

I) Vocabulaire

Définition

Un algorithme est une liste ordonnée d'instructions permettant d'effectuer une tâche, de résoudre un problème.

Exemple. Voici plusieurs exemples d'algorithmes :

- une recette de cuisine est un algorithme qui permet de cuisiner un plat
- les itinéraires déterminés par un GPS sont des algorithmes permettant de passer d'un lieu à un autre
- les programmes de construction en géométrie sont des algorithmes permettant de tracer une figure
- ...

Remarque. Un algorithme peut être écrit en langage naturel (français, anglais ou tout autre langue) ou dans un langage de programmation afin de permettre à un ordinateur ou encore à une calculatrice d'interpréter et d'exécuter les instructions.

Propriété

Un algorithme se découpe en trois parties :

- les informations nécessaires à l'utilisation de l'algorithme
- la suite d'instructions à effectuer
- la réponse attendue à la fin

Exemple. Pour une recette de cuisine :

- les informations nécessaires à l'utilisation de l'algorithme sont les ingrédients.
- la suite d'instructions à effectuer sont les étapes de la recette
- la réponse attendue à la fin est le plat cuisiné

II) Les variables

II.1) Qu'est-ce qu'une variable ?

Définition

Une **variable** est une petite information stockée en mémoire **temporairement**. Comme son nom l'indique, cette information peut être amenée à **varier** au cours de l'exécution du programme.

Une variable est toujours constituée de deux éléments :

- **son nom** : c'est ce qui permet de la reconnaître.
- **sa valeur** : c'est l'information qu'elle contient.

Remarque. Une variable peut contenir différents types d'informations. On peut par exemple y stocker des nombres ou encore du texte.

Exemple. Lors de la programmation d'un jeu vidéo, on pourrait vouloir stocker le nombre de vies restantes du joueur. Ce nombre est amené à changer au cours du jeu, si le joueur perd une vie par exemple.

Méthode

Pour créer une variable sur Scratch, il faut se rendre dans le sous-menu "Variables" puis cliquer sur le bouton "Créer une variable".

II.2) Affecter une valeur à une variable

Définition

Affecter une valeur à une variable, c'est donner une valeur à cette variable.

Exemple. Pour affecter la valeur 5 à la variable "nombreDeVies", il faut se rendre dans le sous-menu "Variables" et ajouter le bloc suivant à son programme :



Remarque. Lors de l'utilisation d'un bloc **demandez et attendez** (disponible dans le sous-menu "Capteurs"), une variable **réponse** (aussi disponible dans le sous-menu "Capteurs") est automatiquement créée. Elle permet de récupérer la réponse entrée par l'utilisateur à la question posée.

II.3) Afficher le contenu d'une variable

Méthode

Lorsqu'une variable est créée, on peut accéder à sa valeur grâce à l'ovale **variable** disponible dans le sous-menu "Variables".

Exemple. Reprenons l'exemple d'un jeu vidéo que l'on voudrait programmer. Si l'on désire mettre en place un système de vies pour le joueur, le programme pourrait donc commencer comme ceci :

Programme :



Résultat :



Remarque. Nous aurions pu utiliser d'autres blocs pour afficher le résultat, comme par exemple le bloc **pense à**.


II.4) Faire des calculs simples

Propriété

Pour agir sur des variables, on utilise des opérateurs :

