Algorithmes et programmation

1. Algorithme

Définition

Un **algorithme** est un ensemble d'actions ou d'instructions claires, à réaliser dans un ordre bien précis, permettant de répondre à une situation donnée.

Quelques exemples:

- recettes de cuisine
- passage à la cantine
- notice de montage d'un meuble
- programmes de calcul
- programme de construction géométrique

Dans certains cas, on peut découper l'algorithme en plusieurs sous-algorithmes pour le rendre plus facile à comprendre.

Un algorithme se compose de 3 grandes parties: pour notre recette de gâteau

- les **informations** dont on a besoin au départ *(liste des ingrédients)*
- la **succession d'instructions** à appliquer *(étapes de la préparation)*
- la **réponse** que l'on obtient à l'arrivée *(le gâteau final)*

2. Programmation

Définition

Un **programme** est un algorithme écrit dans un langage que peut comprendre une machine ou un logiciel informatique.

Pour programmer au collège, nous allons utiliser le logiciel SCRATCH .

3. Le logiciel Scratch

Il est téléchargeable et utilisable gratuitement : https://scratch.mit.edu/scratch2download/
On peut aussi l'utiliser en ligne : http://scratch.mit.edu/

C'est un logiciel de programmation visuelle qui permet, de faire exécuter des **commandes** à un ou plusieurs **lutins**, de créer des animations, des programmes, des algorithmes.

Amusez-vous !

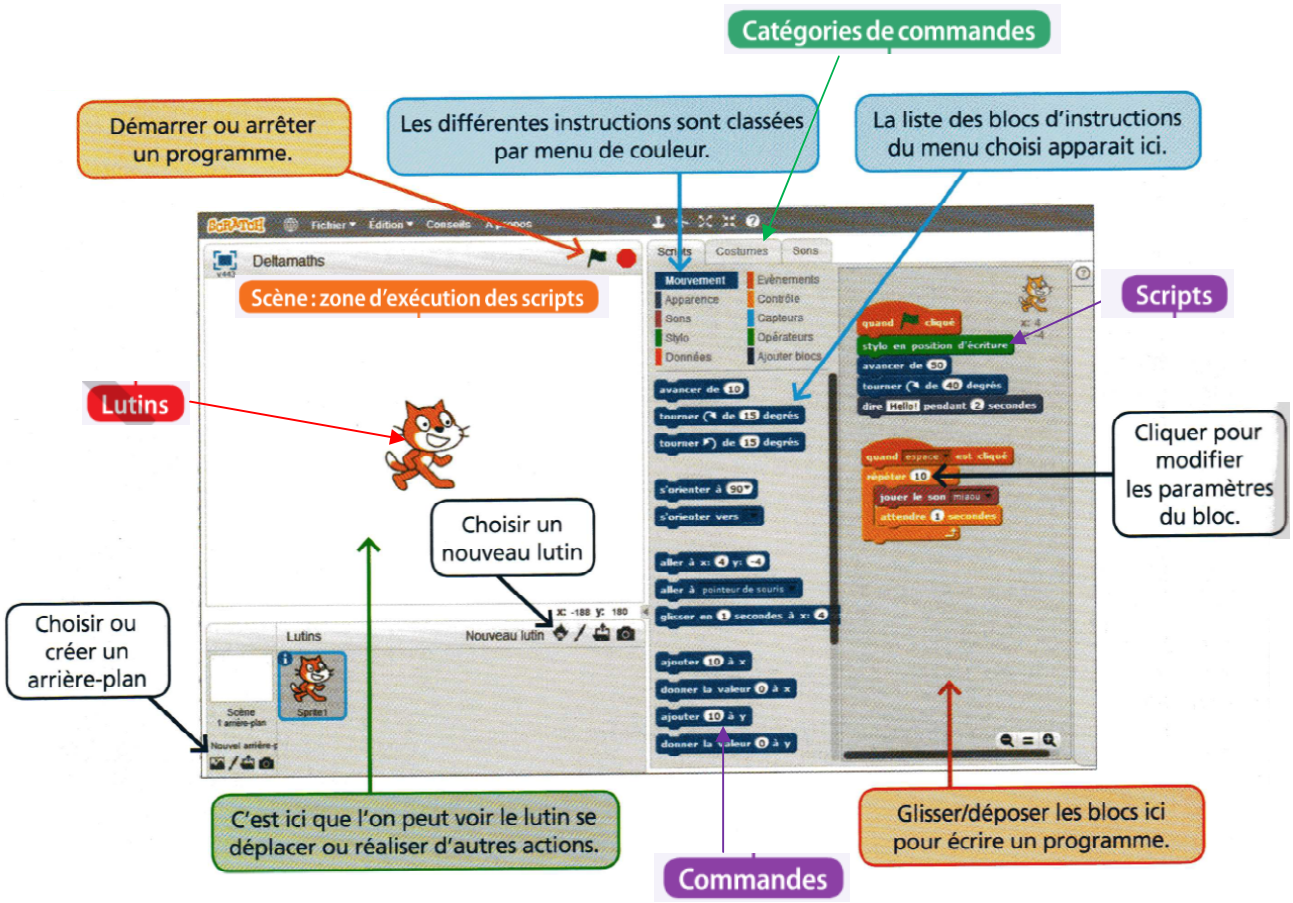
Voici un lien vers une notice d'utilisation pour bien commencer avec Scratch:

