

TITRE DE LA LEÇON 3 : INITIATION AU LOGICIEL SCRATCH

SITUATION D'APPRENTISSAGE

Vidéo en guise de situation d'apprentissage.

Présentation de Scratch : <https://youtu.be/FChrbv0nPSQ>

RESUME DE LA LEÇON

Avant d'entamer la programmation proprement dite, faisons un rappel sur les composantes d'un ordinateur.

URL: <https://youtu.be/VRGXel-cyZAi>

1. LA PROGRAMMATION INFORMATIQUE

Suis attentivement cette vidéo. URL : <https://youtu.be/ZuxcqHVxW5I>

- Un **algorithme** est une méthode permettant de résoudre un problème.
- Une **instruction** est

Exemple :

Voici ma recette pour faire une omelette avec trois œufs :

- Prendre trois œufs, une tomate, un oignon et du sel ;
- Casser l'œuf ;
- Verser son contenu dans une tasse ;
- Répéter les actions précédentes pour chacun des trois œufs ;
- Découper finement l'oignon ;
- Découper en cubes la tomate ;
- Les verser dans la tasse ;
- Battre l'ensemble avec une fourchette ;
- Ajouter une pincée de sel ;
- Faire chauffer de l'huile dans une poêle ;
- Renverser le contenu de la tasse dans la poêle ;
- Si le feu est trop fort alors il faut le réduire ;
- Laisser cuire pendant trois minutes
- Retirer dans une assiette puis le déguster.

C'est fini !

Cette recette est une méthode permettant de faire une omelette. C'est un *algorithme*.

Chacune des actions de cette recette est une *instruction*.

- Un algorithme se décompose en 3 phases :
 - L'**entrée des données** : « Entrée »
 - Le **traitement des données** : « Instructions »
 - La **sortie des résultats** : « Sortie »

Reprenons l'exemple de la recette de cuisine faire une omelette

Ingrédients (= **entrée des données**) : Trois œufs, une tomate, un oignon et du sel

Préparation (= **traitements des données / instructions**)

- Prendre trois œufs, une tomate, un oignon et du sel ;
- Casser l'œuf ;
- ...
- Laisser cuire pendant trois minutes

Le plat obtenu (= **Sortie des résultats**)

- Retirer dans une assiette puis le déguster.
- Pour donner des instructions à une machine, on utilise un **langage de programmation**, compréhensible à la fois par la machine et par l'être humain
Exemple : **Scratch** est un langage de programmation.
- Un **programme** est l'*expression d'un algorithme dans un langage de programmation*.
- La **programmation informatique** ou codage informatique est l'écriture des programmes pour développer des logiciels, des pages web, etc., mais aussi pour donner des ordres à des machines, leur dire ce qu'elles doivent faire, les guider ; par exemple des les machines des usines, les robots de cuisines, ...

2. LE LANGAGE DE PROGRAMMATION SCRATCH

Le logiciel Scratch repose sur une approche simplifiée et ludique de l'algorithmique, pour aider les élèves à développer leurs capacités de raisonnement.

Développé par le groupe de recherche de *Lifelong Kindergarten* en collaboration avec le laboratoire Média du *MIT*, ce langage de programmation simplifiée facilite les simulations numériques et leur partage. Il permet ainsi de créer une multitude de ressources pédagogiques comme des jeux, des histoires interactives, des animations,

3. INSTALLER SCRATCH

La vidéo suivante t'apprend comment installer Scratch sur un ordinateur.

