = [«donnée 1", «donnée 2", «donnée 3", «donnée 4"] ;
```

Pour accéder à un élément du tableau, on utilise la syntaxe suivante : `tableau1[indice]`. (exemple : `tableau1[3]` indique la valeur contenue à la position 4 du tableau).

N.B : la position 1 est repérée par l'indice 0

Exemple 7.10 : Utilisation des tableaux JavaScript

```
montableau = new Array( ) ;
montableau[0]= "Xaverie" ;
montableau[1]= "Robert" ;
montableau[2]= "Alain" ;
montableau[3]= "Zoé" ;
montableau[4]= "Martin" ;
montableau[5]= "Sophie" ;
montableau[3]= "Balbine" ;
montableau.sort( ) ;
alert ("Voici la liste des prénoms par
ordre alphabétique")
for (indice=0 ; indice<montableau.
length ; indice=indice+1) {
alert (montableau[indice]) ;
}
```

- La première ligne du script déclare un tableau vide ;
- Ensuite on insère respectivement les textes «Xavérie», «Robert», «Alain», «Zoé», «Martin», «Sophie» ;
- On remplace la valeur d'indice 3 (c'est-à-dire «Zoé») par «Balbine» ;
- On range les données du tableau par ordre alphabétique ;
- On entre dans une boucle «For» avec pour compteur «indice» initialisée à 0, la condition de continuation «indice<6» car la taille du tableau est 6 ;
- La boucle va afficher les valeurs du tableau indice après indice.

Propriété et méthode :



Une propriété ou attribut est une caractéristique d'un tableau, comme la taille d'un tableau. Une méthode est un ensemble d'opérations que l'objet peut réaliser. Les méthodes seront étudiées plus loin dans le chapitre.

IV. Utilisation des fonctions

1. Notion de fonction et de procédure

On appelle fonction un sous-programme qui permet d'effectuer un ensemble d'instructions par simple appel dans le corps du programme principal. Les fonctions et les procédures, en réalisant l'exécution dans plusieurs parties du programme d'une série d'instructions, permettent une simplicité du code et donc une taille de programme minimale.

Dans JavaScript, les fonctions et les procédures sont définies par le mot clé **function** suivi du nom de la fonction et de liste de ses arguments. La syntaxe de la déclaration d'une fonction est la suivante :

```
function mafonction(argument1, argument2, ..., argumentN) {
    <actions>
    [Return sortie]
}
```

Exemple 7.11 : Création de fonction et de procédure

```
function addition(a,b) {
    c=a+b;
    return c;
}
function danger( ) {
    alert("fonction danger exécutée");
}
```

- Une fonction sans argument doit tout de même comporter des parenthèses « mafonction() » ;
- L'instruction «return sortie» est facultative ;
- «sortie» est un argument ou une variable utilisée par la fonction



REMARQUE

Fonction ou procédure :

Une fonction retourne une valeur (numérique, booléenne, etc.) ; ce qui n'est pas le cas pour une procédure. Le retour de valeur se fait dans le code par le mot clé return.

Pour appeler une fonction, on utilise simplement le nom de la fonction, suivi de la liste des arguments.

Exemple 7.12 : Appel de fonction en JavaScript

```
somme = addition(2, 3) ;  
danger ( ) ;
```

- Lorsque la fonction renvoie une valeur, on peut l'affecter à une variable. Ici, on affecte le résultat de l'addition de 2 et de 3 dans la variable «somme» ;
- Si la fonction n'a pas d'arguments, on met tout de même les parenthèses. Le script suivant ouvre la fenêtre d'alerte définie dans la fonction danger().

2. Notion de méthode

Les méthodes sont des fonctions associées à des objets (images, liens, tableaux, etc). Elles sont généralement spécifiques aux objets, mais certaines peuvent s'appliquer à différents objets. Les méthodes s'utilisent avec le nom de l'objet, suivi d'un point et du nom de la méthode. (Exemple : monobjet.méthode).

On a déjà rencontré la méthode `sort` qui permet de trier un tableau : `montableau.sort()`.

Une méthode assez largement utilisée est la méthode «write» de l'objet `document` qui permet d'écrire dans une page Web.

Exemple 7.13 : Utilisation de `document.write`

```
document.write("bonjour"+nom) ;  
document.write("Le titre est  
  \"Alcools\".")
```

- La méthode `write` permet d'écrire un texte (mis en quote) et/ou la valeur d'un objet. Si la variable «nom» contient la valeur « Siriki », la fonction va écrire « Bonjour Siriki »

Appel de fonction:

Ladéclaration d'une fonction est son expression abstraite (Exemple : $f(x)=2x^2+4x-7$). Son appel est le calcul de l'image d'une valeur par la fonction (Exemple : $f(2)=11$).



REMARQUE



REMARQUE

Fonction Prompt :

La fonction prompt réalise une opération de lecture, au même titre que l'instruction algorithmique "lire" étudiée en classe de première.

- Si on souhaite ajouter des caractères spéciaux, on doit faire comprendre à JavaScript qu'il s'agit d'un caractère à afficher et non à interpréter. On utilise pour cela le caractère \. Ainsi le code va afficher le texte « Le titre est "Alcools" »

V. Eléments d'interaction

1. Les fenêtres de dialogue

a. Demande d'information

Lorsqu'on souhaite que l'utilisateur entre une donnée (son nom, un montant, une date, ...) on utilise la fonction prompt. Sa syntaxe est la suivante :

```
Prompt ("question", "réponse par défaut") ;
```

Exemple 7.14 : Utilisation de la fonction prompt en JavaScript

```
mavariabile=prompt("Entrer votre nom",  
"Votre nom") ;
```

- L'exécution de la fonction «prompt» va provoquer l'affichage de la boîte de dialogue de la Figure 7.6 ;
- L'utilisateur pourra entrer son nom ou laisser le texte « Votre nom » affiché par défaut. S'il clique sur « OK », le texte affiché sera affecté à la variable «mavariabile». S'il clique sur « Annuler », la valeur «Null» sera affectée à «mavariabile».

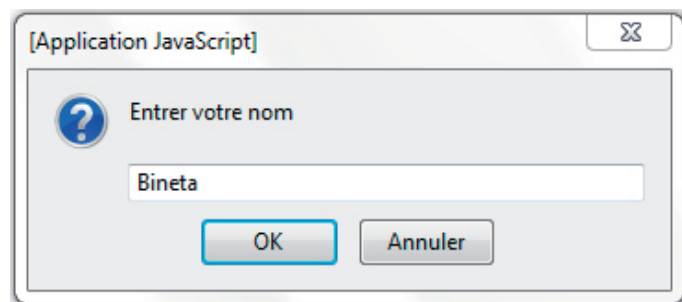


Figure 7.6: Exécution de la fonction «prompt»

La réponse de la commande `prompt` est toujours une chaîne de caractères. Si on désire récupérer un nombre, il faut utiliser la fonction `parseInt` (pour convertir en un nombre entier) ou `parseFloat` (pour la convertir en nombre à virgule).

Exemple 7.15 : Conversion de type d'une fonction `prompt`

```
mavariabile=parseInt(prompt("Entrer un
montant", "0")) ;
```

On peut également demander une confirmation pour une action (suppression d'élément, fermeture de page, ...). On utilise pour cela la fonction «*confirm*». Sa syntaxe est :

```
confirm("Demande de confirmation") ;
```

Exemple 7.16 : Utilisation de la fonction `confirm` en JavaScript

```
mavariabile=confirm("Voulez-vous fermer
la page ?") ;
```

- L'exécution de la fonction «*confirm*» va provoquer l'affichage de la boîte de dialogue de la Figure 7.7 ci-après ;
- Si l'utilisateur clique sur « OK », «*mavariabile*» aura la valeur `true`. Sinon la valeur `false`.

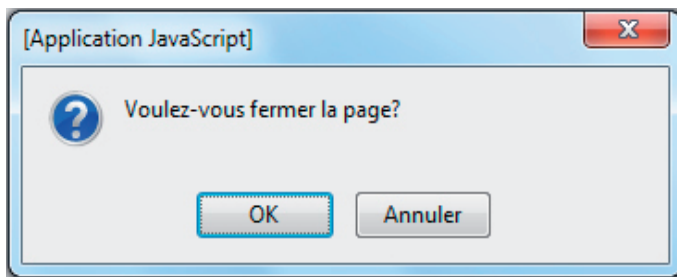


Figure 7.7: Exécution de la fonction «*prompt*»

b. Alertes

La fonction `alert` permet d'ouvrir une fenêtre d'alerte contenant un certain message, afin d'avertir l'utilisateur. Il doit alors cliquer sur OK avant de continuer.

Fonction Alert :

La fonction `alert` réalise une opération d'écriture, au même titre que l'instruction algorithmique "écrire" étudiée en classe de première.





REMARQUE

Caractères spéciaux :

Les caractères spéciaux permettent de mettre en évidence soit des caractères non visuels tels le retour à la ligne (`\n`) et la tabulation (`\t`), soit des caractères à ne pas confondre avec ceux du script tels les guillemets (`\"`) .

Exemple 7.17 : Utilisation de la fonction `alert` en JavaScript

```
alert("Bonjour !\nBienvenu dans JavaScript");
```

- La fonction «`alert`» va provoquer l’affichage de la boîte de dialogue de la Figure 7.8
- Le texte «`\n`» est un caractère qui permet d’aller à la ligne

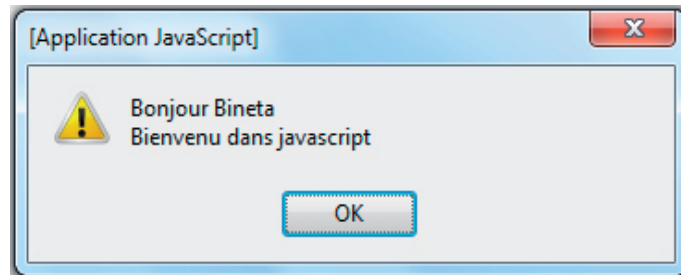


Figure 7.8: Exécution de la fonction «`alert`»

2. Les événements

Les événements sont des actions de l’utilisateur qui vont pouvoir donner lieu à une interactivité. L’événement par excellence est le clic de souris, car c’est le seul que le HTML gère.

Il est possible d’associer des fonctions ou des méthodes à des événements, de la même manière qu’on le fait avec CSS.

Ce sont les gestionnaires d’événements qui permettent d’associer une action à un événement. Le nom d’un gestionnaire d’événement commence toujours par le mot «`on`» (`onclick`, `onchange`, `onsubmit` ...)

La syntaxe d’un gestionnaire d’événement est la suivante :

```
onEvenement= "Action_Javascript ou  
Fonction() ;"
```

Exemples 5.18 :

```
onclick= "alert("Bonjour");"  
onchange="mafonction() ;"
```

Les gestionnaires d'événements sont associés à des objets, et leur code s'insère dans la balise de ceux-ci. Les événements ne peuvent pas être associés à n'importe quel objet. Ainsi, l'événement On-Change ne peut s'appliquer à un lien hypertexte. Le tableau ci-après présente les principaux gestionnaires d'événements suivant les objets auxquels ils s'appliquent :

Tableau 7.3 : Objet et gestionnaires d'événements en JavaScript

Objet	Gestionnaires d'événements associables	Commentaires
Lien hypertexte	onclick, onmouseover, onmouseout	Clic gauche sur le lien, survol du lien, fin du survol du lien
Bouton, case à cocher, bouton radio, bouton reset	onclick	Clic gauche sur l'objet
Liste de sélection d'un formulaire	onchange	Choix d'un élément de la liste
Bouton submit	onsubmit	Bouton d'envoi du formulaire
Champ de texte ou zone de texte	onchange	Saisie puis validation d'une valeur

3. Ajout de contrôles

Comme on l'a déjà dit, JavaScript permet d'écrire dans une page Web, grâce à la fonction `write` de l'objet `document`. En fait, JavaScript va plus loin et permet de construire entièrement une page Web en écrivant le code HTML.

La syntaxe de la fonction «`document.write`» est la suivante :

```
document.write("Code_html") ;
```

Exemple 7.18 : Ajout d'une image à une page Web

```
document.write("<img src=\"image1.  
jpg\" />")
```

- pour insérer l'image, on écrit le code HTML de l'image :
` ;`

- les guillemets qui sont placés à côté des parenthèses font partie de la syntaxe JavaScript de la fonction «document.write». Ceux qui sont à côté du nom de l'image font partie de la syntaxe HTML de la balise . Ils sont précédés du caractère «\» pour que JavaScript ne les interprète pas.

Exemple 7.19 : Ajout d'une zone de saisie à une page Web

```
document.write("<input type=\"text\"  
name=\"Nom\" />") ;
```

Exemple 7.20 : Ajout d'un bouton à une page Web

```
document.write("<input type=\"text\"  
name=\"Nom\" />") ;
```

4. Les cookies

Le protocole HTTP ne possède pas d'état, ce qui signifie qu'il ne possède pas de mémoire. Un client ouvre une connexion vers un serveur, récupère un document (une image ou d'autres données), puis ferme la connexion. Par la suite, même si le serveur envoie des données à ce client spécifique, il le fera sans le reconnaître. Pour contourner cette limitation, on utilise les cookies.

Les cookies sont des fichiers textes stockés sur le disque dur de l'utilisateur (l'internaute) qui permettent au serveur de le reconnaître d'une page à une autre. Ils sont généralement utilisés pour noter les préférences de l'utilisateur ou ses choix de navigation.

JavaScript permet de d'écrire et de lire des cookies. Spécifiquement, pour créer un cookie de session, on utilise la syntaxe suivante :

```
document.cookie= "nomcookie=contenu"
```

5. Accès aux éléments de la page

Pour accéder à un élément de la page (lien, zone de texte, image, etc) on utilise les attributs « name » ou « id » de ces éléments.

a. Accès par l'attribut «name»

Pour accéder à un élément de la page, on lui donne une valeur d'attribut «name» dans le code HTML. Ensuite, dans le script JavaScript, on utilise l'une des syntaxes suivantes :

`document.type_élément.name`

`document.type_élément["name"]`

- L'attribut «name» ne doit pas comporter des caractères tels que les espaces ou les tirets ;
- «type_élément» désigne des éléments d'une page Web, suivant l'arborescence de la Figure 7.9 ci-après :

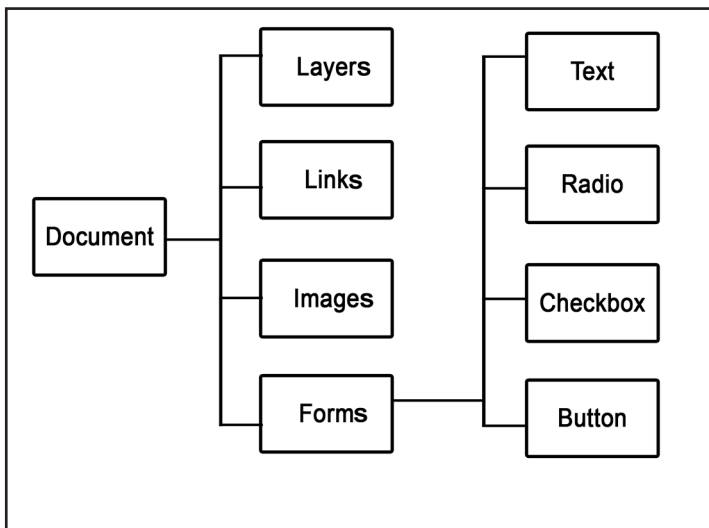


Figure 7.9 : Eléments d'une page Web

Exemple 7.21 : Accès à des éléments HTML par "name"

```
document.link.page.href="page2.html";  
document.images["img"].src= "img.jpg";  
document.form.text.textel.value= "Bon-  
jour";
```

- La première ligne permet de modifier la référence hypertexte du lien nommé «page2» pour la faire pointer vers la page Web nommée «page2.html» ;
- La deuxième ligne permet de modifier la source de l'image nommée «img» ;

- La troisième ligne permet de modifier le texte inscrit dans la zone de texte nommé «textel».

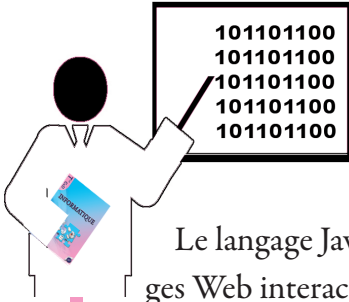
b. Accès par l'attribut «id»

Pour accéder à un élément de la page, on lui donne une valeur d'attribut «id» dans le code HTML. Ensuite, dans le script JavaScript, on utilise la syntaxe suivante :

```
document.getElementById("id")
```

Exemple : Accès à des éléments HTML par "Id"

```
document.getElementById("page2").  
    href="page2.html";  
document.getElementById("img").scr=  
    "img.jpg";  
document.getElementById("textel").va-  
    lue= "Bonjour";
```



RESUME

Le langage JavaScript définit un ensemble d'éléments nécessaire à la création de pages Web interactives. Chaque élément joue un rôle particulier dans les scripts :

- Les variables permettent de conserver des valeurs utiles dans les opérations arithmétiques ;
- Les structures de contrôles servent à vérifier des conditions (structures alternatives) et à exécuter des actions tant qu'une condition n'est pas vérifiée (structures itératives) ;
- Les tableaux contiennent des valeurs de même type et ces dernières sont manipulées à l'aide d'un indice pour parcourir le tableau.
- Les fonctions exécutent un bloc d'instructions et retournent un résultat par l'usage du mot-clé Return.
- Les procédures exécutent un bloc d'instructions réalisant une tâche bien précise.
- Les éléments d'interactions permettent d'interagir avec l'utilisateur au travers de fenêtres de dialogue, d'événements, d'ajout de contrôles et d'utilisation des cookies.

EXERCICE
THEORIQUE**EXERCICE 1 : Le choix des variables en JavaScript.**

1. Dire si les expressions suivantes peuvent être considérées comme des variables :
- a) MavariabLe ; b) ma variable ; c) mes2variables ; d) 2variables ; e) MaVariable ;
 f) mèsvariables ; g) une variable ; h) une déclaration de variable ; i) date ;
 j) unedéclarationdevariable ; k) uneDéclarationDeVariable.

Justification _____

2. Dire si les variables suivantes sont différentes :
- a) *MaVariable* et *mavariabLe* ; b) *MavariabLe* et *mavariabLe*
 c) *unedéclarationdevariable* et *unedéclarationdevariable*
 d) *mavariabLe* et *mavariabLe* e) *résultat* et *resultat*

Justification _____

EXERCICE
THEORIQUE**EXERCICE 2 : Les cinq types de variables en langage JavaScript.**

1. Compléter les phrases suivantes :
- Une variable de type :
- a) Number : est une variable qui contient _____
 b) String : est une variable qui contient _____
 c) Boolean : est une variable qui peut prendre _____
 d) Array : est une variable qui stocke _____
 e) Date : est une variable qui stocke _____ au format _____
2. Dire si les types de variable suivants sont corrects :
- a) date ; b) Booléan ; c) array ; d) string ; e) String ; f) number ; g) Array

Justification _____

EXERCICE
THEORIQUE**EXERCICE 3 : Déclaration de variables sans affectation :***var mavariable = new typevariable() ;*

A partir des déclarations de variables sans affectation compléter les phrases suivantes :

a) `var nombre = new Number() ;`

La variable _____ de type _____ contient la valeur par défaut _____

b) `var texte = new String() ;`

La variable _____ de type _____ contient la valeur par défaut _____

c) `var condition = new Boolean() ;`

La variable _____ de type _____ contient la valeur par défaut _____

d) `var tableau = new Array() ;`

La variable _____ de type _____ contient la valeur par défaut _____

e) `var datte = new String() ;`

La variable _____ de type _____ contient la valeur par défaut _____

EXERCICE
THEORIQUE**EXERCICE 4 : Déclaration de variables avec affectation :***var mavariable = new typevariable(valeur) ;*

Donnez les valeurs initiales des variables suivantes déclarées avec affectation :

a) `var nombre = new Number (2) ;`

La variable _____ de type _____ contient la valeur initiale _____

b) `var texte = new String ('elle est Super cette banane') ;`

La variable _____ de type _____ contient la valeur initiale _____

c) `var texte = new String ("elle est Super cette banane") ;`

La variable _____ de type _____ contient la valeur initiale _____

d) `var condition = new Boolean (true) ;`

La variable _____ de type _____ contient la valeur initiale _____

e) `var tableau = new Array (10, 36, 84, 54) ;`

La variable _____ de type _____ contient les valeurs initiales _____

f) `var datte = new Date (2012,03,08) ;`

La variable _____ de type _____ contient la valeur initiale _____



EXERCICE 5 : Utilisation des variables :

Après avoir réalisé la trace des codes ci-après, quel est le contenu des variables données :

a) nombre _____