http://etab.ac-poitiers.fr/coll-Truffaut-chef-boutonne/IMG/pdf/debuter_avec_scratch.pdf

et un site pour commencer à jouer : http://code.org

La langue d'utilisation est par défaut l'anglais. Pour la changer, il faut cliquer en haut à gauche de l'écran dans la barre de menu.

L'interface de Scratch est partagée en plusieurs zones:



- Un **script** est un programme (c'est à dire une suite de blocs) qui définit les actions d'un **lutin**.
- Un lutin peut avoir plusieurs scripts; chaque script peut être démarré de différentes manières.
- Pour **écrire un programme**, on encastre des blocs dans la fenêtre de script, à droite.
- Pour **lancer un programme**, on double-clique sur l'assemblage d'instructions ou, si l'on a fait commencer la série par le bloc événement avec le drapeau vert, il suffit de cliquer sur le drapeau vert situé au-dessus de la fenêtre d'action.

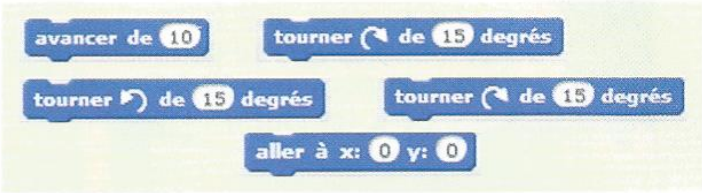
  Pour **arrêter un programme**, on clique sur l'octogone rouge situé à droite du drapeau vert.

SCRATCH

Les principales instructions

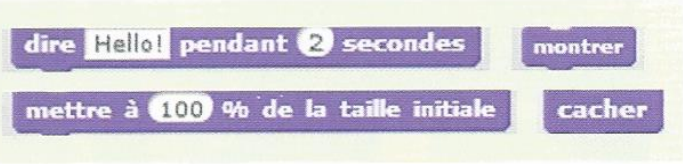
Mouvement

Ce menu sert principalement à déplacer les lutins dans la scène et à leur donner des directions de déplacement.



Apparence

Ce menu permet de changer l'apparence des lutins et surtout de leur faire dire des choses ce qui permet de communiquer les résultats d'un calcul par exemple.



Sons

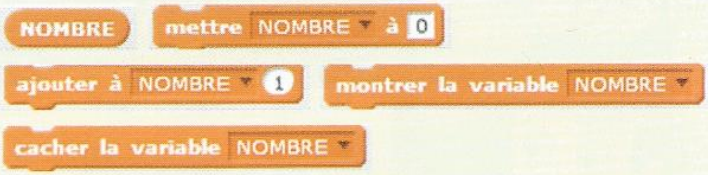
Ce menu permet de faire jouer des sons courts ou une musique de fond et de gérer ces sons.

Stylos

Ce menu sert à laisser une trace, ou non, lors des déplacements des lutins ce qui permet par exemple de tracer des figures.

Données

Ce menu permet de définir des variables et ensuite d'opérer sur celles-ci en ajoutant ou affectant des valeurs.

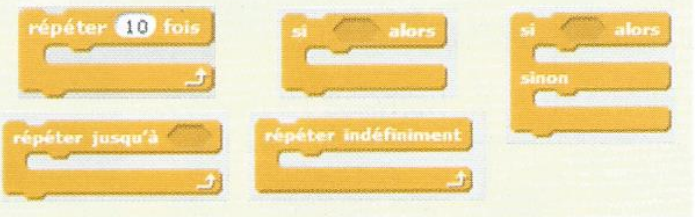


Événements

Ce menu permet de définir l'événement qui déclenchera le programme : un appui sur le drapeau vert ou sur une touche du clavier par exemple.