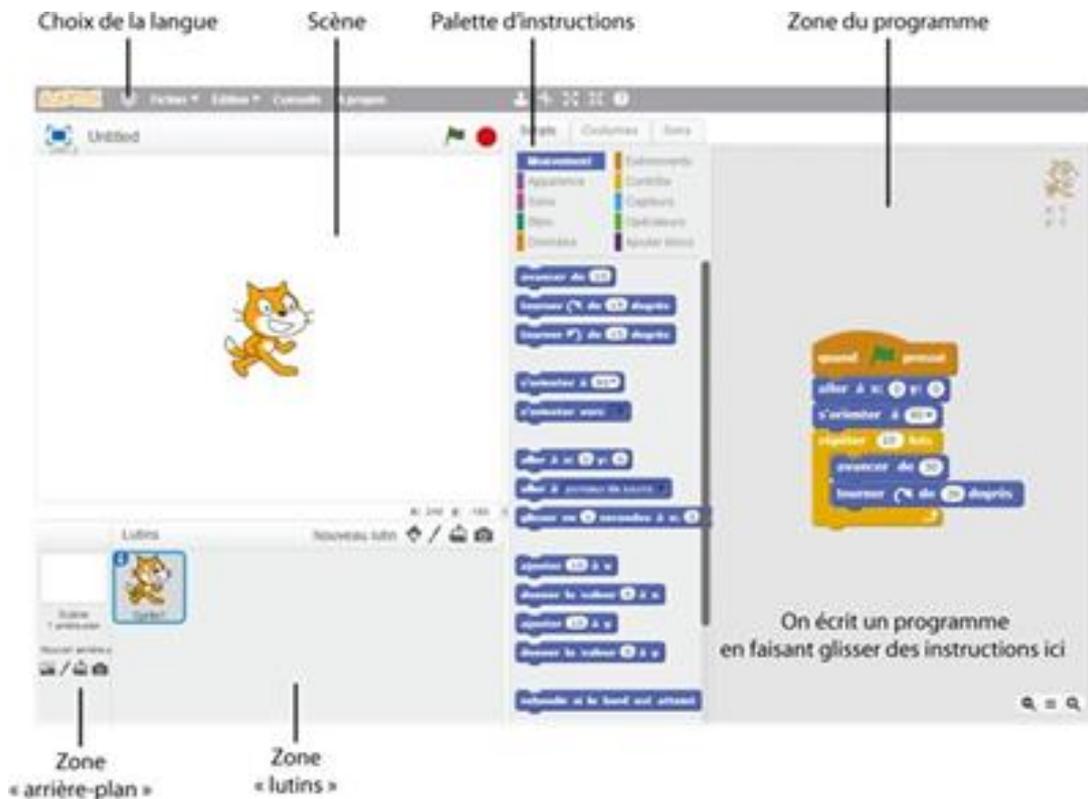
URL : <https://youtu.be/fsgWVQ3xpGw>

4. L'INTERFACE DE SCRATCH

Voici une présentation assez détaillée de l'interface de de Scratch :

URL: <https://youtu.be/xMS-UYtJ4dA>

Si c'est la première fois que Scratch est utilisé sur ta machine, il se peut que le logiciel soit en anglais. Dans ce cas, tu peux très facilement le passer en français en cliquant sur l'icône représentant un globe terrestre (en haut à gauche, à côté du logo « Scratch »).



Scratch est un langage de programmation conçu spécifiquement pour apprendre à programmer. Lorsque l'on ouvre le logiciel, il y a un *lutin* à l'écran (un chat). On peut lui donner des instructions simples.

- La **scène** : c'est là que se déroule le « jeu » (ou, plus généralement, le programme... on peut faire autre chose que des jeux dans Scratch !).
- L'**arrière-plan** : L'arrière-plan est fixe, contrairement aux lutins qui peuvent bouger. Par défaut, dans Scratch, l'arrière-plan est un fond blanc uni (plus tard, on le modifiera).
- L'onglet « **script** » : C'est ici que l'on va trouver les instructions (ou « **blocs** ») que l'on va pouvoir utiliser dans notre programme. Il y a de nombreuses instructions, qui sont regroupées par couleur (exemple : tout ce qui concerne le mouvement du lutin est dans un onglet bleu foncé, tout ce qui concerne son apparence est dans un onglet violet, etc.).

- La zone « **programme** », à droite de la palette d'instructions. C'est ici que l'on va écrire le programme, tout simplement

5. LES LUTINS :

Les lutins des ressources graphiques qui peuvent être programmées par des *blocs*. Ce sont les personnages ou les objets qui seront manipulés dans le programme. Ils peuvent se déplacer, changer de forme, parler, interagir avec les autres lutins. Lorsqu'on lance Scratch, il n'y a qu'un seul lutin d'affiché à l'écran : un chat. Plus tard, on ajoutera d'autres lutins et on supprimera le chat.

Pour en savoir plus sur les lutins, regarde cette vidéo. Tu y sauras ce qu'est un lutin, comment ajouter un lutin à une scène. Tu pourras faire même ton premier programme Scratch.

URL : <https://youtu.be/hfACv9kXYqA>

6. LES BLOCS

Pour programmer un lutin, fais glisser les blocs de programmation de la palette des blocs dans l'aire des scripts. Il suffit de cliquer sur un bloc pour exécuter l'instruction de programmation correspondante.