```
var nombre1 = new Number ( 3 );
var nombre2 = new Number ( 5 );
var nombre = new Number ( );
nombre = nombre1 + nombre2;
```

b) nombre _____

```
var nombre1 = new Number ( 3 );
var nombre2 = new Number ( 5 );
var nombre = new Number ( 10 );
nombre = nombre1 + nombre2 ;
nombre = nombre + nombre1 ;
```

c) nombre1 _____

```
var nombre1 = new Number ( 3 );
var nombre2 = new Number ( 5 );
var nombre = new Number ( 10 );
nombre = nombre1 + nombre2 ;
nombre1 = nombre + nombre1 ;
```

d) nombre2 _____

```
var nombre1 = new Number ( 3 );
var nombre2 = new Number ( );
var nombre = new Number ( );
nombre = nombre1 + nombre2 ;
nombre2 = nombre + nombre2 ;
```



EXERCICE 6 : Opérations sur les variables :

Après avoir réalisé la trace du code ci-contre, quel est le contenu des variables suivantes :

a) nombre _____

b) nombre1 _____

c) nombre2 _____

d) nombre3 _____

```
var nombre1 = new Number ( 2 );
var nombre2 = new Number ( );
var nombre3 = new Number ( 5 );
var nombre = new Number ( );
nombre = nombre1 + nombre ;
nombre1 = nombre * nombre3 ;
nombre2 = nombre1 / nombre ;
nombre3 = nombre1 - nombre2 ;
```



EXERCICE 7 : Consolidation des acquis sur la déclaration des variables

1. Parmi les déclarations suivantes donnez celles qui sont incorrectes et corrigez :

a) var cahierde = new string(' Mathématiques ') ; _____

b) var âge = new Number(24) : _____

c) var prénom = new String(" Mamadou ") : _____

d) var nom = new string(Bineta) : _____

e) var condition = boolean(true) : _____

2. Répondre par Vrai ou faux :

- a) Une variable commence toujours par une lettre et peut contenir des espaces : _____
- b) Les noms de variables ayant la même orthographe mais écrits l'un en majuscule l'autre en minuscule sont les mêmes : _____
- c) Les noms de variable peuvent contenir des accents : _____
- d) Les types de variable commencent toujours par une lettre majuscule : _____
- e) Une variable est un objet représenté dans un programme par un nom : _____
- f) La manipulation des chaînes de caractères requiert les apostrophes ' ' ou les cotes " " : _____
- g) Une déclaration de variable sans affectation, donne à cette variable une valeur par défaut en fonction du type de la variable : _____
- h) Toute instruction tient sur une ligne et se termine toujours par un point virgule(;) _____



EXERCICE 8 : Les structures conditionnelles

1. Quelle est la valeur de la variable *note* à la fin de l'exécution du script ci-contre : _____

```
var elevedesordonne = new Boolean (false) ;
var note = new Number (12);
If ( elevedesordonne == true ) {
    Note = note - 2;
}
```

2. Soit le script ci-contre

a) Quelle est la valeur la variable *note* à la fin de l'exécution du script suivant : _____

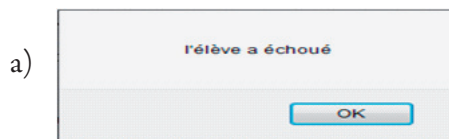
b) Quel est le bloc d'instructions qui a été exécuté : _____

c) Si la variable *elevedesordonne* contenait la valeur booléenne true, quelle aurait été la valeur de la variable *note* _____

```
var elevedesordonne = new Boolean (false) ;
var note = new Number (12);
If ( elevedesordonne == true ) {
    note = note - 2 ;
}
else{
    note = note + 2 ;
}
```

3. Soit le script ci-contre. Qu'affiche-t-il?

```
var moyenne = new Number ( 6 ) ;
if ( moyenne >= 10 ){
    alert(" l'élève a reussi " ) ;
}
else{
    alert(" l'élève a échoué " ) ;
}
```





EXERCICE 9 : La structure itérative for

1. Examiner le script JavaScript ci-après et compléter le tableau suivant :

Compteur	Condition	Somme
0		0
1	1 <= 5 ? vrai	0+1=1
2	2 <= 5 ? vrai	1+2=3
3	3 <= 5 ? vrai	3+3=6
---	-----	-----
---	-----	-----
---	-----	-----

```

var compteur = new Number ( );
var somme = new Number ( );
for (compteur = 1 ; compteur <= 5 ; compteur + 1 ){
    somme = somme + compteur ;
}
Alert (somme) ;
    
```

2. A partir du script JavaScript et du tableau précédent, répondre aux questions suivantes :

- a) Quelle est la valeur initiale de la variable *compteur* : ___ et sa valeur finale : ___
- b) Combien de fois on exécute la boucle for : ___ Donner la(s) ligne(s) affectée(s) ___
- c) Quelle est la valeur du compteur lorsque la condition devient fausse : ___
- d) Que fait la ligne 6 du script précédent : _____

3. Compléter les phrases suivantes :

- a) for(compteur = 0 ; compteur < 5 ; compteur = compteur + 1) ;
 valeur initiale du compteur : ___ valeur finale : ___ valeur qui rend la condition fausse : ___
 Combien de fois on exécute la boucle for : ___
- b) for(compteur = 1 ; compteur <= 5 ; compteur = compteur + 1) ;
 valeur initiale du compteur : ___ valeur finale : ___ valeur qui rend la condition fausse : ___
 Combien de fois on exécute la boucle for : ___
- c) for(compteur = 0 ; compteur != 5 ; compteur = compteur + 1) ;
 valeur initiale du compteur : ___ valeur finale : ___ valeur qui rend la condition fausse : ___
 Combien de fois on exécute la boucle for : ___
- d) for(compteur = 4 ; compteur > 0 ; compteur = compteur - 1) ;
 valeur initiale du compteur : ___ valeur finale : ___ valeur qui rend la condition fausse : ___
 Combien de fois on exécute la boucle for : ___
- e) for(compteur = 1 ; compteur > 0 ; compteur = compteur + 1) ;
 valeur initiale du compteur : ___ valeur finale : ___ valeur qui rend la condition fausse : ___
 commentaire : _____



EXERCICE 10 : La structure itérative While

1. Examiner le script JavaScript ci-après et remplir le tableau suivant :

Lignes	Compteur	Condition	Somme
1	1		
2			0
4		1<=5 ? vrai	
5			0+1=1
6	1+1=2		
4		2<=5 ?vrai	
__			-----
6	-----		
__		-----	
5			3+3=6
6	3+1=4		
4		4<=5 ?vrai	
5			6+4=10
6	4+1=5		
4		5<=5 ? __	
5			10+5=15
6	5+1=6		
4		-----	
__			__

```

Ligne 1 var compteur = new Number ( 1 ) ;
Ligne 2 var somme = new Number ( ) ;
Ligne 3
Ligne 4 while ( compteur <= 5 ) {
Ligne 5     somme = somme + compteur ;
Ligne 6     compteur = compteur + 1 ;
Ligne 7 }
Ligne 8 alert (somme) ;
    
```

2. Compléter les phrases suivantes :

- a) Quelle est la valeur initiale de *compteur* : __
- b) Quelle est la valeur finale de *compteur* : __
- c) Quelle est la valeur de *compteur* qui rend la condition fausse : ____
- d) Combien de fois est exécutée la boucle while : _____
- e) quelle est la valeur de somme à la fin de l'exécution du script : ____
- f) que fait ce script : _____



EXERCICE 11: Utilisation des tableaux

1. Soit la déclaration avec affectation du tableau suivant :

```
var numerotelephone = new Array( 10,36,84,54 ) ;
```

- a) Complétez le tableau en fonction de la déclaration :

0	1	2	3
__	__	__	__
- b) Combien d'éléments possède ce tableau : ____ quelle est donc la taille du tableau : ____
- c) Valeur contenue dans la 1^{ère} case du tableau : ____ que vaut donc numerotelephone [0]: ____
- d) Valeur contenue dans la 2^{ème} case du tableau: ____ que vaut donc numerotelephone [1]: ____

- e) Valeur contenue dans la 3^{ème} case du tableau : ___ que vaut donc numerotelephone[2] : ___
- f) Valeur contenue dans la 4^{ème} case du tableau : ___ que vaut donc numerotelephone [3] : ___

2. Examiner le code JavaScript ci-après et répondre aux questions suivantes :

- a) Quelle est la valeur initiale de la variable indice dans la boucle for : ___
- b) Quelle est la valeur finale de la variable indice dans la boucle for : ___
- c) Combien de fois est exécutée la boucle for : ___
- d) Que fait ce code : _____


```
var numerotelephone = new Array ( 10, 36, 84, 54 );
var indice = new Number ( ) ;
for ( indice = 0 ; indice <= 3 ; indice = indice + 1 ) {
    alert ( numerotelephone[indice] );
}
```



EXERCICE 13 : Appel d'une fonction

1. En vous inspirant du code ci-contre, répondre aux questions :
 - a) A quoi servent les symboles // : _____
 - b) Quelle est le nom de la fonction : _____
 - c) Cette fonction a combien de paramètres : _____
 - d) Donner le nom de ces paramètres : _____
 - e) Que fait cette fonction : _____
 - f) Quelle est la valeur de s : _____

```
function aire (longueur, largeur) { //entête de la fonction
    var resultat = longueur x largeur ; // calcul de l'aire
    return resultat ; // retourne le résultat calculé
}
var s = aire (20,10) ; // appel de la fonction avec les paramètres
alert(s) ; // affichage du résultat
```



EXERCICE 14 : Appel d'une procédure

1. En vous inspirant du code ci-contre, répondre aux questions suivantes:
 - a) Quel est le nom de la fonction ? _____

```
Ligne 1  fonction pair (nombre) {
Ligne 2  var reste = nombre % 2 ;
Ligne 3  if ( reste == 0 ) {
Ligne 4  alert (" nombre pair " ) ;
Ligne 5  }
Ligne 6  else{
Ligne 7  alert(" nombre impair " ) ;
Ligne 8  }
Ligne 9  }
Ligne 10 pair(10);
Ligne 11 pair(3);
```

- b) Cette fonction a-t-elle des paramètres : _____
- c) Que fait cette fonction : _____
- d) Que fait la ligne 10 : _____
- e) Que fait la ligne 11 : _____
- f) Cette fonction retourne t - elle un résultat : _____
- g) Comment appelle t - on les fonctions qui ne retournent aucun résultat : _____



EXERCICE 15 : Appel d'une procédure

1. Examiner le code ci-contre et répondre aux questions suivantes :
 - a) Combien de fois la boucle While est-elle exécutée : _____
 - b) A chaque exécution qu'affiche t-elle : _____
 - c) Que fait ce script : _____

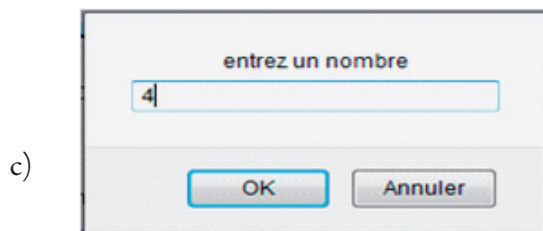
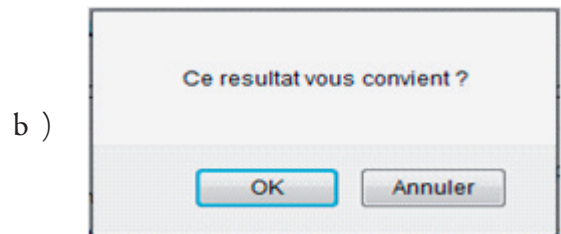
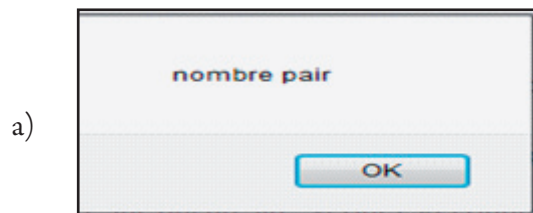
```

function identite () {
    var nom = prompt (" quel est votre nom ");
    var prenom = prompt (" quel est votre prenom ");
    alert (vous vous appelez «" + nom + " " + prenom );
}
var compteur = new number ( 1 );
while (compteur <= 2 ) {
    identite ();
}
    
```



EXERCICE 16 : Alert, prompt et confirm

Examiner les images des boîtes de dialogue ci-dessous et le script qui les crée.



```
Ligne 1  function pair (nombre) {  
Ligne 2  var reste = nombre % 2 ;  
Ligne 3  if( reste == 0) {  
Ligne 4      alert (" nombre pair " );  
Ligne 5  }  
Ligne 6  else{  
Ligne 7  alert(" nombre impair " );  
Ligne 8  }  
Ligne 9  }  
Ligne 10 var nombre = new Number ( ) ;  
Ligne 11 var confirmation = new Boolean ( ) ;  
Ligne 12  
Ligne 13 nombre = prompt (" entrez un nombre");  
Ligne 14 pair (nombre) ;  
Ligne 15 confirmation = confirm (" Ce résultat vous convient ? " ) ;  
Ligne 16  
Ligne 17 if (confirmation == true ) {  
Ligne 18     alert (" OK " );  
Ligne 19 }  
Ligne 20 else{  
Ligne 21     alert (" à vous de nous dire où sa coince !!! " );  
Ligne 22 }
```

1. Commenter le script qui est proposé
2. Associer les boîtes de dialogue aux lignes du script qui ont provoqué leur affichage
3. Laquelle des lignes 18 et 21 sera exécutée ? Pourquoi.



EXERCICE 17 : Consolidation des acquis sur les fenêtres de dialogue

Répondez par vrai ou faux :

- a) La boîte de dialogue *alert* permet d'afficher un texte à l'écran : _____
- b) *alert* est une fonction du langage JavaScript : _____
- c) La boîte de dialogue de la fonction *confirm* affiche deux (2) boutons (Ok et delete) à l'utilisateur pour confirmer une opération : _____

- d) Lorsque l'utilisateur ne clique pas sur le bouton OK de la boîte de dialogue de la fonction *confirm*, la valeur booléenne qui est retournée est *true* : _____
- e) *prompt* est une fonction du langage JavaScript : _____
- f) *confirm* est une procédure du langage JavaScript : _____
- h) La boîte de dialogue *prompt* permet d'afficher un texte à l'écran puis de récupérer la valeur saisie par l'utilisateur et de la ranger dans une variable : _____

8

INTEGRATION DE JAVASCRIPT A HTML

SOMMAIRE

- ▶ Intégration directe au code HTML
- ▶ Intégration à un fichier externe
- ▶ Code de la page d'activité avec scripts JavaScript
- ▶ Code de la fonction «corrigé»

MOTS-CLES

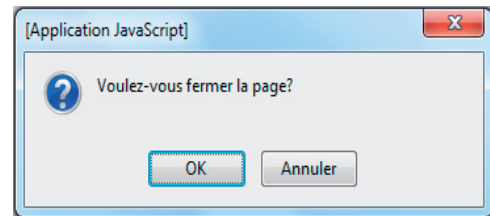
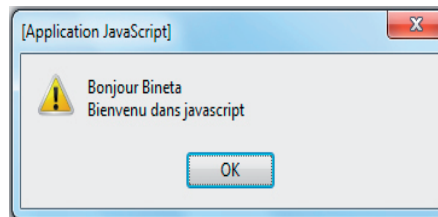
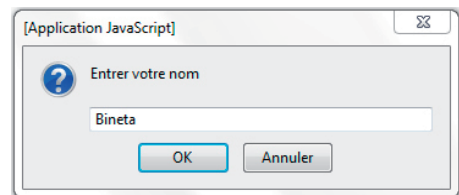
- PseudoURL
- Gestionnaire d'évènements
- Fichier externe

PRE-REQUIS

- Langage JavaScript
- Langage HTML



```
<html>
<head> </head>
<body>
  <script type="text/javascript">
    var nom = new String();
    nom=prompt("Entrer votre nom");
    alert("Bonjour "+nom+"\nBienvenu dans javascript");
    confirm("Voulez-vous fermer la page?");
  </script>
</body>
</html>
```



OBJECTIFS

- Utiliser la balise <script>
- Utiliser des pseudo-URL
- Utiliser des gestionnaires d'évènements
- Utiliser un fichier externe «.js»

Test d'auto-évaluation

Nom : Classe :

Questionnaire sous forme de QCM. Pour chaque question, une seule réponse est juste. Chaque bonne réponse vaut 1 point, chaque mauvaise -0,5 point.

Question 1 : Un réseau de ville est encore appelé :

- LAN.
- MAN.
- WAN.

Question 2 : Une paire torsadée utilise des connecteurs :

- RJ-35.
- BNC.
- RJ-45 répétitives.

Question 3 : La topologie Ethernet est utilisée dans les réseaux :

- En bus.
- Maillés.
- En étoile.
- En anneau.

[Voir le corrigé](#)

1. On souhaite, lorsque l'élève aura inscrit son nom dans la zone nom et qu'il aura validé, qu'un message lui souhaite la bienvenue. Comment peut-on le faire avec JavaScript ?
2. On souhaite, après un clic sur le lien hypertexte [Voir le corrigé](#). Comment peut-on le faire avec JavaScript ?
3. On souhaite enfin qu'en cliquant sur le bouton Valider, que l'on comptabilise le nombre de questions validées (celles pour lesquelles on a coché une réponse), le nombre de bonnes réponses, et le résultat. Le tout est affiché dans une boîte de dialogue. Comment peut-on le faire avec JavaScript ?