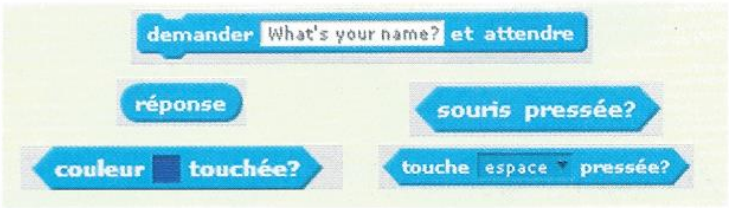
Contrôle

C'est dans ce menu que se trouvent les différentes boucles et les instructions conditionnelles.



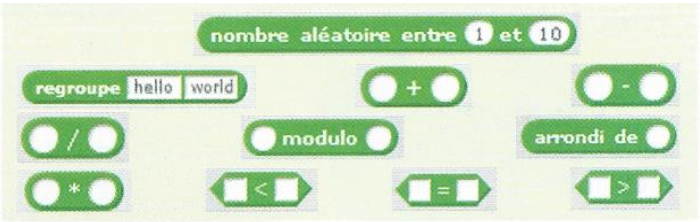
Capteurs

Par ce menu, on aura la possibilité de faire poser une question aux lutins et de mémoriser temporairement la réponse.
On trouvera également différents tests à insérer dans les boucles ou instructions conditionnelles.



Opérateurs

Ce menu permet d'effectuer des calculs, de regrouper des chaines de caractères et de créer des tests à insérer dans les boucles ou instructions conditionnelles.



La scène

La largeur de la scène est égale à 480 points et sa hauteur à 360 points.

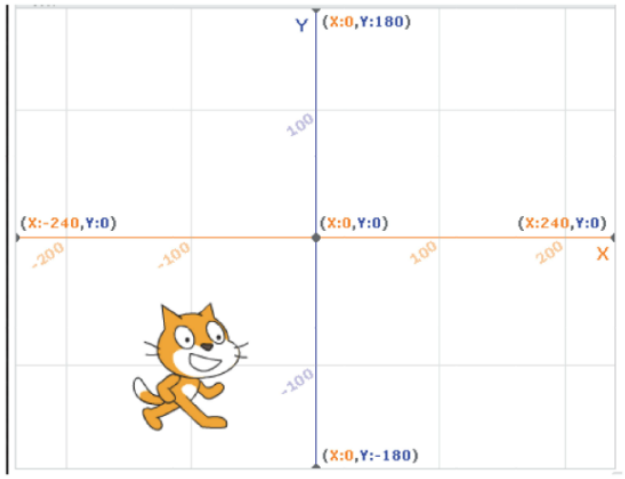
Dans SCRATCH, un objet est appelé « lutin ».

On le positionne à l'aide de deux coordonnées qui sont désignées par les lettres **x** et **y**.

La première coordonnée, **x**, varie entre – 240 et 240 et la seconde coordonnée, **y**, varie entre – 180 et 180.

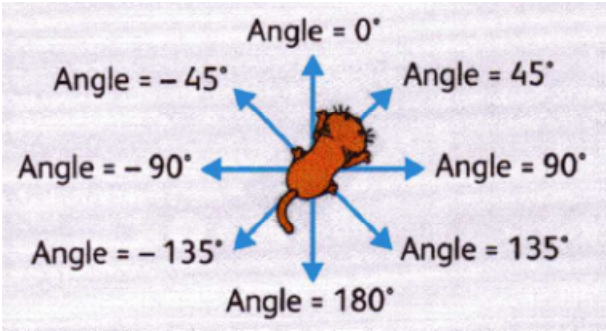
Par exemple, le chat ci-contre est à la position (– 100 ; – 100).

Les coordonnées du chat s'affichent en haut à droite de la fenêtre *Scripts*.

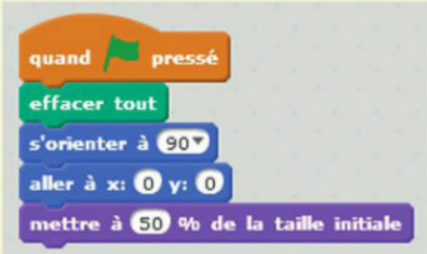


L'arrière-plan *xy-grid* permet de voir la grille et les coordonnées.

L'orientation du lutin



Un programme pratique pour réinitialiser l'affichage



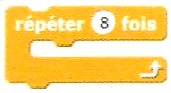
Cet algorithme peut démarrer aussi avec :

On remet **le lutin** en situation initiale dès que l'on appuie sur le drapeau.

SCRATCH

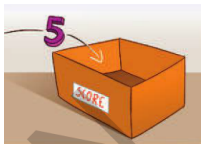
Les boucles

Une **boucle** permet d'effectuer une même partie d'un programme **plusieurs fois**.

