SCRATCH ET ALGORITHMES

COURS

Scratch et algorithmes

DURÉE ESTIMÉE

20 minutes

OBJECTIFS DU CHAPITRE

- Utiliser une variable dans Scratch
- Utiliser une instruction conditionnelle dans Scratch
- Utiliser une boucle dans Scratch
- Traduire un programme de calcul en Scratch
- Dessiner une figure géométrique avec Scratch

Un algorithme est une méthode systématique pour résoudre un problème. Le logiciel Scratch permet de traduire un algorithme en un programme visuel, à l'aide de blocs colorés que l'on assemble.

1 - Notion d'algorithme

Algorithme

Un **algorithme** est une suite finie d'instructions, exécutées dans un ordre précis, qui permet de résoudre un problème ou d'effectuer un calcul.

Exemple

Le programme de calcul suivant est un algorithme :

- Choisir un nombre
- Multiplier ce nombre par 3
- Ajouter 5 au résultat
- Annoncer le résultat

Pour $x = 4$: on obtient $4 \times 3 + 5 = 17$.

Pour $x = 10$: on obtient $10 \times 3 + 5 = 35$.

Remarque

Dans Scratch, chaque type d'instruction est représenté par un bloc de couleur différente. Les blocs s'emboitent les uns dans les autres pour former un programme.

2 - Variables

Variable

Une **variable** est un espace mémoire qui porte un nom et qui stocke une valeur (nombre ou texte). L'**affectation** consiste à donner une valeur à une variable.

Exemple

Le programme suivant crée une variable appelée « score », lui donne la valeur 0, puis lui ajoute 5 :

Télécharger le projet Scratch .sb3

A la fin de ce programme, la variable « score » contient la valeur 5.

Attention

Lorsqu'on affecte une nouvelle valeur à une variable, l'ancienne valeur est effacée. Par exemple, si on écrit « mettre score à 10 » alors que score valait 5, la variable contient désormais 10 (et non 15).

3 - Instructions conditionnelles

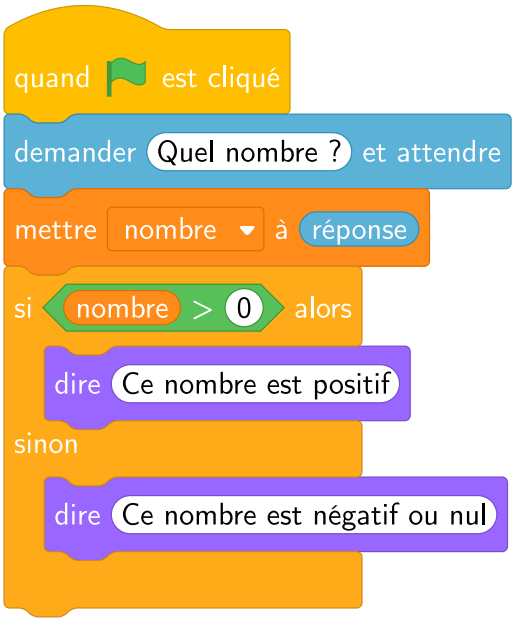
Instruction conditionnelle

Une **instruction conditionnelle** (ou test) permet d'exécuter des instructions uniquement si une condition est vérifiée. On distingue deux formes :

- **si ... alors** : les instructions ne sont exécutées que si la condition est vraie
- **si ... alors ... sinon** : un premier bloc d'instructions est exécuté si la condition est vraie, un second bloc si elle est fausse

Exemple

Ce programme demande un nombre et indique s'il est positif ou négatif :



Projet : positif ou négatif .sb3

Remarque

La condition est un test qui ne peut prendre que deux résultats : vrai ou faux. Les opérateurs de comparaison les plus courants sont : $=$, $<$ et $>$.

4 - Boucles

Boucle

Une **boucle** permet de répéter un groupe d'instructions. On distingue deux types :

- **Répéter ... fois** : les instructions sont répétées un nombre de fois connu à l'avance
- **Répéter jusqu'à ...** : les instructions sont répétées tant qu'une condition n'est pas vérifiée

Exemple

Ce programme trace un carré de 100 pas de côté :

Télécharger le projet Scratch .sb3 ?

On utilise « répéter 4 fois » car un carré a 4 côtés. A chaque passage dans la boucle, le lutin avance et tourne d'un angle droit.

Exemple

Ce programme calcule la somme $1 + 2 + 3 + \ldots + 100$ en utilisant une boucle « répéter jusqu'à » :

Projet : somme de 1 à 100 .sb3

La boucle s'arrête lorsque le compteur dépasse 100. La variable « somme » contient alors 5050.

Attention

Il faut toujours initialiser les variables avant une boucle. Si la variable « somme » n'est pas mise à 0 au départ, elle conserve sa valeur précédente et le résultat est faux.