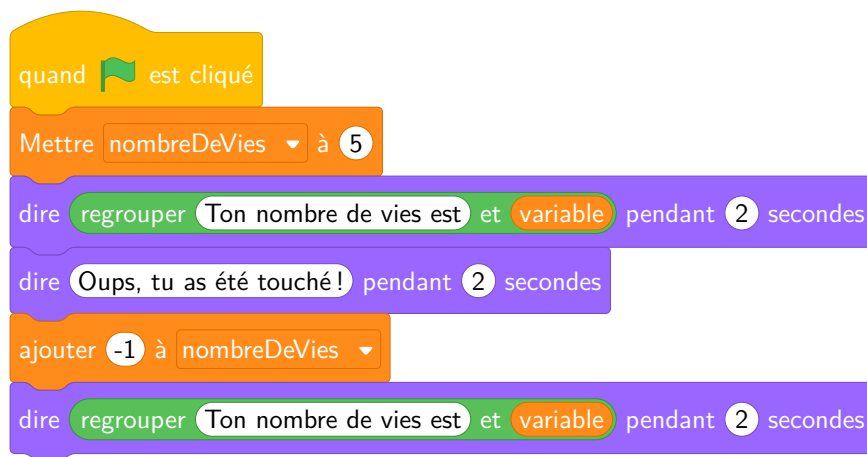
- Si la valeur stockée est un nombre, on peut alors effectuer des additions, des soustractions, des multiplications ou encore des divisions.
- Si la valeur stockée est un texte, alors il est possible de le regrouper avec un autre texte, de récupérer une lettre dans ce texte ou de connaître la longueur de ce texte.

Les opérateurs sont disponibles dans le sous-menu "Opérateurs" de Scratch.

Remarque. Pour additionner ou soustraire un nombre à une variable sur Scratch, il est aussi possible d'utiliser le bloc . Pour effectuer une soustraction, il suffira que le nombre soit négatif.

Exemple. Nous pouvons maintenant améliorer un peu notre jeu vidéo :

- nous pouvons afficher "Tu as 5 vies" plutôt que de n'afficher que 5 grâce au regroupement de texte.
- Nous pouvons faire perdre ou gagner une vie au joueur.



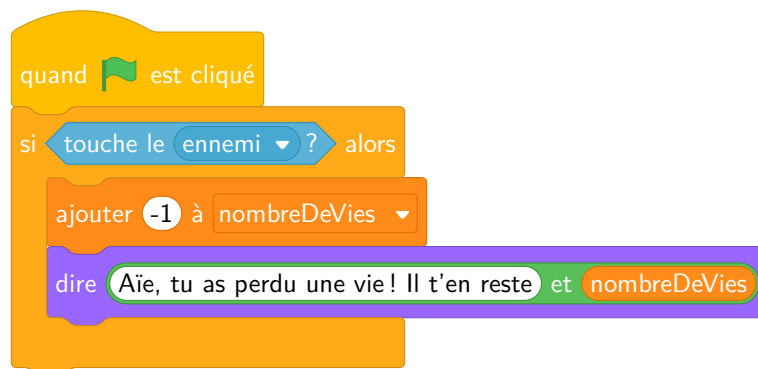
III) Les conditions

Propriété

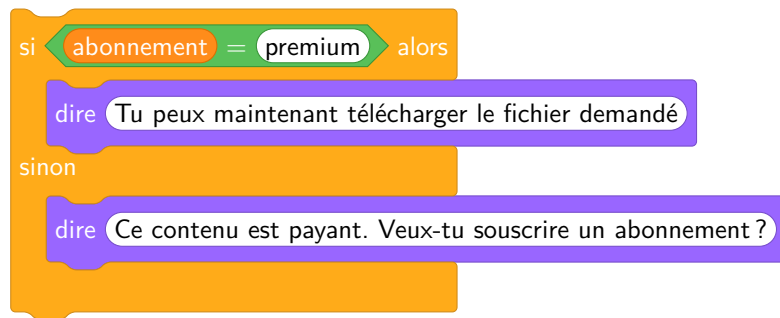
Il est parfois nécessaire de n'exécuter des instructions que dans certaines conditions. On va pour cela effectuer un test et il y a deux structures possibles :

- Si **une condition est vérifiée** alors **faire une action particulière**
- Si **une condition est vérifiée** alors **faire une action particulière** sinon **une autre action est réalisée**

Exemple. • Dans un jeu vidéo, le joueur peut, par exemple, perdre une vie lorsqu'il est touché par un ennemi. La condition serait alors : Si **le joueur est touché par l'ennemi** alors **le joueur perd une vie**



- Sur un site internet, il peut parfois arrivé que le contenu change en fonction de la catégorie de l'utilisateur. La condition serait alors : Si **l'utilisateur possède une abonnement premium** alors **on affiche le contenu désiré** sinon **on propose à l'utilisateur de souscrire un abonnement**



IV) Les boucles

Il peut parfois être utile de réaliser plusieurs fois la même action. Pour cela, on peut utiliser les boucles qui évitent d'écrire plusieurs fois les mêmes instructions. Il en existe deux types.