```
<P><B>Question 2 :</B>&nbsp;Une paire torsadée utilise
des connecteurs :
```

```
<I> <LI><INPUT TYPE="radio" NAME="question2"
VALUE="A"> RJ-35.</LI>
<LI><INPUT TYPE="radio" NAME="question2"
VALUE="B"> BNC.</LI>
<LI><INPUT TYPE="radio" NAME="question2"
VALUE="C"> RJ-45 répétitives.</LI> </I>
```

```
<P><B>Question 3 :</B>&nbsp;La topologie Ethernet est
utilisé dans les réseaux :
```

```
<I> <LI><INPUT TYPE="radio" NAME="question3"
VALUE="A"> En bus.</LI>
<LI><INPUT TYPE="radio" NAME="question3"
VALUE="B"> Maillés.</LI>
<LI><INPUT TYPE="radio" NAME="question3"
VALUE="C"> En étoile.</LI>
<LI><INPUT TYPE="radio" NAME="question3"
VALUE="D"> En anneau.</LI> </I>
```

```
<center><INPUT TYPE="button" VALUE="Valider">
</center>
<A href="corrigé.html">Voir le corrigé</A>
</FORM>
</blockquote>
</BODY>
</HTML>
```

I. Intégration directe au code HTML

1. Les balises <script>

Le code Javascript peut se placer n'importe où dans le code HTML, entre les balises <script> et </script>, dans le «head» ou dans le

«body». Toutefois, il est important de comprendre que le navigateur interprète le code HTML de haut en bas. Ainsi, si le script est placé dans le «head», il sera exécuté avant l’affichage de la page. Aussi on observera les deux règles suivantes :

- Placer le code du script après le code des éléments auxquels il accède ;
- Placer les fonctions dans le «head» pour éviter que les scripts qui les appellent ne puissent y accéder, faute de n’avoir pas encore été chargés.

La syntaxe générale d’un script JavaScript est la suivante :

```
<SCRIPT type= "text/javascript">  
    Code du script ;  
</SCRIPT>
```

Exemple 8.1 : Intégration de JavaScript par la balise <script>

```
<BODY >  
    <BLOCKQUOTE>  
    <FONT FACE="Times New Roman">  
  
    <SCRIPT  
        document.  
write("<CENTER><H2>Test d'auto-éva-  
luation</H2></CENTER>");  
        .....  
    <SCRIPT>  
</BODY>
```

2. Les pseudo-URL

Les pseudo-URL sont des références de lien qui permettent d’exécuter un script JavaScript lorsqu’on clique sur ledit lien. Ils commencent par le mot-clé « javascript : ». Lorsqu’une telle référence est chargée, le code qui se trouve derrière est exécuté. Ce code peut être un ensemble de commandes ou une fonction.

Exemple 8.2 : Intégration de JavaScript par une pseudo-URL

```
<A href="javascript: fen=open('corrigé.html'); fen.focus();"> Voir le corrigé</A>
```

- Le code «`fen=open('corrigé.html')`» crée une nouvelle fenêtre de navigateur nommée «fen» et y affiche la page Web «corrigé.html» de la Figure 8.1;
- Le code «`fen.focus()`» permet d'afficher la page Web en premier plan ;
- Avec les navigateurs qui disposent d'onglets, la nouvelle fenêtre va s'afficher dans un nouvel onglet.

Test d'auto-évaluation

Nom : Classe :

Questionnaire sous forme de QCM. Pour chaque question, une seule réponse est juste. Chaque bonne réponse vaut 1 point, chaque mauvaise -0,5 point.

Question 1 : Un réseau de ville est encore appelé :

- LAN.
- MAN.
- WAN.

Question 2 : Une paire torsadée utilise des connecteurs :

- RJ-35.
- BNC.
- RJ-45 répétitives.

Question 3 : La topologie Ethernet est utilisé dans les réseaux :

- En bus.
- Maillés.
- En étoile.
- En anneau.

Figure 8.1 : Page "corrige.html"

3. Les gestionnaires d'événements

On peut associer un ensemble d'instructions ou une fonction à un gestionnaire d'événements. Les évènements ont été étudiés dans la section 6.2 du chapitre 6.

Exemple 8.3 : Intégration de JavaScript par un gestionnaire d'évènement

```
<input type="text" name="nom" onchange=
  "javascript:alert('Bonjour'+
  document.form.qcm.nom.value+ '\n
  Bienvenue sur le site des tests
  d'auto-évaluation'); ">
```

- Le code précédent permet d'exécuter un script après la validation de la saisie dans la zone de texte «Nom»
- Il permet d'afficher un message qui est fonction du texte de la zone de texte «Nom». Si l'utilisateur y insère par exemple le texte «Jean Paul», le script affiche la boîte de dialogue de la Figure 8.2

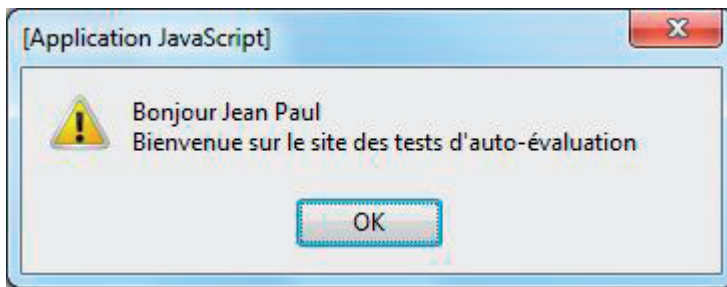


Figure 8.2 : Exemple en mode exécution

Exemple 8.4 : Association d'une fonction nommée «corrige» au clic sur le bouton «Valider»

```
<INPUT TYPE="button" VALUE="Valider"
  onClick="corrige(this.form); ">
```

- Le terme «this.form» permet de passer le formulaire sur lequel on travaille comme paramètre à la fonction «corrige»

II. Intégration à un fichier externe

Il est possible de présenter le code Javascript dans un fichier externe. Ce fichier, rédigé à partir d'un éditeur de texte, doit porter l'extension « .js ». Le script JavaScript est alors appelé dans la page Web à partir de la balise <script> n'importe où dans le code


```

Classe : &nbsp;  <input type="text" name="classe">
</blockquote>

```

<I><H4>Questionnaire sous forme de QCM. Pour chaque question, une seule réponse est juste. Chaque bonne réponse vaut 1 point, chaque mauvaise -0,5 point.</H4></I>

<P>Question 1 : Un réseau de ville est encore appelé :

```

<I> <LI><INPUT TYPE="radio" NAME="question1" VALUE="A">
LAN.</LI>
      <LI><INPUT TYPE="radio" NAME="question1" VALUE="B">
MAN.</LI>
      <LI><INPUT TYPE="radio" NAME="question1" VALUE="C">
WAN.</LI> </I>

```

<P>Question 2 : Une paire torsadée utilise des connecteurs :

```

<I> <LI><INPUT TYPE="radio" NAME="question2"
VALUE="A"> RJ-35.</LI>
      <LI><INPUT TYPE="radio" NAME="question2"
VALUE="B"> BNC.</LI>
      <LI><INPUT TYPE="radio" NAME="question2"
VALUE="C"> RJ-45 répétitives.</LI> </I>

```

<P>Question 3 : La topologie Ethernet est utilisé dans les réseaux :

```

<I> <LI><INPUT TYPE="radio" NAME="question3"
VALUE="A"> En bus.</LI>
      <LI><INPUT TYPE="radio" NAME="question3"
VALUE="B"> Maillés.</LI>
      <LI><INPUT TYPE="radio" NAME="question3"
VALUE="C"> En étoile.</LI>
      <LI><INPUT TYPE="radio" NAME="question3"
VALUE="D"> En anneau.</LI> </I>

```

```
<center><INPUT TYPE="button" VALUE="Valider">
</center>
<A href="javascript:open('corrigé.html');">Voir le cor-
rigé</A>
</FORM>
</blockquote>
</BODY>
</HTML>
```

IV. Code de la fonction «corrige» de l'activité

```
function corrige(form) {
    var rep_bonnes = new Number ();
    var rep_valides = new Number ();

    // Test de la Question 1
    if (form.question1[1].checked) {
        rep_valides=rep_valides+1;
        rep_bonnes=rep_bonnes+1;}
    else{
        if (form.question1[0].checked || form.question1[2].checked) {
            rep_valides=rep_valides+1;}}

    // Test de la Question 2
    if (form.question2[2].checked) {
        rep_valides=rep_valides+1;
        rep_bonnes=rep_bonnes+1;}
    else{
        if (form.question2[0].checked||form.question2[1].checked) {
            rep_valides=rep_valides+1;}}

    // Test de la Question 3
    if (form.question3[2].checked) {
```

```

        rep_valides=rep_valides+1;
        rep_bonnes=rep_bonnes+1;}

    else{
        if (form.question3[0].checked||form.question3[1].checked
        ||form.question3[3].checked){
            rep_valides=rep_valides+1;}}

    alert («Nom : «+form.nom.value+»\n\nClasse : «+form.classe.
    value+»\n\nQuestions validées : «+rep_valides+»\n\nNombre de
    bonnes réponses : «+rep_bonnes+»\n\nNote : «+rep_bonnes/rep_
    valides*20»);
}

```

Commentaires de la fonction «corrige»

- La fonction «corrige» prend en argument «form» qui représente le formulaire de la page Web ;
- Les boutons radio sont des tableaux dont la première valeur est repérée par l'indice 0. Ainsi, la valeur «WAN» de la question1 est stockée dans question1[2] ;
- Pour la question 1, la bonne réponse est MAN, qui est la deuxième valeur. Ainsi, on teste si cette option a été cochée (form.question1[1].checked). Si c'est le cas, on ajoute 1 au nombre de réponses valides (c'est-à-dire cochées) ;
- Si la bonne réponse n'a pas été cochée, on vérifie qu'on a tout de même coché une des deux autres réponses (form.question1[0].checked || form.question1[2].checked), auquel cas on ajoute 1 aux réponses valides ;
- La fonction «alert» affiche une boîte de dialogue similaire à celle de la Figure 8.3.

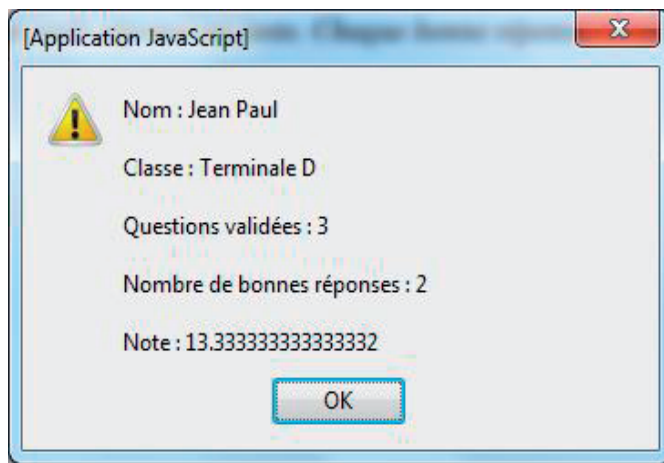
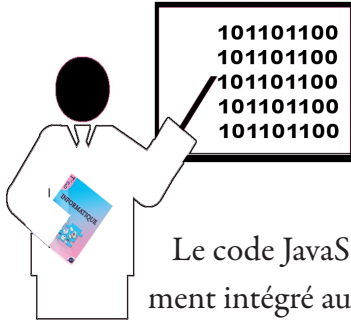


Figure 8.3 : Code de la fonction «corrige» en exécution



RESUME

Le code JavaScript peut se présenter de deux manières dans une page Web : directement intégré au code HTML ou intégré par le biais d'un fichier externe :

- Les balises `<script type="text/javascript"></script>`, les pseudo-URL et les gestionnaires d'évènements permettent d'intégrer le code JavaScript dans du code HTML.
- L'intégration par le biais d'un fichier externe se fait à partir de la balise `<script type="text/JavaScript" src="monscript.js"> </script>`.

Le navigateur interprète le code HTML de haut en bas. Ainsi, si le script est placé dans le «`head`», il sera exécuté avant l'affichage de la page. Le code JavaScript peut également être placé entre les balises `<body >` et `</body >` du code HTML.

EXERCICE
THEORIQUE**EXERCICE 1 : Répondre aux questions suivantes**

- 1) Donner la structure d'un programme HTML.
- 2) Peut-on imbriquer d'autres balises dans une balise html ? Donner un exemple.
- 3) Dire si le langage HTML est sensible à la casse. En est-il de même pour le langage JavaScript ?
- 4) Présenter deux façons différentes d'intégrer un script JavaScript dans un programme HTML.
- 5) Quels sont les outils logiciels nécessaires à l'écriture et à l'exécution des programmes en HTML et JavaScript ?
- 6) Quelle est l'extension d'un document en HTML et en JavaScript ?

EXERCICE
THEORIQUE**EXERCICE 2 : répondre par vrai ou faux**

1. Il est possible d'intégrer un code html au code JavaScript.
2. Le code JavaScript peut se placer n'importe où dans le code HTML.
3. Le navigateur interprète le code HTML de haut en bas.
4. Si le code javascript est placé dans le «head», il sera exécuté avant l'affichage de la page.
5. La syntaxe suivante permet d'intégrer un script JavaScript : `<SCRIPT "language = javascript"> Code du script ;</SCRIPT>`

EXERCICE
PRACTIQUE**EXERCICE 3 : Intégration d'un script JavaScript à une page Web**

Intégrer le script JavaScript (a) au code de la page html (b) :

a) Script JavaScript :

```
var texte1 = new String (" BIENVENUE ");  
var texte2 = new String (" AU LYCEE BILINGUE DU CAMEROUN ");  
var texte = new String ( ) ;  
texte = texte1 + texte2 ;  
alert( texte ) ;  
alert (" c'est super le javascript " ) ;
```

b) Code html