Pour créer des programmes dans Scratch, tu assembles tout simplement des blocs identifiables par leur couleur et leur instruction afin de créer des piles. Lorsque tu places sur la tête de la pile un bloc de contrôle de type chapeau, la pile devient un script. Les blocs sont conçus pour s'assembler de telle façon que la logique de la syntaxe soit respectée excluant toute source d'erreur. Les différents types de données ont des formes différentes, éliminant ainsi toute incompatibilité de type.

Clique n'importe où sur la pile pour exécuter l'ensemble du programme, de haut en bas.

La vidéo suivante t'en dit plus : https://youtu.be/Xupx4_h8F_o

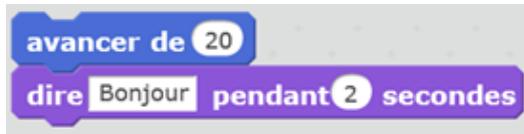
Par exemple, pour demander au chat de se déplacer de 10 pas, il suffit de faire glisser l'instruction « avancer de 10 » depuis la palette d'instructions vers la zone du programme. Si l'on clique ensuite sur cette inscription, on remarque que le chat avance bien de 10 pas (1 pas = 1 pixel de l'écran).



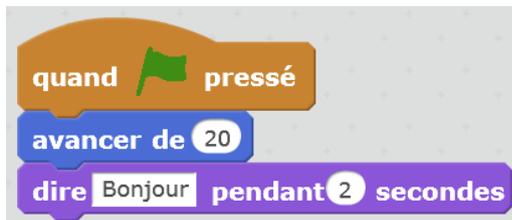
Si l'on souhaite avancer de 20 pas, il suffit de changer « 10 » en « 20 » en cliquant dans la zone dédiée.

Si maintenant on souhaite que le chat avance de 20 pas, puis dise « Bonjour », il suffit de coller la nouvelle instruction à la fin du programme. L'instruction « dire bonjour » n'existe pas, mais il y a une instruction « dire Hello » dans la catégorie

« apparence » de la palette d'instructions. Il suffit de prendre cette instruction puis de remplacer le texte « Hello » en « Bonjour » en cliquant sur ce texte. Écrire un programme se fait simplement en emboîtant des instructions entre elles.



Si maintenant on veut que le chat fasse cela à chaque fois que l'on clique sur le drapeau vert (en haut à droite de la scène, le drapeau vert permet de lancer le programme), alors il faut rajouter l'instruction « Quand drapeau vert pressé » à chercher dans la catégorie « événements » des instructions. Cela donne :



Pour supprimer une instruction (ou tout un bloc d'instructions) : il suffit de faire glisser cette instruction (ou ce bloc) depuis la zone du programme vers la palette des instructions.

Chaque catégorie de blocs est représentée par une couleur :

- **Mouvement** (bleu foncé) : ces blocs permettent de définir des mouvements que l'on veut faire effectuer par le lutin

Voici une vidéo qui t'apprend comment Déplacer le lutin

URL: <https://youtu.be/onJG9sLbtNE>

- **Apparence** (violet) : ces blocs permettent de modifier l'apparence du lutin et de, principalement, lui faire dire des choses ou interagir avec d'autres lutins et/ou l'arrière-plan.
- **Sons** (mauve) : permet d'ajouter et de contrôler des éléments sonores. - Stylo (vert) : permet l'utilisation d'un stylo, pour représenter les déplacements du lutin.
- **Données** : permet de créer de nouvelles variables. - **Événement** : permet de définir un événement qui déclenche le programme.
- **Contrôle** (jaune) : contrôle l'exécution du script (les différentes étapes choisies et enregistrées au préalable), permet d'effectuer les boucles, les instructions, etc.
- **Capteurs** (vert bleu) : dans cette catégorie, il y a des blocs pour faire parler le lutin et les blocs de tests. Ces blocs permettent d'effectuer une action

dans le cas où le lutin touche un autre élément de la scène, ou si la souris touche un élément, etc. Cela permet de demander l'exécution d'une instruction dans le cas où l'ordinateur détecte une action en particulier (désignée par l'utilisateur au préalable).

- **Opérateurs** (vert clair) : les blocs opérateurs permettent d'effectuer des opérations mathématiques.
- **Ajouter blocs** : permet de créer un nouveau bloc.

7. LES COSTUMES

Un même lutin peut avoir plusieurs aspects (**costumes**) et on peut modifier sa taille.

Exemple : le chat (lutin par défaut) a deux costumes, c'est à dire deux positions et en les alternant on peut donner l'impression qu'il marche, pour cela on utilise l'instruction « costume suivant » dans la rubrique « Apparence ».