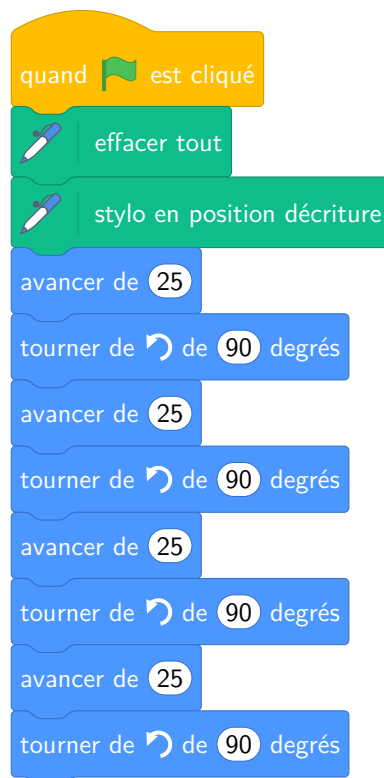
IV.1) Les boucles itératives

Définition

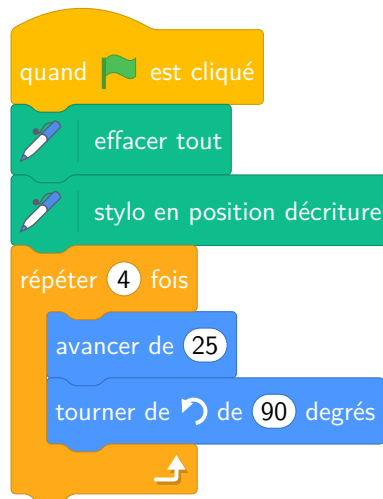
On appelle **boucle itérative** la répétition d'une instruction (ou d'un groupe d'instructions) un nombre de fois donné.

Une **itération** est une répétition de cette instruction (ou groupe d'instructions).

Exemple. Si l'on désire tracer un carré sur Scratch, on pourrait exécuter le programme suivant :



On remarque cependant qu'une partie du code est répétée 4 fois. On peut donc utiliser une boucle itérative pour simplifier celui-ci, ce qui donne :



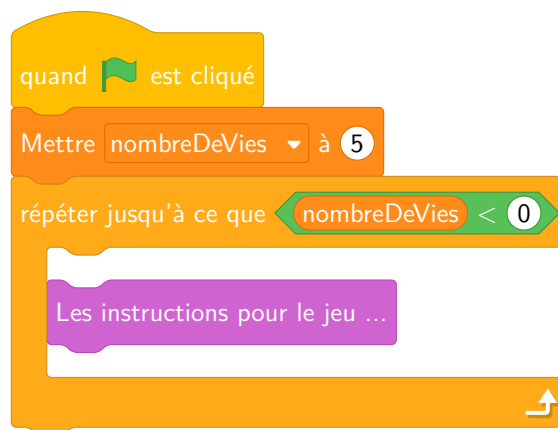
IV.2) Les boucles conditionnelles

Définition

On appelle **boucle conditionnelle** la répétition d'une instruction (ou d'un groupe d'instructions) tant qu'une condition est vérifiée.

Remarque. Dans ce cas, on ne sait pas à l'avance combien de fois l'instruction (ou le groupe d'instructions) seront répétés.