```

<html>
  <head>
    <title> Bienvenue Camerounais Lamda </title>
    -----
    -----
    -----
    -----
    -----
    -----
    -----
  </head>
  <body>
    Ereuka
  </body>
</html>

```



EXERCICE 4 : Intégration du code JavaScript

1. Donner le code de la page Web intégrant le code JavaScript de l'exercice 6 du chapitre précédent.
2. Modifier le code obtenu pour qu'il affiche à l'écran le contenu des variables : nombre, nombre1, nombre2.



EXERCICE 5 : Intégration du code JavaScript

1. Ecrire une fonction qui prend en entrée les notes d'un élève de terminale en une séquence et calcule sa moyenne ;
2. Ecrire un script qui permet à un utilisateur d'entrer au clavier les différentes notes ;
3. Modifier le script précédent en y insérant la fonction de la question 1 pour calculer la moyenne
4. Intégrer ce code à une page Web qui liste les matières au programme en classe de terminale.

PROJET DE CONSOLIDATION 2

L'objectif de ce projet est de mettre en place un site Web interactif disponible depuis n'importe lequel des ordinateurs du réseau mis en place dans le projet de consolidation 1. Le site est une ressource éducative pour les élèves du Lycée Bilingue du Cameroun. Lesdits élèves peuvent accéder à des QCM de Mathématiques, de Sciences de la Vie et de la Terre et d'Informatique.

Le site Web contient huit pages :

- *La page "index.html"* : c'est la page d'accueil du site. Elle est présentée en activité du chapitre 4. Elle contient un lien "Accéder aux QCM" qui permet d'accéder à la page "acces_qcm.html" ;
- *La page "acces_qcm.html"* : cette page contient des liens hypertextes vers les pages de QCM. Elle est similaire à la page "index.html" (positionnement des blocs) ;
- *Les pages de QCM* ("maths.html", "svt.html", "info.html") : elles fonctionnent sur le modèle de la page Web de l'activité du chapitre 6, sauf que les zones "Nom" et "Classe" ne sont plus présentes. Elles contiennent un lien "Corrigé" qui permet d'accéder à la page de corrigé du QCM ;
- *Les pages de corrigé* ("cor_maths.html", "cor_svt.html", "cor_info.html"). Elles proposent le corrigé des QCM.

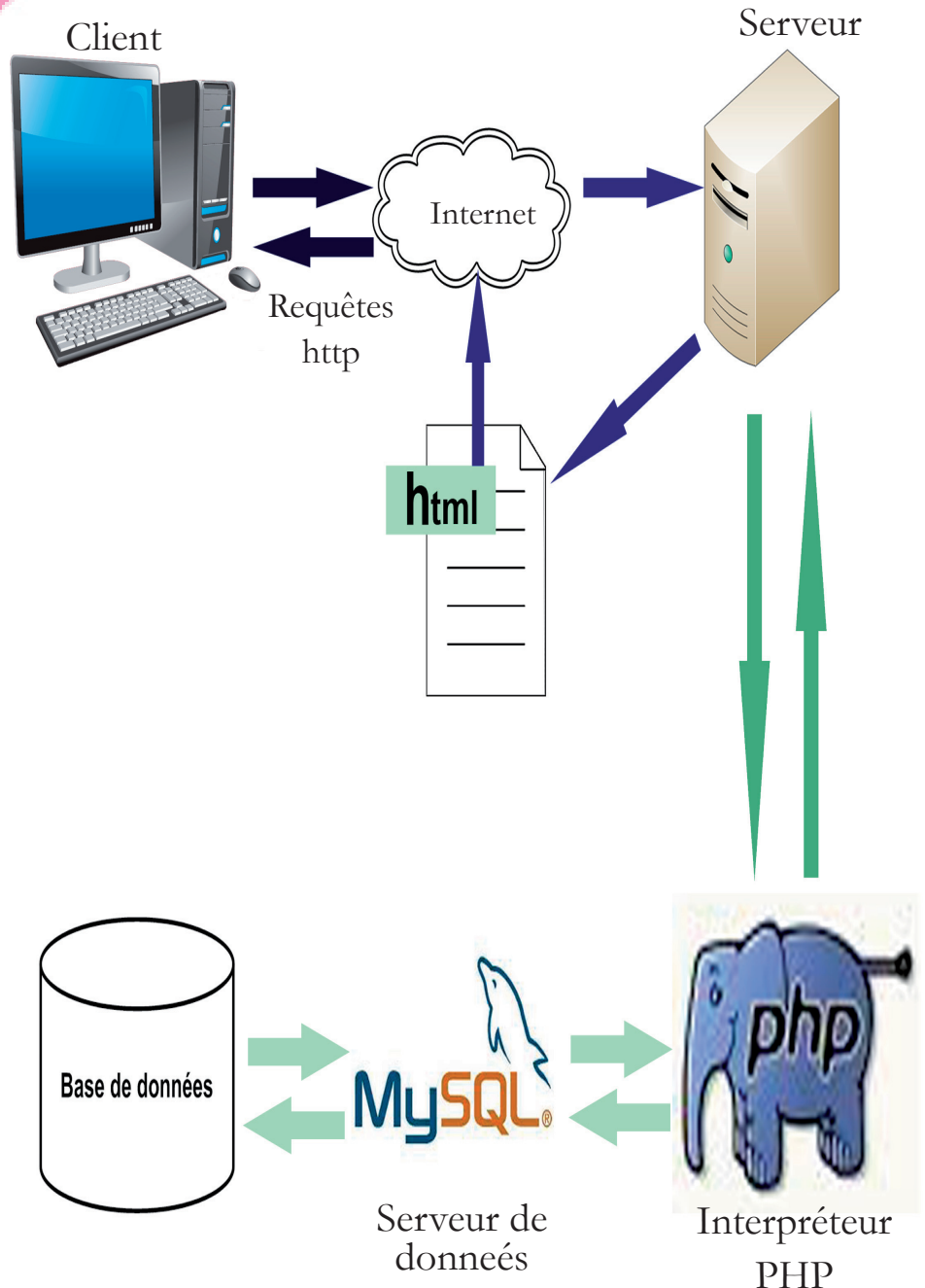
Dans un groupe de quatre à cinq élèves, réaliser les opérations suivantes :

1. Configurer un serveur HTTP (par exemple "apache") sur l'ordinateur serveur du réseau ;
2. Rédiger le code HTML de chacune des pages Web et les tester dans votre navigateur (disposition du texte, des éléments CSS et fonctionnement des liens) ;
3. Dans un fichier externe avec l'extension ".js", écrire le code des fonctions "maths_corrigé", "svt_corrigé" et "info_corrigé" qui permettent de corriger chacun des QCM de mathématiques, de SVT et d'informatique.
4. Intégrer les fonctions "maths_corrigé", "svt_corrigé" et "info_corrigé" dans les codes HTML appropriés ;
5. Copier toutes les pages Web, la feuille de style et le fichier externe créés dans un dossier nommé "www.eleveslbc.com" et copier ce dossier dans le répertoire "www" du serveur HTTP ;
6. Tester le fonctionnement du site sur l'ordinateur serveur en tapant dans la barre d'adresse de votre navigateur "http://localhost/www.eleveslbc.com" ;
7. Tester le fonctionnement du site sur les autres ordinateurs en tapant "http://adresseIP/www.eleveslbc.com" où adresseIP est l'adresse IP de l'ordinateur serveur.

Module 3

PROGRAMMATION DYNAMIQUE DE PAGES Web

Lorsqu'on demande une page Web sur Internet, notre ordinateur (client) envoie au serveur Web une requête HTTP. Le serveur rassemble les éléments de la page Web et vérifie s'il faut des données supplémentaires. Le cas échéant, il va chercher ces éléments dans une base de données. Pour ce faire, il envoie une requête vers un serveur de données (MySQL), notamment à travers un script (php). Le serveur de données interroge la base de données (requêtes SQL), extrait les données cherchées, les renvoie au serveur qui termine de rassembler les éléments de la page Web, et ensuite les retourne au client sous forme de document HTML. Ainsi le statique devient dynamique.



9

INTRODUCTION AUX BASES DE DONNEES

SOMMAIRE

- ▶ Terminologie des bases de données
- ▶ Notion de SGBD

MOTS-CLES

- Base de données
- Table
- Champ
- Enregistrement
- SGBD

ORACLE®

ELEVE			
Matricule	Nom	Date_Nais	Quartier
E0001	YAGOYA Francis	05/04/1992	Mendong
E0002	BERTOYA Sophie	24/09/1994	Iouvence
E0003	KROU Adèle	30/12/1994	Mendong
E0004	MBOUDA Martin	18/07/1991	Obili
E0005	EBOLWA Sidonie	01/02/1993	Nsam
E0006	OBALA Quentin	08/08/1995	Efoulan
E0007	NKONGOA Gaston	08/09/1992	Mendong
E0008	KUMBA Marie-Ange	11/12/1993	Nsam
E0009	AYOS Salomé	14/03/1995	
E0010	LAUM Edwige	10/06/1993	Tam-Tam
E0011	TIND Prince	29/05/1995	Nsimoyong
E0012	BAFANI Christelle	04/08/1993	Melen

NOTE		
Matricule	CodeMat	Note
E0001	C01	
E0002	C03	
E0002	C04	
E0003	C01	09,75
E0003	C02	17,50
E0003	C05	08,75
E0005	C01	13,00
E0005	C02	10,50
E0006	C05	14,00
E0008	C01	12,00
E0010	C01	09,00

MySQL®

PostgreSQL



MATIERE	
CodeMat	NomMat
C01	Mathématiques
C02	Français
C03	Histoire
C04	EPS
C05	Informatique

Microsoft®
SQL Server™

dBASE™

OBJECTIFS

- Définir : Base de donnée, table, champ, enregistrement, SGBD
- Donner la différence entre un fichier et une base de données
- Donner les types de base de données et de SGDB
- Donner les principales fonctions d'un SGBD
- Lister les principaux SGBD

Dans un lycée, on gère dans un registre les adhésions des élèves aux différents clubs. Le tableau ci-après représente ledit registre :

NOM	DATE_NAIS	QUARTIER	CLUB
YAGOUA François	05/04/1992	Mendong	Mathématiques
KRIBI Adèle	30/12/1994	Mendong	Informatique
LOUM Edwige	10/06/1993	Tam-Tam	Mathématiques
KRIBI Adèle	30/12/1994	Mendong	Mathématiques
EBOLOWA Sidonie	01/02/1993	Nsam	Français
KUMBA Marie-Ange	11/12/1993	Nsam	Mathématiques
TIKO Prince	29/05/1995	Nsimeyong	Français
LOUM Edwige	10/06/1993	Tam-Tam	Français
KRIBI Adèle	30/12/1994	Mendong	Français
BERTOUA Sophie	24/09/1994	Jouvence	EPS

1. Combien de fois retrouve-t-on le mot «Mathématiques» ?
2. On souhaite changer le mot «Français» par «Langues». Combien de fois devra-t-on répéter l'opération ?
3. On souhaite supprimer l'élève «KRIBI Adèle». Combien de lignes doit-on supprimer ?
4. Comment peut-on éviter ces redondances ?

On se propose désormais d'utiliser les deux tableaux suivants :

NOM	DATE_NAIS	QUARTIER
YAGOUA François	05/04/1992	Mendong
KRIBI Adèle	30/12/1994	Mendong
LOUM Edwige	10/06/1993	Tam-Tam
KRIBI Adèle	30/12/1994	Mendong
EBOLOWA Sidonie	01/02/1993	Nsam
KUMBA Marie-Ange	11/12/1993	Nsam
TIKO Prince	29/05/1995	Nsimeyong
LOUM Edwige	10/06/1993	Tam-Tam
KRIBI Adèle	30/12/1994	Mendong
BERTOUA Sophie	24/09/1994	Jouvence

CLUB
Mathématiques
Français
Histoire
Informatique
EPS

5. Reprendre les questions 1 à 3.
6. Comment peut-on relier les élèves à un club ?
7. Comment appelle-t-on ce type d'organisation des données?

I. Terminologie des bases de données

Dans une base de données, les informations sont regroupées par thèmes ou sujets. Ces sujets ont chacun un certain nombre de propriétés et plusieurs occurrences (élément particulier d'un sujet). Exemple : On peut considérer comme sujet les élèves. Leurs propriétés sont : matricule, nom, date de naissance, quartier, etc. Une occurrence est un élève particulier.

Base de données

Ensemble de données composé de plusieurs fichiers organisés de manière à éviter des redondances et permettant d'offrir une vue partielle ou totale des données selon les utilisateurs ;



1. Définition des concepts

- **Table** : Ensemble de données organisées suivant un modèle en colonnes (correspondant aux propriétés du sujet) et en lignes (correspondant aux occurrences du sujet) ;
- **Champ ou attribut** : Propriété d'un sujet, colonne d'une table. Les valeurs d'un champ peuvent être générées automatiquement, saisies librement ou choisies dans une liste ;
- **Enregistrement** : Occurrence d'un sujet, ligne d'une table ;
- **Clé primaire** : Champ dont les valeurs permettent d'identifier de manière unique un enregistrement dans une table. Il ne peut être vide ;
- **Relation** : Association de deux tables par la mise en commun d'un champ ;
- **Clé étrangère** : Clé primaire d'une table qui est attribut dans une autre table et qui marque la relation qui lie les deux tables (voir champ « matricule » de la table « NOTE »)

Champs ou attributs	
CodeMat	NomMat
C01	Mathématiques
C02	Français
C03	Histoire
C04	EPS
C05	Informatique

Clé Primaire

Enregistrements

Figure 9.1 : Structure de la table «Matière»

2. Différences entre un fichier et une base de données

Dans un fichier, les informations sont stockées de manière séquentielle (les uns à la suite des autres). Cette méthode de stockage est très différente de celle des bases de données. On peut citer comme différences entre un fichier et une base de données:

- *Le stockage interne* : une base de données est composée de plusieurs fichiers virtuels appelé *tables* ;
- *La méthode de stockage* : un fichier est très lié aux programmes qui permettent de le manipuler tandis qu'une base de données peut être manipulée par divers programmes;
- *L'absence de redondance* : dans une base de données, la même information est stockée une et une seule fois ;
- *La recherche* : pour retrouver une information dans un fichier, il faut parcourir toutes celles qui la précèdent tandis qu'on peut accéder directement à n'importe quelle information stockée dans une base de données.

3. Type de base de données

On distingue plusieurs types de base de données :

- *Hiérarchiques ou réseaux* : ici les enregistrements sont liés les uns aux autres ;
- *Relationnelles* : ici les tables sont reliées les unes aux autres par des champs communs. Ce sont les plus courantes ;
- *Objet* : les bases de données objet découlent des concepts de programmation orientée-objet ;



DEFINITION

SGBD :

Ensemble logiciel destiné à manipuler les données et objets d'une base de données, tout en cachant la complexité des manipulations des structures internes.

II. Notion de Système de Gestion de Bases de Données (SGBD)

Pour manipuler les bases de données, on utilise généralement un logiciel spécialisé appelé SGBD (Système de Gestion de Bases de Données).

Il y a parfois confusion, par abus de langage, entre base de données et SGBD. Le SGBD est un logiciel intermédiaire entre les utilisateurs (par le biais des programmes qui accèdent aux données) et la base de données proprement dite. En cachant les structures de base, le SGBD met à la disposition de l'utilisateur une vue synthétique du contenu de la base de données.

On appelle « *système d'information* » l'ensemble composé par les bases de données, le SGBD utilisé et les programmes associés.

1. Fonctions d'un SGBD

Il existe plusieurs SGBD sur le marché, mais tous (ou presque) remplissent les fonctions suivantes :

- **La définition des données** : elle permet la définition et la mise à jour des structures de la base (tables, champs, etc). Elle est réalisée par un ensemble de commandes appelé *Langage de Définition des Données (LDD)* ;
- **La manipulation des données** : elle permet de manipuler les enregistrements de la base et de les mettre à jour. Elle est réalisée par un ensemble de commandes appelé *Langage de Manipulation des Données (LMD)*. Une partie du LMD est le Langage d'Interrogation des Données (LID) qui permet de rechercher des informations en interrogeant la base ;
- **Le contrôle de l'accès aux données** : il permet de définir des droits d'accès aux différents utilisateurs de la base. Il est réalisé par un ensemble de commandes appelé *Langage de Contrôle des Données (LCD)*.

Chacune des opérations du SGBD doit garantir l'indépendance des données. En effet, les données sont hiérarchisées suivant trois niveaux :

- **Le niveau externe** : A ce niveau les données sont présentés sous forme de vues (table virtuelle correspondant à la copie d'une partie des données de la base) ;



REMARQUE

Les SGBD les plus connus sont payants (Oracle, SQLServer, Access). Toutefois, MySQL, qui est le SGBD le plus utilisé pour les applications Web est libre (sous licence OpenSource)

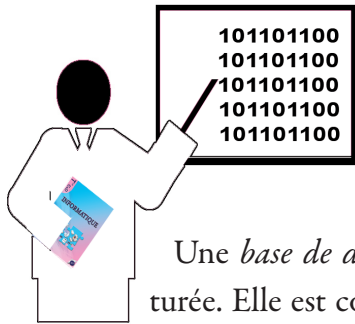
- *Le niveau conceptuel* : A ce niveau on organise la base de données en tables, champs, relations, etc.
- *Le niveau physique* : A ce niveau les données sont enregistrées et consultées sous forme d'enregistrement dans des fichiers

L'indépendance des données est la capacité du SGBD à permettre la modification de l'un des niveaux sans en affecter les autres.

2. Types de SGBD

Le type des SGBD est lié au type de bases de données qu'ils manipulent. Toutefois, dans la grande majorité les SGBD sont relationnels. De par leur fonctionnement on peut distinguer :

- *Les services* (logiciels qui fournissent des données sans nécessiter chez le client l'installation préalable d'un certain nombre de modules). C'est le cas de MySQL, SQL Server, PostgreSQL, Oracle Database ;
- *Les logiciels applicatifs* : ils nécessitent une installation complète pour utiliser la base de données : Paradox ;
- *Les LAG* : ce sont des logiciels applicatifs qui combinent gestion de base de données et programmation : Microsoft Access, Progress, dBase.



RESUME

Une *base de données* désigne un ensemble des données stockées de manière structurée. Elle est composée d'un ensemble de *tables* comportant des colonnes appelées *champs* et des lignes appelées *enregistrements*. Elles diffèrent des fichiers informatiques en termes de stockage, de rédundances et de méthode de recherche.

Pour manipuler ces données, on utilise un *SGBD* (Système de Gestion de Bases de Données) qui est un logiciel complexe. Les SGBD permettent de définir la structure de la base (table, champs, ...), de manipuler les enregistrements et de contrôler les accès à la base. Ils peuvent être *hiérarchiques ou réseau, objet*, mais sont pour la plupart *relationnels*. Ils permettent de garantir l'indépendance entre les données et les traitements. L'ensemble composé par les données, le SGBD et les programmes associés constitue le *système d'information*.

EXERCICE
THEORIQUE**Exercice 1 : Répondre aux questions suivantes**

1. Définir les termes suivants : bases de données, table, clé primaire, relation, clé étrangère, SGBD.
2. Différencier une base de données d'un fichier.
3. Différencier une base de données d'un SGBD.
4. Citer les types de base de données.
5. Donner les trois fonctions d'un SGBD.
6. Citer trois types de SGBD en donnant des exemples.

EXERCICE
THEORIQUE**Exercice 2 : Répondre par vrai ou faux**

1. Une redondance est la répétition de la valeur d'un champ sur plusieurs enregistrements.
2. Toute table porte un nom et possède des champs.
3. Chaque champ d'une table a un type de valeurs qu'il peut prendre (numériques, texte, date, etc.).
4. Toute table ne possède pas forcément une clé primaire.
5. Une colonne relative à une clé primaire ne peut recevoir la valeur vide.

EXERCICE
THEORIQUE**Exercice 3 : Compléter les phrases à trous par les mots de la liste suivante : Indépendance des données LDD, SGBD, LCD, base de donnée, LMD**

1. La définition et la mise à jour des structures de la base sont réalisées grâce au _____.
2. La manipulation des enregistrements de la base et de la mise à jour des données sont réalisées grâce au _____.
3. La définition des droits d'accès aux différents utilisateurs de la base est réalisée grâce au _____.
4. La capacité du SGBD à permettre la modification de l'un des niveaux sans en affecter les autres s'appelle _____.
5. La création et la manipulation des données de la _____ est réalisée grâce à un langage de script compréhensible par les _____.



Exercice 4 : Faire correspondre les éléments de la colonne de gauche aux éléments de la colonne de droite :

SGDB
Base de données
Logiciel
Clé
Identifiant

SGDB
Base de données
Logiciel
Clé
Identifiant

10

LE LANGAGE SQL

SOMMAIRE

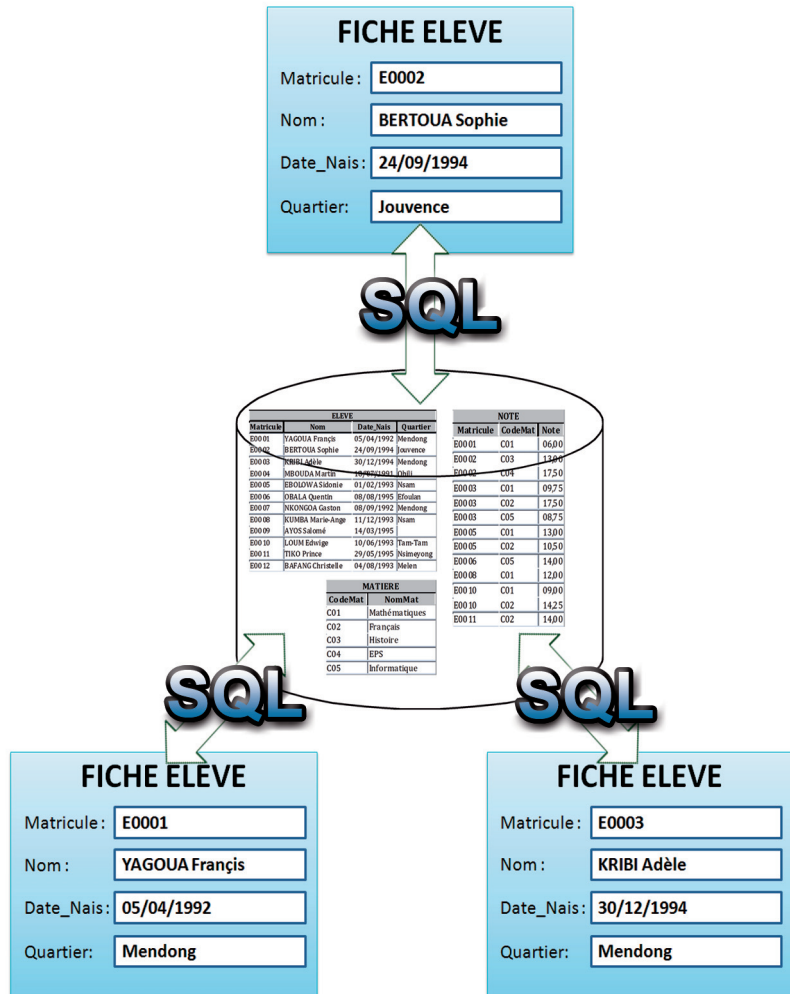
- La définition des données
- La manipulation des données
- La sélection des données

MOTS-CLES

- SQL
- Requêtes
- CREATE
- DROP
- ALTER
- INSERT
- UPDATE
- SELECT

PRE-REQUIS

- Base de données
- Table
- Champ
- Enregistrement
- Clé primaire
- Clé étrangère



OBJECTIFS

- Définir SQL et donner son rôle ;
- Ecrire des requêtes SQL de définitions de données (clauses CREATE, DROP, ALTER)
- Ecrire les requêtes de manipulations d'enregistrements (clauses INSERT, UPDATE)
- Ecrire les requêtes de sélection de données (clause SELECT)
- Utiliser les clauses WHERE et ORDER BY dans un SELECT

On gère les notes de élèves d'une classe lors d'une séquence à partir d'une base de données composée des trois tables données par les tables ci-après :

ELEVE			
Matricule	Nom	Date_Nais	Quartier
E0001	YAGOUA François	05/04/1992	Mendong
E0002	BERTOUA Sophie	24/09/1994	Jouvence
E0003	KRIBI Adèle	30/12/1994	Mendong
E0004	MBOUDA Martin	18/07/1991	Obili
E0005	EBOLOWA Sidonie	01/02/1993	Nsam
E0006	OBALA Quentin	08/08/1995	Efoulan
E0007	NKONGOA Gaston	08/09/1992	Mendong
E0008	KUMBA Marie-Ange	11/12/1993	Nsam
E0009	AYOS Salomé	14/03/1995	
E0010	LOUM Edwige	10/06/1993	Tam-Tam
E0011	TIKO Prince	29/05/1995	Nsimeyong
E0012	BAFANG Christelle	04/08/1993	Melen

NOTE		
Matricule	CodeMat	Note
E0001	C01	06,00
E0002	C03	13,00
E0002	C04	17,50
E0003	C01	09,75
E0003	C02	17,50
E0003	C05	08,75
E0005	C01	13,00
E0005	C02	10,50
E0006	C05	14,00
E0008	C01	12,00
E0010	C01	09,00
E0010	C02	14,25
E0011	C02	14,00

MATIERE	
CodeMat	NomMat
C01	Mathématiques
C02	Français
C03	Histoire
C04	EPS
C05	Informatique

1. Donner pour chaque table : les champs, la clé primaire et éventuellement la clé étrangère ;
2. Formuler des instructions pour créer la table ELEVE ;
3. Formuler des instructions pour ajouter un champ «Redoublant» à la table ELEVE ;
4. Formuler des instructions pour ajouter un élève ;
5. Formuler des instructions pour avoir la liste des élèves ;
6. Formuler des instructions pour avoir la liste des élèves qui habitent Mendong ;
7. Formuler des instructions pour avoir la liste des élèves par ordre alphabétique ;
8. Formuler des instructions pour avoir les notes de l'élève «BERTOUA Sophie» ;
9. Comment appelle-t-on un ensemble d'instructions destinés à réaliser des actions ?
10. Classer les instructions énoncées par fonction de SGBD.

SQL (Structured Query Language) est un langage de commandes conçu en 1970 par la société IBM. Il permet la manipulation de la structure et des données des bases de données. SQL est présent dans presque tous les SGBD et est de ce fait un standard en termes de création et d'administration des bases de données.

Une commande SQL est appelé « requête ». Les requêtes sont regroupées en LDD, LMD et LCD. Si les requêtes de LDD et de LCD sont gérées par le créateur de la base de données, seules les requêtes de manipulation de données sont employées par les utilisateurs

I. Définition de données

SQL permet la définition et la mise à jour (modification et suppression) des tables et des champs.

1. Définition de bases de données

La syntaxe de création d'une base de données est :

```
CREATE DATABASE Nom ;
```

Exemple 10.1 : Création d'une base de données

```
CREATE DATABASE Exemple ;
```

La syntaxe de suppression d'une base de données est :

```
DROP DATABASE Nom ;
```

Exemple 10.2 : Suppression d'une base de données

```
DROP DATABASE Exemple ;
```

2. Définition de tables

a. Création de tables

La création d'une table se fait en définissant le nom de la table ainsi la liste de ces attributs et des types de données. Les principaux types de données sont présentés dans le tableau ci-après :



REMARQUE

Syntaxe SQL :

SQL n'est pas sensible à la casse sauf pour ce qui est des accents. On préférera donc écrire sans accent. Toutefois, il est utile de mettre en majuscule les MOTS-C LE du langage. Par contre, *toute instruction SQL se termine par un point-virgule.*

Tableau 10.1 : Types de données en SQL

TYPES DE DONNEES	SYNTAXE	DESCRIPTION
Données alphanumériques	CHAR(n)	Chaîne de caractères de longueur fixe d'au plus 255 caractères. CHAR(20) désigne un texte d'au plus 20 caractères. Les valeurs sont stockées en ajoutant, s'il le faut, des espaces.
	VARCHAR(n)	Chaîne de caractères de longueur variable. Maximum 65 535
	TEXT(n)	Permet de stocker du texte de longueur variable. Une variable de type TEXT peut avoir du texte sur plusieurs lignes
Données numériques	INTEGER	Nombre entier
	FLOAT	Nombre à virgule
Date	DATE	Date dans le format 'AAAA-MM-JJ'. Exemple : '2012-02-29'



REMARQUE

Création de table

Chaque ligne de définition d'un champ se termine par une **virgule**. Le mot-clé **CONSTRAINT** permet de définir des **clés primaires et étrangères**.

La syntaxe de création d'une table est la suivante :

```
CREATE TABLE Nom_Table (
Champ1 Type1 [contrainte] ,
Champ1 Type1 [contrainte] ,
Champ1 Type1 [contrainte] ,
..... .
Champ1 Type1 [contrainte]
CONSTRAINT Nom Type_Contrainte ) ;
```

Exemple 10.3 : Création de la table «Matiere»

```
CREATE TABLE Matiere (
CodeMat VARCHAR(3) ,
NomMat VARCHAR(20) NOT NULL ,
CONSTRAINT PRIMARY KEY(CodeMat)
) ;
```

- NOT NULL est un attribut de définition de table qui précise que le champ doit toujours être renseigné. Dans notre exemple, une matière aura toujours un nom ;
- La contrainte PRIMARY KEY permet de définir le champ «CodeMat» comme clé primaire.

Exemple 10.4 : Création de la table Eleve