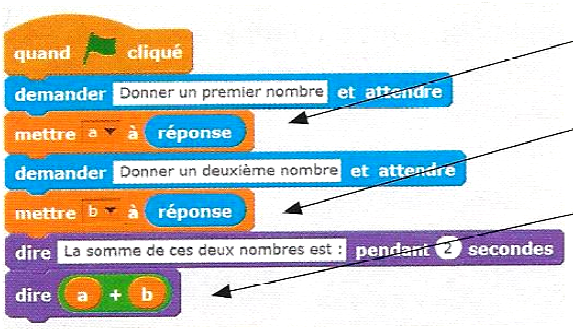
Boucle définie par un nombre donné de répétitions.	Boucle qui s'arrête quand une condition est réalisée.	Boucle qui ne s'arrête pas (sauf si le bouton est actionné).
		

Les variables

Une **variable** est une **boite** dans laquelle on **stocke** une **information** ou une **valeur** pour l'utiliser plus tard et qui peut changer au fil du temps.



On désigne une variable par un nom ou une lettre.



- Le programme range la réponse dans la variable *a*.
- Le programme range la réponse dans la variable *b*.
- Le programme utilise les deux variables pour donner un résultat.

Les instructions conditionnelles

En programmation, on utilise l'instruction **conditionnelle** « Si……. Alors » lorsqu'une action doit être effectuée seulement si une **condition est vérifiée.**

Il existe types différents d'instructions conditionnelles :

et

Les boucles

Une **boucle** permet d'effectuer une même partie d'un programme **plusieurs fois**.

Boucle définie par un nombre donné de répétitions.	Boucle qui s'arrête quand une condition est réalisée.	Boucle qui ne s'arrête pas (sauf si le bouton est actionné).
		

Les instructions conditionnelles

En programmation, on utilise l'instruction **conditionnelle** « Si……. Alors » lorsqu'une action doit être effectuée seulement si une **condition est vérifiée.**

Il existe types différents d'instructions conditionnelles :

 et 

Les boucles

Une **boucle** permet d'effectuer une même partie d'un programme **plusieurs fois**.

Boucle définie par un nombre donné de répétitions.	Boucle qui s'arrête quand une condition est réalisée.	Boucle qui ne s'arrête pas (sauf si le bouton est actionné).
		

Les instructions conditionnelles

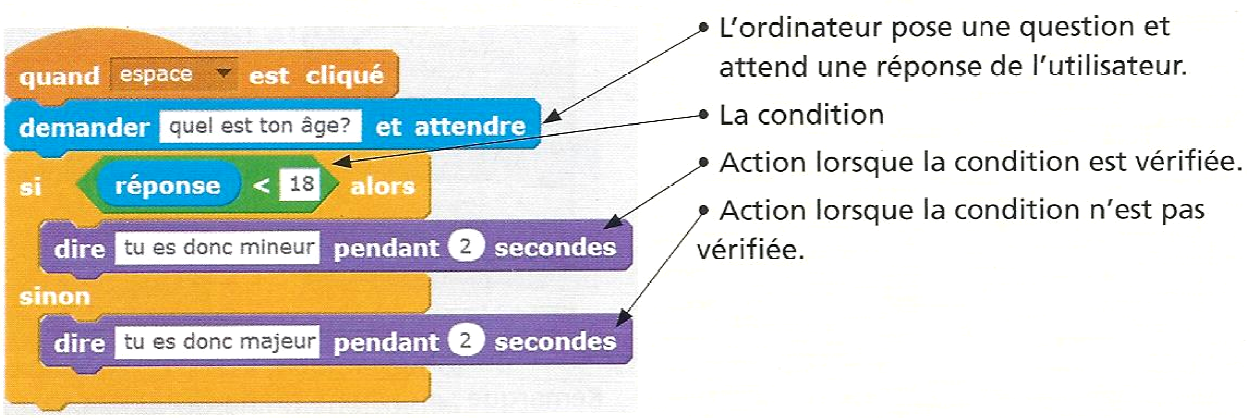
En programmation, on utilise l'instruction **conditionnelle** « Si……. Alors » lorsqu'une action doit être effectuée seulement si une **condition est vérifiée.**

Il existe types différents d'instructions conditionnelles :

 et 

Les blocs

Un **bloc** est un **groupement d'instructions paramétré.** Il est constitué d'une suite d'instructions dont l'une au moins utilise une information précisée dans le bloc lui-même.



Ici, le bloc **Carré** a été créé avec un paramètre : *côté*.
Avec Carré 100, on trace un *carré* de *côté* 100.