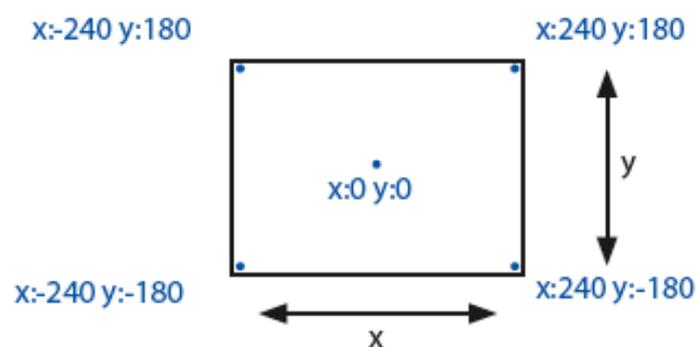
Regarde la vidéo suivante pour en savoir plus :

URL: <https://youtu.be/Joi6kpvfjBk>

8. L'ARRIERE-PLAN

L'**arrière-plan** (le décor, la scène) est l'endroit où des histoires, des jeux et des animations prennent vie. Les lutins se déplacent et interagissent les uns avec les autres sur la scène.

La scène est un rectangle de 480 unités de large et 360 unités de hauteur. L'unité correspond au pixel. Elle est munie d'une grille de repérage avec un système de coordonnées (x, y).



Le centre de la scène a une abscisse $x = 0$ et une ordonnée $y = 0$. Regarde la vidéo suivante pour savoir comment changer d'arrière-plan, etc.

URL : <https://youtu.be/DaC8QoQeCol>

9. LE STYLO

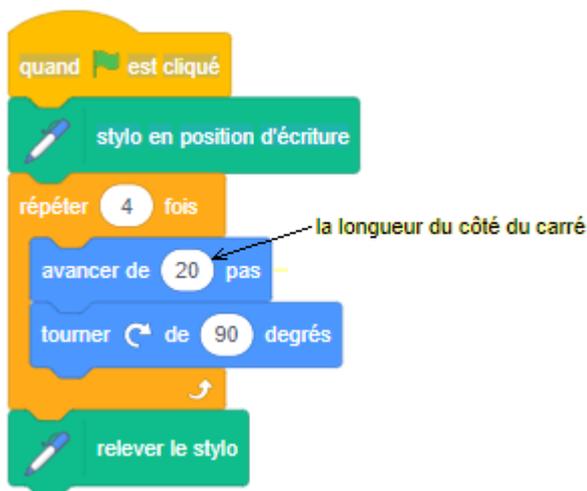
Dans cette vidéo, tu vas apprendre comment utiliser les blocs de type « Stylo » qui permet de dessiner sur la scène en programmant un lutin.

URL: <https://youtu.be/wgm51bz4rls>

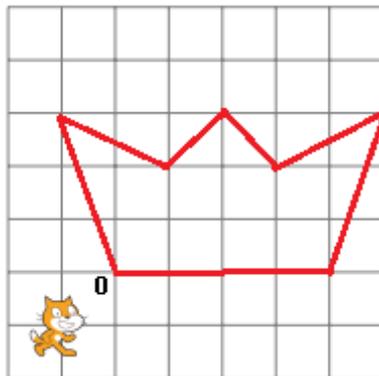
Le bloc « Stylo en position d'écriture » permet de mettre le stylo en mode « écriture » et donc de dessiner sur la scène lorsque le lutin qui contient le bloc bouge.

Différents effets, comme la couleur, saturation, luminosité et transparence de la trace du stylo peuvent être modifiés avec d'autres blocs de l'extension.

Exemple : Dessiner un carré (avec un côté de 20)



SITUATION D'ÉVALUATION :



Réalise un programme pour que le lutin dessine la couronne royale ci-contre. (O désigne le centre de l'arrière-plan).

EXERCICES / DOCUMENTATION :

Exercice 1



- 1) Lance Scratch
- 2) Ecris le programme ci-contre
- 3) Exécute le programme et dis quelle est la figure géométrique que Scratch a tracée
- 4) Donne les dimensions de cette figure

Exercice 2

Ecris un programme qui fait tracer au lutin un rectangle de longueur 250 et de largeur 120.

Exercice 3

Les carreaux font 40 unités de large. A l'aide du script ci-dessous à gauche, dessine sur une feuille quadrillée le chemin du lutin-chat. La position initiale du lutin-chat est à l'intersection des segments qu'il cache.