```
CREATE TABLE Eleve (
  Matricule VARCHAR(5),
  Nom VARCHAR(40) NOT NULL UNIQUE,
  Date_nais DATE ,
  Quartier VARCHAR(20) NULL),
  CONSTRAINT PRIMARY KEY(Matricule)
);
```

- **UNIQUE** signifie que le champ ne comportera pas de doublons. Deux élèves n'auront pas le même nom ;
- **PRIMARY KEY** réunit les attributs **NOT NULL** et **UNIQUE**
- **NULL** signifie dans notre exemple qu'on peut enregistrer un élève sans connaître son nom. C'est la valeur par défaut pour un champ.

Exemple 10.5 : Création de la table Note

```
CREATE TABLE Note (
  Matricule VARCHAR(5),
  CodeMat VARCHAR(3) ,
  Note FLOAT ,
  CONSTRAINT
  PRIMARY KEY (Matricule,CodeMat),
  FOREIGN KEY(Matricule) REFERENCES
    Eleve(Matricule) ,
  FOREIGN KEY(CodeMat) REFERENCES
    Matiere(Codemat)
);
```

- **PRIMARY KEY** définit le couple (Matricule,CodeMat) comme clé primaire. Ceci signifie qu'un élève donné a une seule note dans une matière donnée ;
- **FOREIGN KEY** définit «Matricule» comme clé étrangère issue du champ «Matricule» de la table «Eleve». D'où le code «REFERENCES Eleve(Matricule)». Il en est de même du champ «CodeMat»



REMARQUE

Types de Contraintes :

La contrainte **PRIMARY KEY** peut être définie en même temps que le champ (*inline*) mais il est recommandé de le faire après la définition des champs (*out-of-line*). **NOT NULL** et **UNIQUE** sont *inline* mais **FOREIGN KEY** quant à elle doit impérativement être *out-of-line*.

b. Modification des tables

Les syntaxes de modification d'une table sont résumées dans le tableau ci-après :

Tableau 10.2 : Types de modification de table

Type de modification	Syntaxe	Commentaire
Ajouter un champ	<code>ALTER TABLE Nom_Table ADD Champ Type ;</code>	Permet d'ajouter un champ à une table
Supprimer un champ	<code>ALTER TABLE Nom_Table DROP COLUMN Champ ;</code>	Permet de supprimer un champ
Renommer un champ	<code>ALTER TABLE Nom_Table CHANGE Ancien_champ Nouveau_Champ Nouveau_Type;</code>	Permet de changer le nom d'un champ. Elle combine la suppression et l'ajout d'un champ
Modifier un champ	<code>ALTER TABLE Nom_Table MODIFY Champ Nouveau_Type ;</code>	Permet de modifier le type d'un champ



REMARQUE

Script SQL

On peut combiner plusieurs requêtes pour réaliser une action (création et remplissage de table par exemple). Lorsque c'est le cas, l'ensemble de ces requêtes est enregistré dans un fichier appelé **script SQL** portant l'extension «.sql». Les requêtes d'un script sont exécutées de haut en bas.

Exemple 10.6 : Script SQL de modification de la table Eleve

```
ALTER TABLE Eleve
ADD Genre VARCHAR(10);
Alter TABLE Eleve
CHANGE Genre Sexe VARCHAR (10);
ALTER TABLE Eleve
MODIFY Quartier VARCHAR(30);
ALTER TABLE Eleve
DROP COLUMN Sexe ;
```

- La première requête ajoute un champ «Genre» à «Eleve» ;
- La deuxième renomme le champ «Genre» en «Sexe» ;
- La troisième fait passer la taille du champ «Quartier» de 20 à 30 ;
- La quatrième supprime le champ «Sexe». Lorsqu'on supprime un champ, toutes les données qu'elle contient seront également supprimées.

c. Suppression de tables

Pour supprimer une table on utilise la syntaxe suivante :

```
DROP TABLE Nom_table ;
```

Exemple 10.7: Suppression de la table Eleve

```
DROP TABLE Eleve ;
```

II. Manipulations des données

Dans cette section, nous parlerons essentiellement des requêtes SQL qui permettent d'insérer et de mettre à jour des enregistrements.

1. Insertion d'un enregistrement

Pour insérer des enregistrements à une table, on utilise la syntaxe ci-après :

```
INSERT INTO Nom_table [(champ 1,...,champ  
n)] VALUES (valeur 1,...,valeurN)
```

Exemple 10.8 : Insertion d'un enregistrement à la table «Eleve»

```
INSERT INTO Eleve (Matricule, Nom, Date_  
Nais, Quartier)  
VALUES ("E0001" , "Yagoua Francis" ,  
"1992/04/05" , "Mendong") ;
```

- Il est possible d'indiquer une partie seulement des champs (Matricule, Nom, Quartier, etc.) ;
- La liste des champs est facultative. Lorsqu'on ne la donne pas, cela correspond à indiquer tous les champs. On pouvait alors écrire « `INSERT INTO Eleve VALUES ('E0001', 'Yagoua Francis', '1992/04/05', 'Mendong')` ; »
- L'ordre des valeurs doit correspondre à l'ordre des champs qui a été donné ;
- Les dates sont considérées comme des textes. Les valeurs de types dates doivent donc être mises entre apostrophes.



REMARQUE

UPDATE vs REPLACE

Avec UPDATE, on utilise la clause WHERE qui permet d'indiquer la clé primaire de l'enregistrement qui être modifié. Avec REPLACE, cette clé primaire est indiquée comme première valeur.

De plus, UPDATE permet de modifier plusieurs enregistrements à la fois tandis que REPLACE modifie un et seul enregistrement.

2. Modification d'un enregistrement

La modification d'un enregistrement se fait partiellement ou en totalité. Pour modifier partiellement un enregistrement on utilise la syntaxe ci-après :

```
UPDATE Nom_table
SET champ1=valeur1, champ2=valeur2, ... ,
      champN=valeur n
WHERE Condition
```

Pour modifier en totalité on utilise la syntaxe ci-après:

```
REPLACE INTO Nom_table
VALUES (clé_primaire, valeur2..valeurN);
```

Exemple 10.9 : Modification du Quartier de l'élève de matricule «E0009»

```
UPDATE Eleve
SET Quartier="Simbock"
WHERE Matricule="E0009"
```

Exemple 10.10 : Modification de toutes les informations de l'élève de matricule «E0010»

```
REPLACE INTO Eleve
VALUE ("E0010", "OTELE Gaston",
      "1993/05/06", "TKC") ;
```

3. Suppression d'un enregistrement

Pour supprimer un enregistrement on utilise la syntaxe ci-après :

```
DELETE FROM Table
WHERE Condition
```

Exemple 10.11 : Suppression d'enregistrements

```
DELETE FROM Eleve
WHERE Quartier="Mendong" ;
```

III. Interrogation des données

La sélection d'enregistrements est la requête la plus employée avec SQL. Elle permet de rechercher des informations à partir d'une ou de plusieurs tables, suivant d'éventuels critères. Elle retourne une valeur ou une pseudo-table, appelée dans le jargon de programmation «jeu d'enregistrement ou recordset».

Sa syntaxe est :

```
SELECT [ALL] | [DISTINCT] [attributs] | *
FROM liste_table
[WHERE] Condition1
[GROUP BY] liste_champ
[HAVING] condition2
[ORDER BY] liste_champ [ASC] | [DESC] ;
```

Une requête SELECT suit la procédure suivante :

- On choisit les tables avec la clause FROM. Lorsqu'il y en a plus d'une on parle de *jointure* ;
- On filtre les enregistrements avec la clause WHERE : on parle de *restriction* ;
- On regroupe les enregistrements avec la clause GROUP BY : on parle de *regroupement* ;
- On filtre les enregistrements de chaque groupe avec la clause HAVING : on parle de *restriction de groupe* ;
- On trie les enregistrements avec la clause ORDER BY : on parle de *tri* ;
- On sélectionne des colonnes avec la clause SELECT : on parle de *projection*.

L'étude de l'instruction SELECT sera faite à travers des exemples portant sur la base de données «Lycee» présentée en activité.



REMARQUE

Syntaxe du SELECT :

La syntaxe présentée ici est simplifiée. La syntaxe complète, qui dépasse largement le cadre de notre cours, fait l'objet d'une littérature abondante consultable à l'adresse www.sql.developpez.com

1. Projection

Exemple 10.12 : Matricules et noms de tous les élèves

```
SELECT Matricule, Nom
FROM Eleve ;
```

Résultat :

Matricule	Nom
E0001	YAGOUA François
E0002	BERTOUA Sophie
E0003	KRIBI Adèle
E0004	MBOUDA Martin
E0005	EBOLOWA Sidonie
E0006	OBALA Quentin
E0007	NKONGOA Gaston
E0008	KUMBA Marie-Ange
E0009	AYOS Salomé
E0010	LOUM Edwige
E0011	TIKO Prince
E0012	BAFANG Christelle

Exemple 10.13 : Code des matières sur lesquelles on a composé

```
SELECT ALL Codemat
FROM Note ;
```

Résultat :

CodeMat
C01
C03
C04
C01
C02
C05
C01
C02
C05
C01
C01
C02
C02

Exemple 10.14 : Code des matières sur lesquelles on a composé (sans doublons)

```
SELECT DISTINCT Codemat
FROM Note ;
```

Résultat :

CodeMat
C01
C03
C04
C02
C05



REMARQUE

ALL vs DISTINCT

Avec ALL, chaque valeur de champ apparaît autant de fois que sur la table originelle : on parle de doublons. Avec DISTINCT, chaque valeur de champ apparaît une seule fois, même si elle est présente plus d'une fois dans la table originelle.

2. Restriction

Avec la clause WHERE, il est possible de définir des conditions de sélection d'enregistrements. On utilise pour cela :

- Des opérateurs de comparaisons (=, <>) ;
 - Des opérateurs logiques (<, >, <=, >=, AND, OR, NOT)
 - Des opérateurs intégrés (LIKE, BETWEEN, IN, ISNULL).
- Ils seront étudiés à travers les exemples qui suivent.

Exemple 10.15 : Noms des élèves qui habitent Mendong

```
SELECT Nom
FROM Eleve
WHERE Quartier="Mendong" ;
```

Résultat :

Nom
YAGOUA François
KRIBI Adèle
NKONGOA Gaston

Exemple 10.16 : Noms des élèves dont le nom commence par B

```
SELECT Nom
FROM Eleve
WHERE Nom LIKE "B%" ;
```

Résultat :

Nom
BERTOUA Sophie
BAFANG Christelle

Exemple 10.17 : Noms et date de naissance des élèves nés entre 1992 et 1994

```
SELECT Nom, Date_Nais
FROM eleve
WHERE Date_Nais BETWEEN "1992/01/01"
AND "1994/12/31";
```

Résultat :

Nom	Date_Nais
YAGOUA François	05/04/1992
BERTOUA Sophie	24/09/1994
KRIBI Adèle	30/12/1994
EBOLOWA Sidonic	01/02/1993
NKONGOA Gaston	08/09/1992
KUMBA Marie-Ange	11/12/1993
LOUM Edwige	10/06/1993
BAFANG Christelle	04/08/1993



REMARQUE

Plus loin avec la clause WHERE :

Les conditions de la clause WHERE peuvent être utilisés dans les instructions UPDATE et DELETE pour sélectionner les enregistrements sur lesquelles portent la même modification ou la suppression.

3. Regroupement

Le regroupement permet de réunir en une seule ligne des enregistrements qui ont en commun une même valeur de champ. (Exemple : élèves qui habitent le quartier Mendong). Les regroupements sont très utiles, notamment pour établir des statistiques.

L'instruction GROUP BY s'utilise avec des fonctions de groupe ou avec la clause HAVING.

a. Les fonctions de groupe

Ce sont des fonctions qui permettent d'effectuer des opérations sur un ensemble de valeurs d'un champ. Elle sont déclarées dans la clause SELECT ou dans la clause HAVING. Les plus courantes sont résumées dans le tableau ci-après :

Tableau 10.3 : Fonctions de groupe dans SQL

Type de fonction	Commentaires	Exemple d'utilisation
SUM	Permet de sommer un ensemble de valeurs <i>numériques</i>	SUM(Note)
AVG	Permet de calculer la moyenne d'un ensemble de valeurs <i>numériques</i>	AVG(Note)
COUNT	Compte le nombre de valeur d'un champ	COUNT(Matricule)
MAX	Renvoie le maximum d'une liste de valeurs <i>numériques</i>	MAX(Note)
MIN	Renvoie le minimum d'une liste de valeurs <i>numériques</i>	MIN(Note)

Exemple 10.18 : Moyenne des notes obtenues par matière

```
SELECT CodeMat, AVG(Note)
FROM Note
GROUP BY CodeMat ;
```

Résultat :

CodeMat	AVG(Note)
C01	9.45
C02	12.9375
C03	13
C04	17.5
C05	11.375



REMARQUE

Fonctions de groupe ou fonctions de tables

Lorsque SELECT est combiné avec GROUP BY, les fonctions s'appliquent aux enregistrements d'un même groupe. Sans GROUP BY, elles s'appliquent à tous les enregistrements de la table.

Exemple 10.19 : Meilleure note par matière

```
SELECT CodeMat, MAX(Note)
FROM Note
GROUP BY CodeMat ;
```

Résultat :

CodeMat	MAX(Note)
C01	12
C02	17.5
C03	13
C04	17.5
C05	14

Exemple 10.20 : Nombre d'élèves par quartier

```
SELECT Quartier, COUNT(Matricule)
FROM Eleve
GROUP BY Quartier ;
```

Résultat :

Quartier	COUNT(Matricule)
NULL	1
Efoulan	1
Jouvence	1
Melen	1
Mendong	3
Nsam	2
Nsimeyong	1
Obili	1
Tam-Tam	1

b. La clause HAVING

La clause HAVING fonctionne comme la clause WHERE, mais pour les éléments d'un même groupe.

Exemple 10.21 : Code des matières pour lesquelles au moins deux élèves ont composé

```
SELECT CodeMat
FROM Note
GROUP BY CodeMat
HAVING COUNT(Note) >=2 ;
```

Résultat :

CodeMat
C01
C02
C05

4. Tri

Exemple 10.22 : Noms et quartier des élèves triés par ordre croissant des quartiers et décroissant des noms

```
SELECT Nom, Quartier
FROM eleve
ORDER BY Quartier ASC, Nom DESC;
```

Résultat :

Nom	Quartier
AYOS Salomé	NULL
OBALA Quentin	Efoulan
BERTOUA Sophie	Jouvence
BAFANG Christelle	Melen
<i>YAGOUA François</i>	<i>Mendong</i>
<i>NKONGOA Gaston</i>	<i>Mendong</i>
<i>KRIBI Adèle</i>	<i>Mendong</i>
<i>KUMBA Marie-Ange</i>	<i>Nsam</i>
<i>EBOLOWA Sidonie</i>	<i>Nsam</i>
TIKO Prince	Nsimeyong
MBOUDA Martin	Obili
LOUM Edwige	Tam-Tam

- La requête permet de classer les élèves par ordre de quartier. la valeur NULL, qui représente un manque d'informations, est le premier par ordre alphabétique (croissant ou ASC) ;
- Pour les élèves de même quartier, on classe par ordre alphabétique inverse (décroissant ou DESC)

5. Jointure

Exemple 10.23 : Noms, matière et notes des élèves

```
SELECT Nom, NomMat, Note
FROM Eleve, Matiere, Note
WHERE (Eleve.Matricule = Note.Matricule) AND (Matiere.Codemat = Note.Codemat)
ORDER BY Nom ASC, NomMat DESC ;
```

- La requête commence par la projection de champs situés dans tables différentes (Nom de la table Eleve, NomMat de la table Matiere et Note de table Note)
- La condition de jointure est matérialisé par la clause WHERE. On s'assure que la matricule de l'élève est bien dans la table NOTE, et que le code de la matière est bien présent dans la table NOTE ;

- Pour un bon rendu visuel, on trie les données par ordre croissant des noms et ensuite par ordre croissant des notes

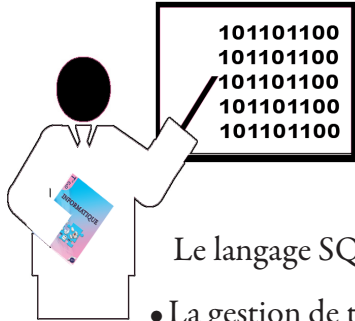
MATIERE	
CodeMat	NomMat
C01 ←	Mathématiques
C02	Français
C03 ←	Histoire
C04 ←	EPS
C05	Informatique

NOTE		
Matricule	CodeMat	Note
E0001	C01	06,00
E0002	C03	13,00
E0002	C04	17,50
E0003	C01	09,75
E0003	C02	17,50
E0003	C05	08,75
E0005	C01	13,00
E0005	C02	10,50
E0006	C05	14,00
E0008	C01	12,00
E0010	C01	09,00
E0010	C02	14,25
E0011	C02	14,00

ELEVE	
Matricule	Nom
E0001	YAGOUA François
E0002	BERTOUA Sophie
E0003	KRIBI Adèle
E0004	MBOUDA Martin
E0005	EBOLOWA Sidonie

Résultat :

Nom	NomMat	Note
BERTOUA Sophie	Histoire	13
BERTOUA Sophie	EPS	17.5
EBOLOWA Sidonie	Mathématiques	10.5
EBOLOWA Sidonie	Français	6
KRIBI Adèle	Mathématiques	9.75
KRIBI Adèle	Informatique	8.75
KRIBI Adèle	Français	17.5
KUMBA Marie-Ange	Mathématiques	12
LOUM Edwige	Mathématiques	9
LOUM Edwige	Français	14.25
OBALA Quentin	Informatique	14
TIKO Prince	Français	14
YAGOUA Francis	Mathématiques	6



RESUME

Le langage SQL permet :

- La gestion de tables (CREATE, DROP, ALTER)
- La manipulation de données (INSERT INTO , UPDATE, REPLACE, et DELETE) ;
- L'interrogation et la recherche dans les tables : SELECT.

Les commandes SQL peuvent être synthétisées par le schéma ci-après :