Exemple. Dans un jeu vidéo, on veut que le joueur puisse jouer tant qu'il possède encore des vies. On aurait alors le programme suivant :



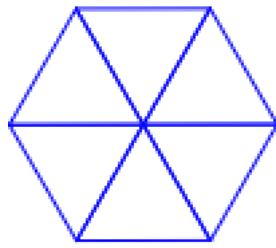
V) Les fonctions

Nous avons vu qu'il était possible de répéter des instructions plusieurs fois grâce aux boucles. Il peut cependant arriver que les mêmes instructions soient utilisées à différents endroits dans le même programme.

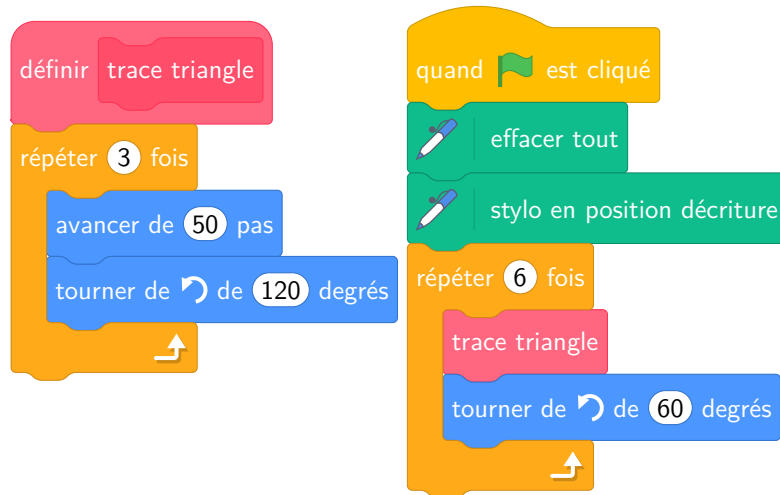
Définition

Les fonctions (ou blocs dans Scratch) sont des "morceaux de programmes" que l'on peut utiliser dans notre programme.
Il est possible de fournir un ou plusieurs paramètre(s) à la fonction.

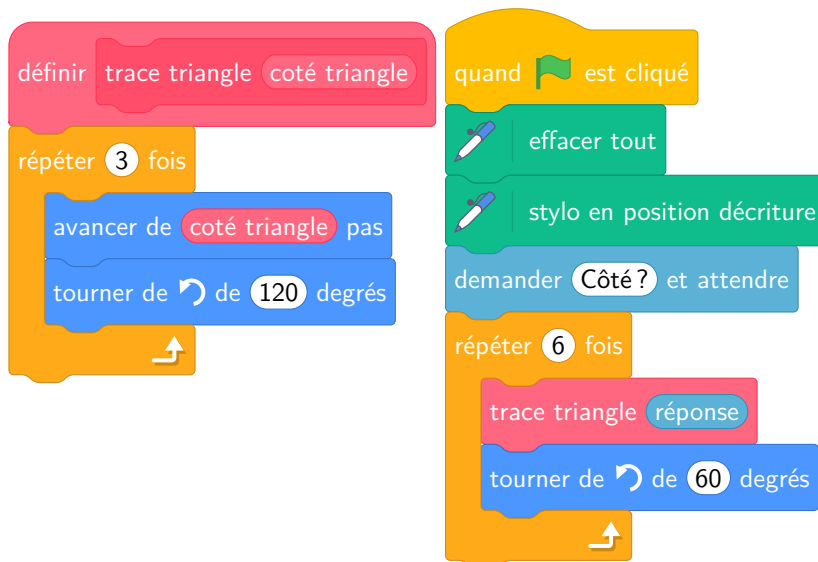
Exemple. Voici une figure :



On remarque que c'est un hexagone composé de six triangles équilatéraux. Pour la reproduire sur Scratch, on pourrait alors définir une fonction qui permet de tracer un triangle équilatéral et demander à reproduire ce triangle six fois.



Il est même possible d'aller plus loin en demandant à l'utilisateur la taille des côtés pour son hexagone. On aura pour cela besoin d'un paramètre dans la fonction "trace triangle" :



On voit ici que la réponse donnée par l'utilisateur est passée en paramètre à la fonction "trace triangle" qui la récupère pour tracer les côtés à la bonne longueur.