```
CREATE TABLE <Nom Table>
```

```
( <Liste de champs + Type> )
```

```
( <liste des contraintes> )
```

```
ALTER TABLE <Table>
```

```
( ADD, MODIFY, DROP COLUMN,  
CHANGE )
```

```
( <Liste de champs + Type> )
```

```
DROP <Table>
```

```
INSERT INTO <Table>
```

```
( <liste de champs> )( <liste de valeurs> )
```

```
DELETE FROM <Table>
```

```
WHERE <liste de critères>
```

```
UPDATE <Table> SET < champ=valeur >
```

```
WHERE <liste de critères>
```

```
ALL, DISTINCT
```

```
*
```

```
AVG, MAX, MIN
```

```
SUM, COUNT
```

```
SELECT <liste de champs>
```

```
FROM <liste de tables>
```

```
WHERE <liste de critères>
```

```
ORDER BY <liste de champs>
```

```
GROUP BY <liste de champs> HAVING
```

```
<liste de critères>
```

```
{ =, <, >, <=, >=, <>
```

```
LIKE, IS NULL,  
IN, BETWEEN
```

```
OR, AND, NOT
```

```
{ ASC, DESC
```

EXERCICE
THEORIQUE

Exercice 1 : Répondre aux questions suivantes

1. Définir les termes suivants : SQL, requête, jointure, projection, doublon.
2. Quel est le langage utilisé pour créer et manipuler les données dans un SGBD ?
3. Citer les types de données utilisés en SQL.
4. Quelles sont les commandes SQL qui permettent de définir la structure des éléments d'une base de données ?
5. Donner la syntaxe des commandes SQL permettant de créer et supprimer une base de données
6. Donner la syntaxe des commandes SQL permettant de créer, modifier et supprimer une table d'une base
7. Quelles sont les commandes SQL qui permettent de manipuler les données qui seront stockées ?
8. Donnez la syntaxe des commandes SQL permettant d'insérer, supprimer, modifier et sélectionner les données dans une base de données
9. Donnez 3 exemples de logiciels dans la vie courante utilisant une base de données.

EXERCICE
THEORIQUE

EXERCICE 2 : Répondre par vrai ou faux

1. SQL (Structure Query Language) a été conçu en 1970 par la société IBM.
2. Les requêtes SQL sont regroupées en LDD, LMD et LCD.
3. SQL n'est pas sensible à la casse sauf pour ce qui est des accents.
4. Toute instruction SQL se termine par une virgule.
5. Le mot-clé CONSTRAINT permet de définir des clés primaires et étrangères.
6. On peut combiner plusieurs requêtes dans un script SQL ayant l'extension .sql pour réaliser une action
7. Une projection sélectionne des lignes avec la clause WHERE
8. L'option DISTINCT de la requête SELECT permet d'éliminer les doublons.
9. Une jointure permet de sélectionner les colonnes appartenant à des tables différentes.
10. L'option DESC de la clause ORDER BY de la requête SELECT permet de trier les enregistrements dans l'ordre croissant.



EXERCICE 3 : Visualisation des résultats d'une requête

On donne les tables ci-après d'une base données

NumProf	NomProf	Matière	Grade
P0001	YAGOUA Francis	Mathématiques	PLEG
P0002	BERTOUA Sophie	Français	PLEG
P0003	KRIBI Adèle	PCT	PCEG
P0004	MBOUDA Martin	Informatique	Vacataire
P0005	EBOLOWA Sidonie	Mathématiques	PCEG
P0006	OBALA Quentin	Histoire	PCEG
P0007	NKONGOA Gaston	Informatique	PLEG
P0008	KUMBA Marie Ange	Mathématiques	Vacataire
P0009	AYOS Salomé	Histoire	PLEG
P0010	LOUM Edwige	Français	PLEG
P0011	TIKO Prince	Anglais	PLEG
P0012	BAFANG Christelle	Anglais	PCEG

Table Professeur

NumClasse	NomClasse	Effectif
C01	Sixième	44
C02	Cinquième	61
C03	Quatrième All	87
C04	Quatrième Esp	101

Table Classe

NumProf	NumClasse	Jour
P0001	C03	Jeudi
P0001	C04	Lundi
P0002	C01	Lundi
P0002	C02	Mercredi
P0002	C03	Jeudi
P0003	C03	Mardi
P0003	C04	Lundi
P0004	C01	Mardi
P0004	C04	Mercredi
P0005	C01	Jeudi
P0005	C02	Vendredi
P0006	C01	Mardi
P0006	C03	Jeudi
P0007	C02	Jeudi
P0007	C03	Vendredi
P0008	C01	Mercredi
P0008	C02	Jeudi
P0009	C02	Vendredi
P0009	C03	Vendredi
P0010	C01	Lundi
P0010	C04	Jeudi
P0011	C02	Vendredi
P0011	C03	Mardi

Table Enseigne

1. Pour chacune des tables données, préciser les types des champs, indiquer les clés primaires et éventuellement les clés étrangères
2. Exprimer en langage courant chacune des requêtes ci-après et donner un schéma des résultats :

a) SELECT NomProf, Matière, Grade

FROM Professeur

ORDER BY Grade ;

b) SELECT NomProf, Matière

FROM Professeur

WHERE Grade like "P%" ;

c) SELECT NomClasse

FROM Classe

WHERE Effectif <=100 ;

d) SELECT COUNT(Numprof)

FROM Professeur

GROUP BY Grade ;

e) SELECT NomProf, Jour

FROM Professeur, Enseigne

WHERE Professeur.NumProf = Enseigne.NumProf

ORDER BY NomProf ;

f) SELECT DISTINCT NomClasse

FROM Classe, Enseigne

WHERE Classe.NumClasse = Enseigne.NumClasse

AND Enseigne.Jour="Mardi";

g) SELECT NomProf, NomClasse, Jour

FROM Professeur, Classe, Enseigne

WHERE Professeur.NumProf = Enseigne.NumProf

AND Classe.NumClasse = Enseigne.NumClasse

AND Professeur.Grade = "PLEG";



EXERCICE 4 : Création et manipulations dans une base donnée à l'aide du langage de requêtes SQL

Lors de la rentrée scolaire, on enregistre dans une base de données les différents élèves du Lycée Bilingue du Cameroun dans leurs classes respectives et par la suite, on affecte des professeurs dans ces classes afin d'enseigner les diverses matières au programme. La description de cette base de données est la suivante :

Eleve(Matricule, NomEleve, PrenomEleve, DateNaissance),

Classe(Cycle, NomClasse, NumClasse),

Professeur(CodeProfesseur, NomProfesseur, PrenomProfesseur),

Enseignement(NomClasse, CodeProfesseur, Annee).

1. Proposer un nom à cette base de données. Quels sont ses différentes tables ?
2. Donner la(ou les) clé primaire(s) de chaque table et la clé étrangère si elle existe.
3. Proposer un type de données pour chaque champ des différentes tables.
4. Donner les requêtes SQL de création de la base de données et des différentes tables.
5. L'on a oublié la matière du professeur dans la structure de la base. Donner la requête SQL pour ajouter le champ matière.
6. Donner la requête d'insertion de l'élève : "E001, Tamo, Maxime, 15-11-1987"
7. Donner la requête d'insertion de cet élève dans sa classe : "2nd cycle, Terminale, C01"
8. L'élève Tamo est plutôt né en 1993. Donner la requête SQL de modification de sa date de naissance
9. Donner la requête SQL suivante : Quel est le matricule, le nom et le prénom des élèves inscrit au second cycle ?
10. Donner la requête SQL suivante: Quel est le nom et la matière des professeurs de la classe de Terminale C.

11

INTRODUCTION A PHP

SOMMAIRE

- ▶ Qu'est-ce que PHP ?
- ▶ Syntaxe de PHP
- ▶ Intégration de PHP à HTML

MOTS-CLES

- PHP
- Opensource
- Portabilité
- Echo

PRE-REQUIS

- Langage JavaScript

```
<html>
<head>
</head>
<body>
<h2>Voici un script PHP</h2>
<?php
  echo"Texte à afficher dans
    la page Web";
  echo"<br/>";
  echo"Liste
    numérotée<br/>";
  echo"<ul>";
  echo"<li>Processeur</li>";
  echo"<li>RAM</li>";
  echo"<li>Disque dur</li>";
  echo"<li>Ecran</li>";
  echo"</ul>";
?>
</body>
</html>
```

Code HTML avant l'exécution du script PHP

Voici un script PHP

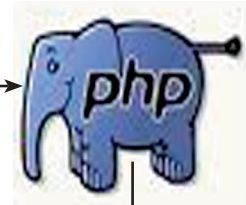
```
Texte à afficher dans la page Web
Liste numérotée


- Processeur
- RAM
- Disque dur
- Ecran

```

Page Web dans le navigateur

Interpréteur PHP



```
<html>
<head>
</head>
<body>
<h2>Voici un script PHP</h2>
Texte à afficher dans la
page Web<br/>Liste numérot
ée<br/><ul><li>Processeur<
/li><li>RAM</li><li>Disque
dur</li><li>Ecran</li></ul>
</body>
</html>
```

Code HTML après l'exécution du script PHP

OBJECTIFS

- Donner l'importance de PHP
- Utiliser les éléments du langage PHP

Dans le site du Lycée Bilingue du Cameroun, on crée une page Web pour chaque élève. Dans une autre page Web, on liste les noms des élèves sous la forme de liens qui permettent d'afficher les pages Web des élèves. Ainsi, en cliquant sur le texte "Mamamou Bineta" on accède à la page Web de l'élève nommé "Mamadou Bineta". Après quelques temps, on se retrouve avec un millier d'élèves.

1. Combien de pages Web doit-on créer ?
2. Combien de liens hypertexte aura la page Web qui liste les élèves ?
3. Comment faire pour retrouver l'élève "Mamadou Bineta" qui est le 500^{ème} élève ?

On souhaite pouvoir retrouver tous les élèves qui se prénomme "Nina" ?

4. Quelle solution faut-il mettre en place ?
5. Quelle procédure permet de récupérer les données ainsi stockées ?
6. Comment peut-on relier les pages Web aux données stockées ?
7. Combien de pages Web aura-t-on si on adopte cette solution ?

I. Qu'est-ce que PHP ?

1. Origines de PHP

PHP est un langage de script qui a été créé en 1995 par Rasmus LERDORF. Au départ, son objectif était de permettre à son auteur de savoir qui venait consulter son CV sur son site personnel. Il s'appelait alors « Personal Home Page ». Par la suite, afin de pouvoir communiquer avec les bases de données et créer des applications dynamiques pour le Web, LERDORF travailla à nouveau son langage qui devint PHP/FI (PHP Form Interpreter). La mise à disposition du code à la communauté des développeurs a permis de considérablement faire avancer le projet PHP qui produira tour à tour :

- PHP/FI 2.0 en 1997 ;
- PHP 3.0 en 1998, PHP signifiait désormais « *PHP Hypertext Preprocessor* » ;
- PHP 4.0 en 2000 ;
- Et PHP 5.0 en 2003 ;

PHP continue d'évoluer. On annonce d'ailleurs la prochaine sortie de PHP 6.0. PHP est présent sur une grande majorité de sites dynamiques de renom au nombre desquels « Yahoo ! », « Voilà », « Le monde ».

2. Intérêt de PHP

PHP est un langage de script qui s'exécute coté serveur. Ainsi, PHP permet de créer des pages Web dynamiques dans lesquelles un utilisateur pourra :

- S'enregistrer à un site ;
- Consulter des pages répondant à ces besoins ;
- Envoyer ou recevoir des mails sans passer par un serveur de messagerie ;

- Créer un blog ;
- Créer un site de diffusion ;
- Créer un site de e-commerce ;
- Etc.

PHP présente un certain nombre d'avantages qui le positionne devant les autres langages de script coté serveur tels que ASP.NET ou Java :

- *PHP est « opensource »* : il est donc gratuit, complètement modifiable et redistribuable ;
- *PHP est portable* : un script PHP marchera aussi bien avec un système d'exploitation « Windows » qu'avec des systèmes Unix, Mac OS ou AS400 ;
- *PHP s'interface avec la plupart des SGBD* : Microsoft SQL Server, Oracle, PostgreSQL, Sybase, SQLite, FilePro, Informix, Interbase, mSQL, dBase, Empress, et bien d'autres. Très souvent, PHP est associé à MySQL ;
- *PHP est abondamment commenté* : il existe sur Internet un grand nombre de sites d'informations et de conseils sur PHP. De plus, une large communauté de développeurs aide au quotidien les apprenants comme les professionnels sur divers forums.

3. Fonctionnement de PHP

Quand un fichier contenant des scripts PHP est demandé au serveur Web, ce dernier le charge en mémoire et y cherche tous les scripts PHP qu'il transmet à l'interpréteur PHP. L'interpréteur accède quelque fois à un serveur de données et exécute le script, ce qui a pour effet de produire du code HTML qui vient remplacer le script PHP dans le document finalement fourni au navigateur. Ce dernier reçoit donc du HTML « pur » et ne voit jamais la moindre instruction PHP. Le fonctionnement de PHP est résumé par la Figure 11.1 ci-après :

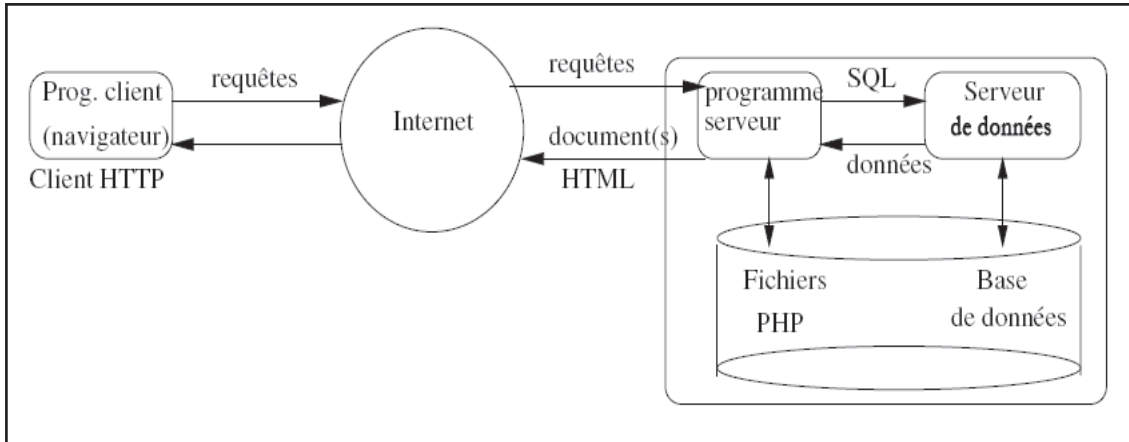


Figure 11.1 : Fonctionnement de PHP

II. Syntaxe de PHP

1. Les variables en PHP

Les variables sont des contenants qui vont stocker des valeurs. Les noms des variables sont toujours précédés du caractère \$ et suivis du nom choisi pour identifier la variable, qui ne doit comporter que des caractères alphanumériques (sauf le premier caractère, qui ne doit pas être un chiffre) ou le caractère souligné _ :

Exemple : « \$mvariable » et « \$Ma variable » ne sont pas des noms de variables valides.

Les types de variables en PHP sont pratiquement les mêmes qu'en JavaScript (String, Number, Boolean, Array, Object). Toutefois, avec PHP, on ne précise pas explicitement le type de variable, mais il est déterminé par l'interpréteur lorsqu'on affecte une valeur à la variable.

Exemple 11.1: Utilisation de variables en PHP

```
$mvariable=10 ;
$mvariable=10.5 ;
$montexte= "Bonjour " ;
```

- \$mvariable est d'abord un entier, ensuite un réel;
- \$montexte est un texte.

2. Opérations sur les variables

Les opérations en PHP sont pratiquement les mêmes qu'en JavaScript :

Tableau 10.1 Opérations en PHP

Type d'opération	Opérations
Arithmétique	+, -, *, /, ++, --, %(modulo)
Comparaison	==, <, >, <=, >=, != ou <>
Logique	OR ou , AND ou &&, !
Affectation	=
Concaténation	.

Exemple 11.2 : Opérations élémentaires en PHP

```
$nombre1=17;
$nombre1 = (nombre1+8) / 3 ;
$nombre1 ++ ;
$nombre1 = $nombre1%3 ;
$nombre2 = (6<=2) OR (3<4) ;
$texte1="Bonjour " ;
$texte1=$texte1."monsieur." ;
```

- \$nombre1 est un entier de valeur 17 ;
- Dans \$nombre1 on affecte le résultat de l'opération «(17+8)/3» c'est-à-dire 8.33333. \$nombre1 devient un réel ;
- \$nombre1 est incrémenté et vaut 9.33333 ;
- Dans \$nombre1 on affecte le résultat de l'opération «9.33333%3» c'est-à-dire 3. \$nombre1 est un entier ;
- Dans \$nombre2 on affecte le résultat de l'opération «(6<=2) OR(3<4)» c'est-à-dire «true». \$nombre2 est un booléen ;
- \$texte1 est de type «string» et vaut «Bonjour » ;
- On concatène le texte «Bonjour » et le texte «Monsieur.» \$texte1 vaut donc «Bonjour Monsieur.»

3. Utilisation des structures de contrôle

Les structures de contrôle en PHP sont les mêmes qu'en JavaScript :

Tableau 10.2 : Structures de contrôle en PHP

Type d'opération	Opérations
Bloc d'instruction	{ et }
Structure alternative	If ... Else
Itérations	While, Do...While, For

Exemple 11.2 : Utilisation d'une structure alternative en PHP

```
if ( $sexe=="masculin" ) {
    echo "Bounjour Monsieur" ;
}
else {
    echo "Bonjour Madame" ;
}
```

Exemple 11.3 : Utilisation d'une boucle While en PHP

```
$chiffre = 1 ;
$somme=0;
while ( $chiffre < 5 ) {
    $somme = $somme + chiffre;
    $chiffre = $chiffre + 1;
}
```

Exemple 11.4 : Utilisation d'une boucle Do ...While en PHP

```
chiffre = 0 ;
$somme=0;
do {
    $chiffre = $chiffre + 1 ;
    $somme = $somme + $chiffre;
} while ( chiffre < 5 )
```

Exemple 11.5 : Utilisation d'une boucle For en PHP

```
$somme = 0 ;
for ( $chiffre = 1 ; $chiffre <= 5 ;
    $chiffre = $chiffre + 1 ) {
```

```

    $somme = $somme + $chiffre ;
}

```

4. Utilisation des tableaux

Les tableaux en PHP sont déclarés de la même manière qu'en JavaScript.

Exemple 11.6 : Déclaration de tableau en PHP

```

tableau1 = Array( 12, 13, 24, 86) ;
tableau2 = ["lundi", "Mardi", "Jeudi"];

```

L'accès aux éléments du tableau est également pareil qu'en JavaScript.

Exemple 11.7 : Accès aux éléments d'un tableau PHP

```

$montableau[0]= "Xaverie" ;
$montableau[1]= "Robert" ;
$montableau[2]= "Alain" ;
$montableau[3]= "Zoé" ;
$montableau[4]= "Martin" ;
$montableau[5]= "Sophie" ;
$montableau[3]= "Balbine" ;
sort($montableau);
echo "Voici la liste des prénoms par ordre
      alphabétique" ;
for (indice=0;indice<length(montableau);
     indice=indice+1) {
    echo "$montableau[indice]" ;
}

```

PHP dispose d'un certain nombre de variables prédéfinies de type tableau. Les principales sont présentées dans le tableau ci-après :

Tableau 10.3 : Variables prédéfinies en PHP

Variable	Observation
\$_COOKIE	Contient le nom et la valeur des cookies enregistrés sur le poste client

<code>\$_GET</code>	Contient le nom et la valeur des données issues d'un formulaire envoyé par la méthode GET
<code>\$_POST</code>	Contient le nom et la valeur des données issues d'un formulaire envoyé par la méthode POST

5. Utilisation des fonctions

Les fonctions en PHP sont déclarées et appelées de la même manière qu'en JavaScript.

Exemple 11.8 : Déclaration et appel de fonctions en PHP

```
function addition($a,$b) {  
    $c=$a+$b;  
    return $c;  
}  
function danger() {  
    echo "fonction danger exécutée";  
}  
$somme = addition(2,3);  
danger();
```

- «addition» est une fonction qui retourne la somme des valeurs des variables «\$a» et «\$b» ;
- «danger» est une procédure qui écrit sur la page Web le texte «fonction danger exécutée» ;
- Dans la variable «\$somme» on affecte la somme des valeurs 2 et 3, c'est-à-dire 5 ;

6. L'instruction echo

L'instruction «echo» permet d'écrire dans le document final le contenu qui la suit. On peut renvoyer par l'instruction «echo» :

- Du texte ;
- La valeur d'une variable ;
- Le résultat d'un calcul ;
- Le résultat d'une fonction ;

- Un code HTML ;
- Un script Javascript ;

Exemple 11.9 : Utilisation de l'instruction "echo"

```
$a=1;
$b=2;
echo "Bonjour <br/> \n" ;
echo "$a" ;
echo "$a*$b" ;
echo "addition($a,$b) <br/> \n" ;
echo "<b>Voici un texte en gras</b> \n" ;
echo "<script type=\"text/javascript\">
\n";
echo "alert('Voici un script javas-
cript')\n;";
echo «</script>»;
```

- La Figure 11.2 présente un document HTML, après l'interprétation du script PHP de l'Exemple 11.9

```
Bonjour <br/>
123 <br/>
<b>Voici un texte en gras</b>
<script type="text/javascript"
alert('Voici un script javascript');
</script>
```

Figure 11.2 : Document HTML après l'Exemple 10.9

- Le caractère «\n» permet de passer à la ligne dans le document HTML ;
- Les variables et opérations «\$a», «\$a+\$b» et «addition(\$a,\$b)» sont remplacées par leurs valeurs respectives (1, 2 et 3) ;
- La Figure 11.3 présente la page Web, après l'interprétation du document HTML de la Figure 11.2

```
Bonjour
123
Voici un texte en gras
```

Figure 11.3 : Page Web après l'exécution du document de la Figure 10.2

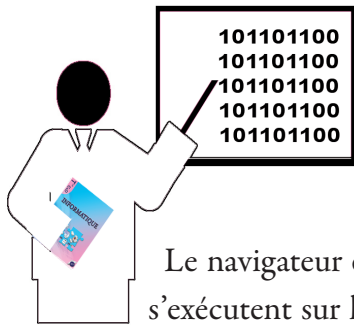
- L'exécution du document HTML de la Figure 10.2 se termine par l'affichage dans une boîte de dialogue du texte «Voici un script javascript».

III. Intégration de PHP à HTML

Le code PHP doit être encadrés par l'une des balises suivantes :

- `<?php et ?>` ;
- `<? et ?>` ;
- `<% et %>` ;
- `<script language="php">` et `</script>`

Seul ce qui est entre ces balises est interprété comme du PHP. Le reste est envoyé tel quel, comme du code HTML. Il est fortement conseillé d'utiliser uniquement l'ouverture "`<?php`", car elle est la seule vraiment portable sur toutes les configurations.



RESUME

Le navigateur exécute les codes HTML et JavaScript. Les scripts PHP quant à eux, s'exécutent sur le serveur Web grâce à l'interpréteur de script qui transforme le code PHP en code HTML. C'est pourquoi l'on dit que les programmes PHP s'exécutent côté serveur.

Ces scripts PHP permettent de formuler des requêtes auprès du serveur de données répondant aux besoins des utilisateurs. C'est pourquoi on dit qu'il permet de réaliser des pages Web dynamiques (variable d'un utilisateur à un autre) par opposition aux pages Web statiques.

La syntaxe du langage PHP est la même que le langage JavaScript à quelques différences près : le nom des variables sont toujours précédé du caractère dollar (\$).

Les scripts PHP s'intègrent aisément dans le code HTML par l'usage des balises : `<?php ?>`, `<? et ?>`, `<% et %>` ou `<script language="php">` `</script>`



EXERCICE PRATIQUE

EXERCICE 1 : Donner la valeur de chacune des variables pendant et à la fin du script suivant en vérifiant l'évolution du type de ces variables :

```
$x="PHP5";  
$a[0]=$x;  
$y=5;  
$z=$y*10;  
$x=$x+$y;  
$y=$y*$z;  
$a[0]="MySQL";  
$a="0";  
$b="TRUE";  
$c=FALSE;  
$d=($a OR $b);  
$e=($a AND $c);  
$f=($a XOR $b);
```



EXERCICE PRATIQUE

EXERCICE 2 : Structure alternative

Rédiger une expression conditionnelle pour tester si un nombre est à la fois un multiple de 3 et de 5.



EXERCICE PRATIQUE

EXERCICE 3 : Structure répétitive

Rechercher le PGCD (plus grand commun diviseur) de deux nombres a et b donnés.



EXERCICE PRATIQUE

EXERCICE 4 : Fonction

Créer une fonction PHP qui prend en paramètres un texte et l'affiche dans une boîte de dialogue à partir de la fonction JavaScript « alert »



EXERCICE PRATIQUE

EXERCICE 5 : Intégration HTML

Intégrer chacun des scripts PHP des exercices 1 à 4 dans des documents HTML.

12

ACCES AUX DONNEES AVEC PHP

SOMMAIRE

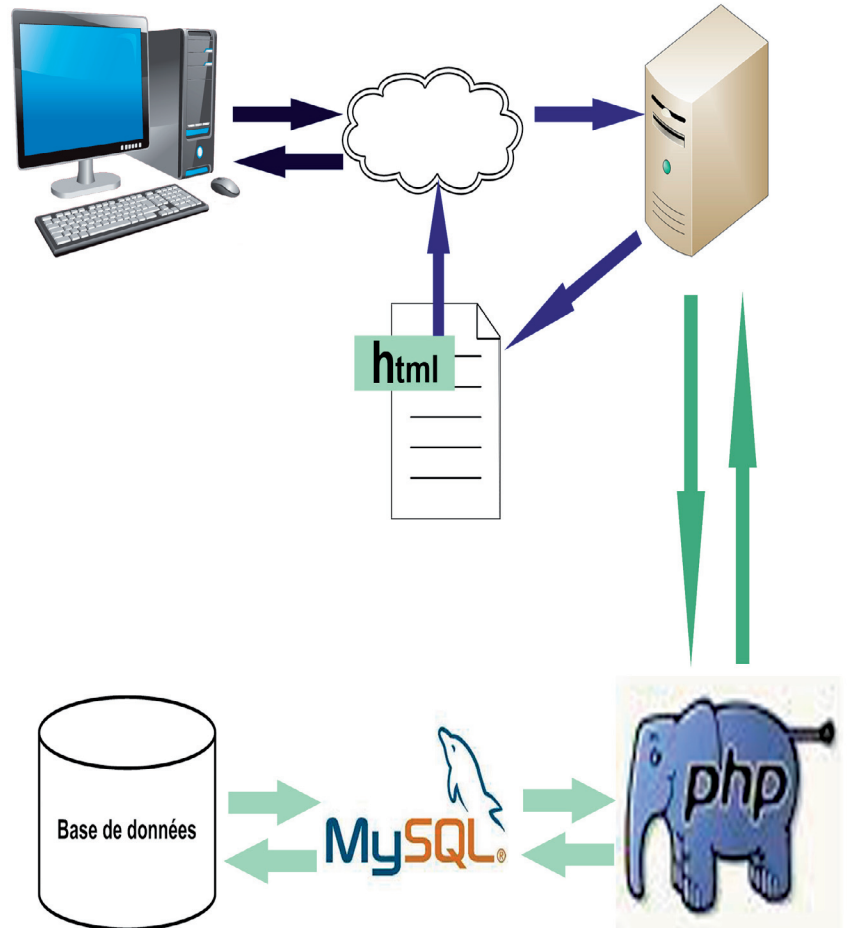
- ▶ Connexion au serveur Web
- ▶ Choix de la base de données
- ▶ Envoi de requêtes
- ▶ Récupération du résultat d'un SELECT
- ▶ Fermeture de la connexion
- ▶ Accès aux éléments d'un formulaire HTML

MOTS-CLES

- Mysql_connect
- Mysql_select_db
- Mysql_query
- Mysql_fetch
- Mysql_close

PRE-REQUIS

- Langage PHP



OBJECTIFS

- Accéder à un serveur Web ;
- Choisir une base de données dans le serveur Web
- Envoyer des requêtes SQL à la base de données
- Récupérer le résultat d'un SELECT
- Manipuler les données d'un formulaire après soumission

Dans un restaurant, un client commande un plat de «poulet à la basquaise accompagné de riz parfumé». La commande du client passe par les étapes ci-après :

- Le serveur transmet la commande au cuisinier ;
- Le cuisinier analyse la commande du client et fait l'inventaire des ingrédients ;
- Le cuisinier envoie son assistant chercher dans le magasin les ingrédients (poulet, tomate, huile, riz, etc.) ;
- Le cuisinier prépare le plat et le transmet au serveur ;
- Le serveur apporte le plat au client qui le déguste.

1. Quel est le rôle du serveur ?
2. Quel est le rôle du cuisinier ?
3. Quel est le rôle de l'assistant cuisinier ?
4. A quoi peut-on assimiler ces concepts ?
5. Donner un algorithme représentant le traitement de la commande du client.

Comme nous l'avons déjà dit, PHP permet d'accéder à diverses bases de données. Toutefois, il est plus souvent associé à MySQL. C'est pourquoi, l'étude de l'accès aux données à travers PHP se fera avec une base de données MySQL nommée «lycee» qui correspond à la base de données de l'activité du chapitre 8.

Quelque soit la base de données utilisée, l'accès aux données suit les étapes ci-dessous :

- **Connexion au serveur Web** : il s'agit de l'ordinateur qui contient la base de données. On utilise la fonction `mysql_connect()` ;
- **Choix de la base de données** : un serveur Web peut contenir plusieurs bases de données. Il faut préciser celle qu'on va utiliser avec la fonction `mysql_select_db()` ;
- **Envoi de requêtes SQL à la base de données** : PHP peut envoyer des requêtes de définition de données, mais il s'agit essentiellement des requêtes de manipulation et d'interrogation de données. On utilise la fonction `mysql_query()` ;
- **Récupération du résultat de la requête** : Les valeurs sont récupérés uniquement en cas de requêtes de sélection de données. On utilise les fonctions `mysql_fetch_row()`, `mysql_fetch_array` ou `mysql_fetch_object`
- **Fermeture de la connexion** : on utilise la fonction `mysql_close()` ;

I. Connexion au serveur Web

La principale fonction qu'on utilise pour accéder au serveur de Web est `mysql_connect()`. Sa syntaxe est la suivante :

```
mysql_connect ($host, $user, $password) ;
```

- `$host` est une chaîne contenant le nom du serveur Web. Ce dernier est défini par l'hébergeur du site. En local, elle correspond à «localhost» ;

- *\$user* est une chaîne contenant le nom sous lequel l'utilisateur est autorisé à accéder au serveur. Il est également fourni par l'hébergeur. En local, elle correspond à «root» ;
- *\$password* est une chaîne contenant le mot de passe associé à l'utilisateur. Si plusieurs utilisateurs sont autorisés à accéder au serveur, chacun doit posséder un identifiant et un mot de passe personnel. En local, elle correspond à une chaîne vide.

Exemple 12.1 : *Connexion à un serveur Web*

- ```
mysql_connect("localhost","root","");
```
- La fonction `mysql_connect()` retourne la valeur «false» lorsque la connexion n'a pas réussi. Aussi, il est recommandé de tester la valeur de retour de `mysql_connect()` pour exécuter des actions lorsque la connexion a réussi et d'autres si ce n'est pas le cas.

Exemple 12.2 : *Test de connexion à un serveur Web*

```
$connexion=mysql_connect("localhost",
 "root","");
if ($connexion==false){
 echo "echec de la connexion";
}
else {
 echo "connexion réussie");
}
```

## II. Choix de la base de données

Si le serveur comporte plusieurs bases de données, le script précise la base désirée au moyen de la fonction `mysql_select_db()`, dont la syntaxe est la suivante :

```
mysql_select_db($nom_base);
```

Cette fonction retourne «true» si la base existe et «false» dans le cas contraire.

Exemple 12.3 : *Test de connexion à une base de données*

```
$bd=mysql_select_db("lycee");
if ($bd==false) {
 echo "connexion à la base réus-
 sie";
}
else {
 echo "echec de la connexion à la
 base");
}
```

### III. Envoi de requêtes SQL

Toute opération à réaliser sur une base nécessite d'envoyer au serveur une requête SQL rédigée à l'aide des commandes détaillées au chapitre 8.

Pour envoyer une requête, on utilise la fonction *mysql\_query()*, dont la syntaxe est la suivante :

**mysql\_query(\$requete) ;**

- La chaîne «\$requete» contient le code de la requête SQL.  
*Elle ne doit pas se terminer par un point-virgule.*
- Les requêtes étant souvent longues, il est recommandé, à des fins de lisibilité du code, de les écrire dans une variable chaîne «\$requete» passée ensuite à la fonction.
- Lorsqu'on passe à la fonction *mysql\_query()* une requête LDD ou LMD, elle retourne une valeur booléenne pour marquer que le requête a abouti ou pas.
- Lorsqu'on passe à la fonction *mysql\_query()* une requête LID, elle renvoie une table virtuelle appelée recordset (jeu d'enregistrement) ou «false» si la requête ne retourne pas de résultat

Exemple 12.4 : *Requête PHP d'insertion d'enregistrement à la base*

```
$requete ="INSERT INTO Eleve(Matricule,
 Nom,Date_Nais,Quartier) VALUES
 ('E0014' , 'BAFOUSSAM Alex' ,
 '1993/07/19' , 'Mendong')";

$test=mysql_query($requete);

if ($test==true){
 echo "La requête a aboutie";
}
else {
 echo "La requête a échouée";
}
```

Exemple 12.5 : *Requête PHP de suppression d'enregistrement à la base*

```
$requete="DELETE FROM Eleve WHERE
 Matricule='E0014';
$test=mysql_query($requete);

if ($test==true){
 echo "La requête a aboutie";
}
else {
 echo "La requête a échouée";
}
```

Exemple 12.6 : *Requête PHP de sélection d'enregistrement à la base*

```
$requete="SELECT * FROM Eleve ;
$test=mysql_query($requete);

if ($test==false){
 echo "La requête a aboutie";
}
else {
 echo "La requête a échouée";
}
```

## IV. Récupération du résultat d'un SELECT

Comme dit précédemment, lorsqu'on passe à la fonction `mysql_query()` une requête de sélection, elle retourne soit une valeur, soit un recordset, c'est-à-dire une table virtuelle contenant les résultats de la requête.

On accède au recordset ligne par ligne :

- Par un tableau : avec la fonction `mysql_fetch_row()` ;
- Par un tableau associatif : avec la fonction `mysql_fetch_array()` ;
- Par un objet : avec la fonction `mysql_fetch_object()`.

Exemple 12.7 : Accès à un recordset par un tableau

```
$requete="SELECT * FROM Eleve";
$recordset=mysql_query($requete);
while ($ligne=mysql_fetch_
 row($recordset)) {
 echo "$ligne[0] :";
 echo "$ligne[1] :";
 echo "$ligne[2] :";
 echo "$ligne[3] :";
 echo "
";
}
```

| ELEVE     |                   |            |           |
|-----------|-------------------|------------|-----------|
| Matricule | Nom               | Date_Nais  | Quartier  |
| E0001     | YAGOUA Francis    | 05/04/1992 | Mendong   |
| E0002     | BERTOUA Sophie    | 24/09/1994 | Jouvence  |
| E0003     | KRIBI Adèle       | 30/12/1994 | Mendong   |
| E0004     | MBOUDA Martin     | 18/07/1991 | Obili     |
| E0005     | EBLOWA Sidonie    | 01/02/1993 | Nsam      |
| E0006     | OBALA Quentin     | 08/08/1995 | Efoulan   |
| E0007     | NKONGOA Gaston    | 08/09/1992 | Mendong   |
| E0008     | KUMBA Marie-Ange  | 11/12/1993 | Nsam      |
| E0009     | AYOS Salomé       | 14/03/1995 |           |
| E0010     | LOUM Edwige       | 10/06/1993 | Tam-Tam   |
| E0011     | TIKO Prince       | 29/05/1995 | Nsimcyong |
| E0012     | BAFANG Christelle | 04/08/1993 | Melen     |

- Lors du premier passage dans la boucle «While», la fonction `mysql_fetch_query($recordset)` se positionne sur la première ligne de la table virtuelle résultat de la requête, et l'affecte à la variable «\$ligne» ;
- Les instructions «echo» permettent d'afficher les valeurs du

|          |                |            |          |
|----------|----------------|------------|----------|
| E0001    | YAGOUA Francis | 05/04/1992 | Mendong  |
| Indice 0 | Indice 1       | Indice 2   | Indice 3 |

tableau recordset suivant les indices (0, 1, 2, et 3), c'est-à-dire respectivement (E0001, YAGOUA Francis, 05/04/1992, Mendong) ;

- Lors du deuxième passage dans la boucle «While», la fonction `mysql_fetch_query($recordset)` se positionne sur la deuxième ligne de la table virtuelle résultat de la requête, et l'affecte à la variable «`$ligne`» ;

|          |                |            |          |
|----------|----------------|------------|----------|
| E0002    | BERTOUA Sophie | 24/09/1994 | Jouvence |
| Indice 0 | Indice 1       | Indice 2   | Indice 3 |

- Les instructions «echo» permettent d'afficher les valeurs du tableau `recordset` suivant les indices (0, 1, 2, et 3), c'est-à-dire respectivement (E0002, BERTOUA Sophie, 24/09/1994, Jouvence) ;
- La boucle se poursuit jusqu'à la dernière ligne.

#### Exemple 12.8 : Accès à un recordset par un tableau associatif

```
$requete="DELETE FROM Eleve WHERE
 Matricule='E0014';
$recordset=mysql_query($requete);
while ($ligne=mysql_fetch_
 array($recordset)) {
 echo "$recordset["Matricule"] :";
 echo "$recordset["Nom"] :";
 echo "$recordset["Date_nais"] :";
 echo "$recordset["Quartier"] :";
 echo "
"
}
```

- Le code s'exécute de la même manière sauf qu'avec la fonction `mysql_fetch_array()`, on doit donner les champs de la requête (ceux qui sont dans le `SELECT`) comme indice de tableau

#### Exemple 12.9 : Accès à un recordset par un objet

```
$requete="DELETE FROM Eleve WHERE
 Matricule='E0014';
$recordset=mysql_query($requete);
while ($ligne=mysql_fetch_
 object($recordset)) {
```

```

echo "$recordset->Matricule :";
echo "$recordset->Nom :";
echo "$recordset->Date_nais :";
echo "$recordset->Quartier :";
echo "
"
}

```

- Le code s'exécute de la même manière qu'avec la fonction `mysql_fetch_array()` ;
- On doit donner les champs de la requête (ceux qui sont dans le SELECT) comme éléments de l'objet (`$recordset->Matricule`)

Quelque soit le type de `mysql_fetch`, les trois exemples ci-dessus produisent le document HTML suivant :

## V. Fermeture de la connexion

```

E0001 :YAGOUA François :05/04/1992 :Mendong

E0002 :BERTOUA Sophie :24/09/1994 :Jouvence

E0003 :KRIBI Adèle :30/12/1994 :Mendong

E0004 :MBOUDA Martin :18/07/1991 :Obili

E0005 :EBLOWA Sidonie :01/02/1993 :Nsam

E0006 :OBALA Quentin :08/08/1995 :Efoulan

E0007 :NKONGOA Gaston :08/09/1992 :Mendong

E0008 :KUMBA Marie-Ange :11/12/1993 :Nsam

E0009 :AYOS Salomé :14/03/1995 : NULL

E0010 :LOUM Edwige :10/06/1993 :Tam-Tam

E0011 :TIKO Prince :29/05/1995 :Nsimeyong

E0012 :BAFANG Christelle :04/08/1993 :Melen


```

Figure 12.1: Document HTML après l'exécution du script de l'Exemple 12.9

Après l'exécution du script PHP, la connexion vers le serveur est automatiquement fermée. Toutefois, il est recommandé de la fermer manuellement. Pour le faire, on utilise la fonction `mysql_close()` dont la syntaxe est la suivante :

```
mysql_close($connexion) ;
```

- «`$connexion`» est la variable dans laquelle on a affecté le résultat de la fonction `mysql_connect()`.

Exemple 12.10 : Fermeture de la connexion au serveur Web

```
mysql_close($connexion) ;
```

## VI. Accès aux éléments d'un formulaire HTML

### 1. Rappel sur les formulaires HTML

Tous les éléments de formulaire doivent être placés entre les balises `<form>` et `</form>`. Ces attributs principaux sont :

- **method** : il s'agit de la méthode d'envoi des éléments du formulaire (get ou post) ;
- **action** : indique l'adresse de la page vers laquelle doit être redirigé le visiteur une fois qu'il aura cliqué sur le bouton d'envoi. Cette page doit être une page de script et donc comporter par exemple l'extension `.php` ou `.asp`, mais pas l'extension `.html` ;
- **name** : c'est le nom du formulaire, il permet d'avoir accès au formulaire dans un script

### 2. Les éléments de formulaire

Tableau 12.1 : Syntaxe des éléments de formulaire HTML

| Eléments                 | Syntaxe                                                                                                                                                                                                                                                                                       | Commentaires                                                                                                                 |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Zone de texte monoligne  | <code>&lt;input type="text" name="texte1" /&gt;</code>                                                                                                                                                                                                                                        | On y accède par l'attribut <code>&lt;&lt;name&gt;</code> . Leur valeur est le texte de leur contenu                          |
| Mot de passe             | <code>&lt;input type="password" name="texte2" /&gt;</code>                                                                                                                                                                                                                                    |                                                                                                                              |
| Zone de texte multiligne | <code>&lt;textarea name="texte3" /&gt;</code>                                                                                                                                                                                                                                                 |                                                                                                                              |
| Case à cocher            | <code>&lt;input type="checkbox" name="item1" checked="checked" /&gt;</code><br><code>&lt;input type="checkbox" name="item2" /&gt;</code>                                                                                                                                                      | Si l'élément 1 est coché, la propriété <code>&lt;&lt;item1.checked&gt;</code> aura la valeur <code>&lt;&lt;true&gt;</code>   |
| Bouton radio             | <code>&lt;input type="radio" name="radio1" value="Valeur1" /&gt;</code><br><code>&lt;input type="radio" name="radio1" value="Valeur2" /&gt;</code>                                                                                                                                            | Si l'élément 1 est coché, la propriété <code>&lt;&lt;radio1.checked&gt;</code> aura la valeur <code>&lt;&lt;true&gt;</code>  |
| Liste déroulante         | <code>&lt;select name="select1" &gt;</code><br><code>&lt;option value="Valeur1"&gt;Valeur1 &lt;/option&gt;</code><br><code>&lt;option value="Valeur2"&gt;Valeur2 &lt;/option&gt;</code><br><code>&lt;option value="Valeur2"&gt;Valeur2 &lt;/option&gt;</code><br><code>&lt;/select&gt;</code> | Si l'élément 1 est coché, la propriété <code>&lt;&lt;Valeur1.checked&gt;</code> aura la valeur <code>&lt;&lt;true&gt;</code> |

### 3. Récupération des éléments du formulaire avec PHP

Lorsqu'on clique sur le bouton d'envoi d'un formulaire, ses éléments sont envoyés dans un tableau associatif qui prend pour indice les valeurs des propriétés «name». Le code HTML de l'exemple 11.11 ci-après est celui de la Figure 12.2 ci-contre

Exemple 12.11 : *Formulaire de connexion à un site*

```
<form name="login" method="post"
 action="connexion.php">
 Login

 <input type="text" name="login"
 size="20" />

 Mot de passe
 <input type="password" name="pwd"
 size="20" />

 <input type="submit"
 value="Valider" />

</form>
```

- L'attribut «method="post"» va nous permettre de récupérer les valeurs des champs «login» et «pwd» dans la variable de type tableau associatif «\$\_POST»;
- Un clic sur le bouton d'envoi ("submit") va provoquer l'ouverture du fichier «connexion.php»

Après un clic sur le bouton d'envoi du formulaire, on exécute le fichier connexion.php. Ce fichier va permettre d'effectuer les opérations suivantes :

- Récupérer les valeurs "login" et "mot de passe" du formulaire (ici le login correspond au matricule de l'élève) ;
- Si le login et le mot de passe correspondent aux données stockées dans la base, on affiche le texte «Bonjour + le nom de l'élève», et on affiche un lien qui permet d'accéder à la zone membre ;

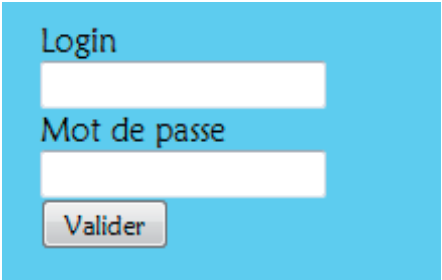
The image shows a simple login form on a light blue background. It consists of two text input fields. The first field is labeled 'Login' and the second is labeled 'Mot de passe'. Below the second field is a button labeled 'Valider'.

Figure 12.2: Formulaire de connexion

- Si le login ou le mot de passe ne correspondent pas, on affiche le texte «Vous n'êtes pas enregistré» et un lien vers une page d'inscription.

Le code de l'Exemple 12.12 ci-après est celui du fichier «connexion.php» qui s'exécute après un clic sur le bouton d'envoi du formulaire

Exemple 12.12 : Code du fichier «connexion.php»

```
<html>
<head>
 <title>Accueil de membre</title>
</head>

<body>
 <?php
 //Récupération des données du formulaire
 $login= $_POST["login"];
 $pwd=$_POST['pwd'];

 //connexion au serveur
 $connexion=mysql_connect("localhost","root","");
 if ($connexion=false) {
 echo "Connexion au serveur impossible !";
 }
 else {

 //choix de la base de données
 $bd=mysql_select_db("lycee");
 if ($bd=false) {
 echo "Connexion à la base de données impos-
sible !";
 }
 }
 }

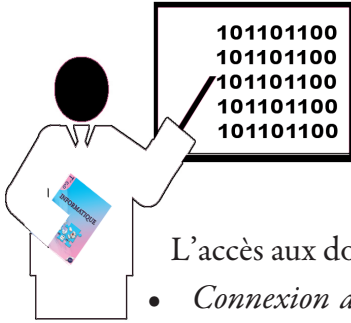
```

```
else {
 //envoi de la requête au serveur
 $requete="SELECT * FROM Eleve ";
 $recordset=mysql_query($requete);

 //Parcours du recordset pour rechercher l'élève
 $nom="";
 While ($ligne=mysql_fetch_object($recordset){
 if ($ligne->Matricule==$login AND $ligne->pwd==$pwd){
 $nom=$ligne->Nom;
 }
 }

 if ($nom!="") {//Visiteur enregistré
 echo "Bonjour $nom
";
 echo "Zone membre";
 }
 else {//Visiteur non enregistré
 echo "Vous n'êtes pas enregistré
";
 echo "Inscrivez-vous";
 }
 } //Fin du if de choix de la base de données
} //Fin du if de connexion au serveur
?>

</body>
```



## RESUME

L'accès aux données d'une base de données suit les étapes ci-dessous :

- *Connexion au serveur Web* : il s'agit de se connecter à l'ordinateur qui héberge la base de données on utilise la fonction `mysql_connect($host,$user,$password)` ;
- *Choix de la base de données* : un serveur Web peut contenir plusieurs bases de données. Il faut préciser celle qu'on va utiliser avec la fonction `mysql_select_db($nom_base)`;
- *Envoi de requêtes SQL à la base de données* : PHP peut envoyer des requêtes de définition de données, mais il s'agit essentiellement des requêtes de manipulation et d'interrogation de données. On utilise la fonction `mysql_query($requete)`;
- *Récupération du résultat de la requête* : Les valeurs sont récupérées uniquement en cas de requêtes de sélection de données. On utilise les fonctions `mysql_fetch_row()`, `mysql_fetch_array` ou `mysql_fetch_object`
- *Fermeture de la connexion* : on utilise la fonction `mysql_close($connexion)`.

La personnalisation de la page Web, en fonction de l'utilisateur est réalisée par la récupération des variables des formulaires HTML



### Exercice 1 : Formulaire HTML et PHP

Soit un document HTML dont le code est donné ci-après :

```
<html>
 <head>
 <title> Insertion des données </title>
 </head>

 <body>
 <form method="post" action="ajout.php" >
 //Suite du formulaire
 </form>
 </body>
</html>
```

1. Le fichier suivant peut-il être enregistré avec l'extension .php ou .html ?
2. Où se fait le traitement des données ?



### EXERCICE 2 : Formulaire HTML et PHP

Créez un formulaire comprenant un groupe de champs ayant pour titre "Adresse client". Le groupe doit permettre la saisie du nom, du prénom, de l'adresse, de la ville et du code postal. Les données sont ensuite traitées par un fichier PHP séparé récupérant les données et les affichant dans un tableau HTML.



### EXERCICE 3 : Formulaire HTML et Base de données

1. Créer un formulaire HTML permettant d'entrer les informations sur un nouvel élève (la structure est celle de la table élève, de la base de données « lycée » qui a été étudiés au chapitre 8) ;
2. Passer le formulaire par la méthode POST
3. Créer un fichier PHP qui s'exécute après la validation du formulaire, récupère les données et insère un nouvel enregistrement dans la table « Eleve »



## PROJET DE CONSOLIDATION 3

L'objectif de ce projet est d'interfacier le site Web du projet de consolidation 2 avec une base de données nommée "eleve". Le nouveau site va permettre aux élèves de s'enregistrer et d'accéder aux pages de QCM. Les élèves non enregistrés n'y auront pas accès. On aura les modifications suivantes :

- La page "index.html" : elle n'aura plus le lien "Accéder aux QCM". Lorsqu'un utilisateur se connecte (en donnant son login et son mot de passe) on ouvre la page "connexion.php" ;
- La page "connexion.php" : elle récupère le login et le mot de passe du formulaire, accède à la base de données pour tester si l'utilisateur est enregistré.
  - S'il est enregistré, on récupère son nom et on affiche le texte "Bonjour Nom" puis un lien hypertexte "Zone membre" qui pointe vers le fichier "acces\_qcm".
  - S'il n'est pas connecté, on affiche le message "Désolé! Vous n'êtes pas enregistré." Ensuite, on affiche deux liens hypertextes permettent de revenir à la page d'accueil (Retour) et d'ouvrir la page d'enregistrement (S'enregistrer). Des représentations de la page "connexion.php" sont données par les figures ci-dessous :

Désolé! Vous n'êtes pas enregistré. <a href="#">S'enregistrer</a> <a href="#">Retour</a>	Bonjour Kribi Adèle. <a href="#">Zone membre</a>
---------------------------------------------------------------------------------------------	-----------------------------------------------------

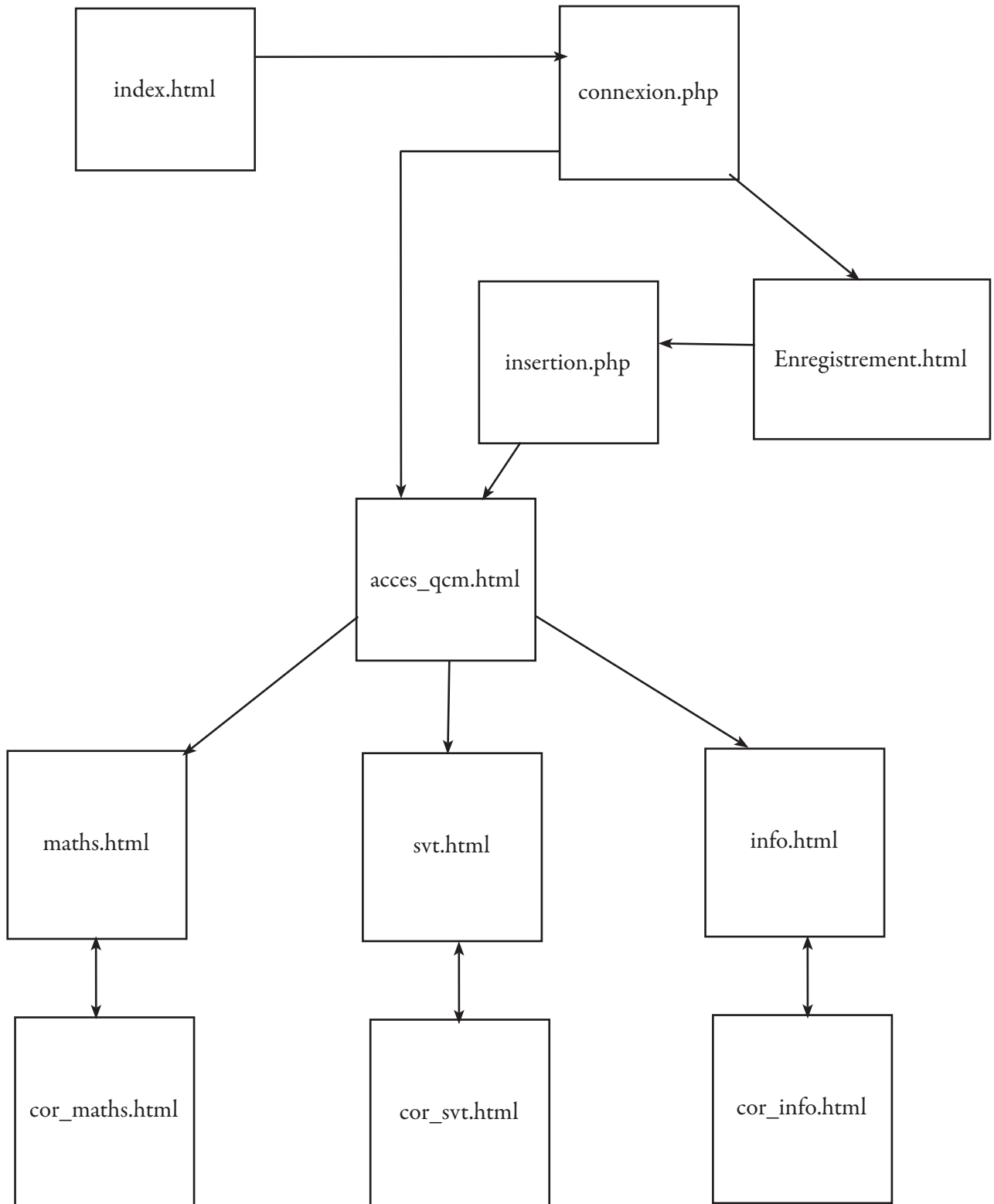
- La page "enregistrement.php" : elle permet à un utilisateur de s'enregistrer en donnant un nom utilisateur (login), un mot de passe et son nom. Le texte du mot de passe devra être masqué. En cliquant sur le bouton "Valider", on exécute le fichier "insertion.php" qui insère les informations de l'utilisateur dans la base de données et affiche un lien hypertexte "Zone membre" qui pointe vers le fichier "acces\_qcm". Une représentation de cette page est donnée par la figure ci-dessous:

Login :	<input type="text"/>
Mot de passe :	<input type="password"/>
Nom :	<input type="text"/>
	<input type="button" value="Valider"/>

Le plan du site est présenté ci-dessous :

# PROJET DE CONSOLIDATION 3

## PLAN DE NAVIGATION DU SITE



# Bibliographie

## Livres

- Danièle DROMARD, Dominique SERET, *Architecture des réseaux*, Pearson Education France, 2009 ;
- Claude SERVIN, *Réseaux et télécoms*, Dunod, 2003 ;
- Fabrice LEMAINQUE, *Tout sur les réseaux sans fil*, Dunod, 2009
- Philippe RIGAUX, *Pratique de MySQL et PHP*, 4<sup>ème</sup> édition, Dunod, 2009
- Jean ENGELS, *PHP5 Cours et exercices*, 2<sup>ème</sup> édition, Eyrolles, 2009
- Christophe PORTENEUVE, *Bien développer pour le Web 2.0*, Eyrolles, 2007
- Christian Wenz, *JavaScript l'essentiel du code et des commandes*, Pearson Education France, 2009
- Christian SOUTOU, *Apprendre SQL avec MySQL*, Eyrolles, 2006
- P. CHALÉAT, D. CHARNAY et J.-R. ROUET, *Les Cahiers du programmeur PHP/MySQL et JavaScript*, Eyrolles, 2005
- Nicolas LARROUSSE, *Création de bases de données*, Pearson Education France, 2009

## Ressources Web

- [www.commentcamarche.net](http://www.commentcamarche.net)
- [www.developpez.com](http://www.developpez.com)
- [www.php.net](http://www.php.net)
- [www.phpdebutant.org](http://www.phpdebutant.org)
- [www.dev.mysql.com](http://www.dev.mysql.com)



# Index

---

---

## A

---

---

Adresse IP  
Adresses privées 45  
Adresses publiques 45  
DCHP 47

---

---

## B

---

---

Base de données  
Attribut. *Voir* Champ  
Champ 133  
Clé étrangère 133  
Clé primaire 133  
Enregistrement 133  
Relation 133  
Système d'information 135  
Table 133

---

---

## C

---

---

Câble réseau  
Câble coaxial 24  
Fibre optique 24  
Paire torsadée 24  
Sertissage 59  
Computer 5  
CSS 71

---

---

## F

---

---

Formulaire HTML  
Input 182  
Select 182

---

---

## H

---

---

Hardware 6  
HTML  
Attribut de balise 72  
Balise 71  
Balises usuelles 72

---

---

## I

---

---

Informatique 5

---

---

## J

---

---

Javascript  
Alert 97  
balise <script> 118  
Confirm 97  
Cookies 100  
Événements 98, 120  
Fichier externe 121  
Fonction 94  
Méthode 95  
Objet 77  
Procédure 94

Prompt 96  
Pseudo-URL 119  
Structures  
    *Do...While* 90  
    *For* 91  
    *If* 88  
    *While* 89  
Tableaux 92  
Variables  
    *Déclaration* 83  
    *Opérations* 85  
    *Utilisation* 85

---

---

## L

---

---

Logiciel 14  
    Logiciels d'application 16  
    Logiciels systèmes 14  
        *BIOS* 14  
        *Systèmes d'exploitation* 15  
Utilitaires 17

---

---

## O

---

---

Ordinateur 5  
    Bus 12  
    Cartes d'extension 12  
    CD, DVD, Clé USB 11  
    Chipset 12  
    Clavier 9  
    Disque dur 11  
    Dossier 14  
    Ecran 10  
    Fichier 13  
    Imprimante 10  
    Modem 12  
    Scanner 9  
    Souris 9  
    Webcam 9

---

---

## P

---

---

PHP  
    \$\_GET 169  
    \$\_POST 169, 183  
    Echo 169  
    Fonctions 169  
    Mysql\_close 181  
    Mysql\_connect 175  
    Mysql\_fetch 179  
    Mysql\_query 177  
    Mysql\_select\_db 176  
    Structures de contrôle 167  
    Tableaux 168  
    variables 165  
  
Protocole  
    HTTP 42  
    IP 42

---

---

## R

---

---

Réseau informatique 23  
    La méthode d'accès 27  
    Les architectures 34  
        *Ad hoc* 35  
        *Client/serveur* 35  
        *Egal à égal* 34  
        *Infrastructure* 35  
    Les messages 25  
    Les périphériques du réseau 25  
    Les supports de transmission 24  
    Nature de la liaison 33

---

---

## S

---

---

SGBD  
    Indépendance des données 136  
    LCD 135  
    LDD 135  
    LID 135  
    LMD 135  
  
Site Web 71

## SQL

ALTER

*TABLE 146*

CREATE

*CONSTRAINT 145*

*DATABASE 143*

*TABLE 144, 145*

DROP

*DATABASE 143*

FROM 150

GROUP BY 152

HAVING 153

INSERT INTO 147

ORDER BY 154

REPLACE 148

SELECT 150

UPDATE 148

WHERE 148, 151, 154

## T

---

---

TCP/IP 41

Topologie

Logique 31

*Ethernet 31*

*Token ring 31*

Physique 28

*Anneau 29*

*Bus 28*

*Etoile 28*

*FDDI 31*

*Maillée 30*

Type de réseau

LAN 32

MAN 32

PAN 32

WAN 32

## U

---

---

Unité centrale 7

Carte mère 7

Mémoire centrale 8

Processeur 8

# INFORMATIQUE

## *Enseignement Secondaire Général*

Cet ouvrage aborde les thèmes du nouveau programme officiel d'informatique en terminale de l'enseignement secondaire général :

- Un module introductif, intitulé *Rappels* présente les notions de base d'exploitation de l'ordinateur et de conception de page web
- Le premier module, intitulé *Les réseaux informatiques* rappelle les notions de base des environnements informatiques et présente les concepts de réseaux informatiques pour déboucher sur la mise en place d'un réseau filaire ou sans fil.
- Le deuxième module, intitulé *Programmation interactive de pages Web* permet de compléter les notions de création de page Web statique étudiées en classe de première en y ajoutant de l'interactivité
- Le troisième module, intitulé *Programmation dynamique de pages Web* permet de compléter le module précédent en ajoutant du dynamisme aux pages Web.

Ce livre s'adresse d'une part aux enseignants. Il a pour support pédagogique l'approche constructiviste, qui permet d'associer les connaissances nouvelles aux notions apprises dans les classes précédentes. Chaque chapitre s'ouvre sur une situation de départ liée au contexte de l'élève. Le contenu des chapitres permet de résoudre les problèmes posés par la situation de départ, et les exercices aident à la consolidation des nouvelles connaissances et compétences.

Ce livre s'adresse également aux élèves. Progressif dans son approche des concepts, il a pour point d'orgue trois projets de consolidation qui permettront aux élèves d'asseoir les savoirs-faire nécessaires à la réalisation et à la publication d'un site Web dynamique intégrant les technologies JavaScript et PHP, le tout exploitable en local à travers un réseau informatique filaire ou sans fil.

## *Les auteurs*

### KAMAHA Alain Théophile

Licencié en Biochimie, il est actuellement élève-professeur en 5ème année Informatique à l'Ecole Normale Supérieure de Yaoundé

### MENGUE ANGO Christelle

Licencié en Droit, elle est actuellement élève-professeur en 5ème année Informatique à l'Ecole Normale Supérieure de Yaoundé

### MENGUE Annie Nadège

Licencié en Lettres Modernes Françaises, elle est actuellement élève-professeur en 5ème année Informatique à l'Ecole Normale Supérieure de Yaoundé

### MOYOU METCHEKA Léonel

Maître en Informatique, il est actuellement élève-professeur en 5ème année Informatique à l'Ecole Normale Supérieure de Yaoundé