5 - Programmes de calcul

Programme de calcul

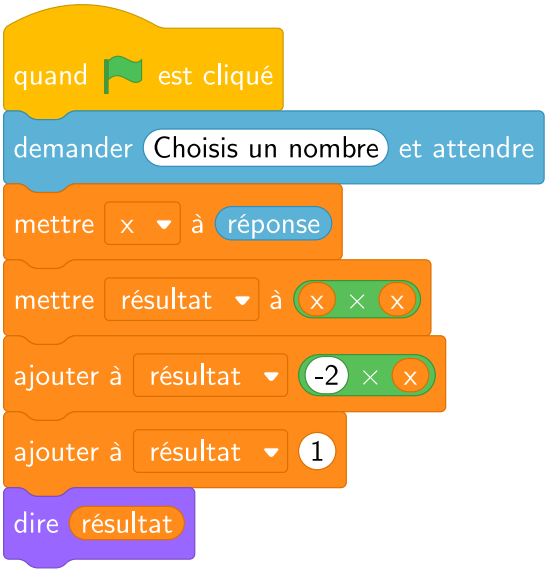
Un **programme de calcul** est une suite d'opérations mathématiques appliquées à un nombre de départ. Il peut être traduit en programme Scratch ou en expression algébrique.

Exemple

Soit le programme de calcul suivant :

- Choisir un nombre x
- Calculer son carré
- Soustraire le double du nombre de départ
- Ajouter 1

Ce programme se traduit en Scratch :



Projet : programme x² – 2x + 1 .sb3

L'expression algébrique correspondante est :

$$x^2 - 2x + 1 = (x - 1)^2$$

Verification pour $x = 3$: le programme donne $9 - 6 + 1 = 4$ et $(3 - 1)^2 = 4$.

6 - Blocs personnalisés

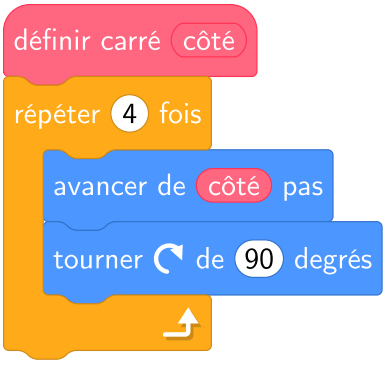
Lorsqu'une même suite d'instructions revient plusieurs fois dans un programme, on peut la regrouper sous un nom unique : c'est un **bloc personnalisé**. Il se crée dans la catégorie « Mes blocs » et fonctionne comme une **fonction** : on le définit une seule fois, puis on l'appelle autant de fois qu'on veut. Un bloc peut recevoir un **paramètre** (une valeur d'entrée) pour s'adapter à chaque appel.

Bloc personnalisé

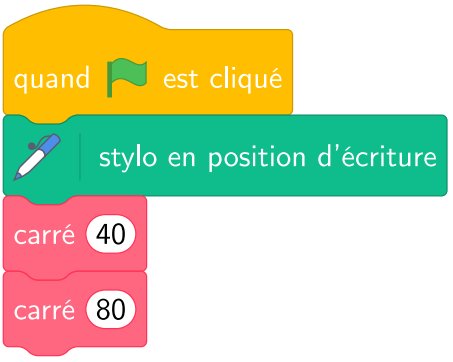
Un **bloc personnalisé** est un bloc créé par l'utilisateur qui regroupe une suite d'instructions sous un nom. On le **définit** avec le chapeau « définir », puis on l'**appelle** dans un programme. Un **paramètre** est une valeur d'entrée que l'on choisit au moment de l'appel.

Exemple

Le bloc « carré (côté) » trace un carré dont la taille est donnée par le paramètre « côté ». On le définit une seule fois :



On peut ensuite l'appeler avec n'importe quelle taille, sans réécrire la boucle :



Projet : blocs personnalisés .sb3

Le bloc « carré 40 » trace un petit carré, « carré 80 » un carré deux fois plus grand : un seul bloc, deux tailles différentes.

Remarque

Pour lire un programme qui contient un bloc personnalisé, il suffit de remplacer chaque appel du bloc par sa définition. C'est la même idée qu'en mathématiques quand on remplace $f(x)$ par son expression.

Les questions essentielles

1. Comment créer et utiliser une variable dans Scratch ?

Il faut d'abord créer la variable dans le menu « Variables », puis l'initialiser avec « mettre ... à ... ». On la modifie ensuite avec « ajouter ... à ... » ou en lui affectant une nouvelle valeur.

Voir la fiche méthode : Utiliser une variable dans Scratch

2. Comment choisir entre « si alors » et « si alors sinon » ?

On utilise « si ... alors » quand il n'y a rien de particulier à faire si la condition est fausse. On utilise « si ... alors ... sinon » quand il y a deux cas distincts à traiter.

Voir la fiche méthode : Utiliser une instruction conditionnelle dans Scratch

3. Comment choisir entre « répéter N fois » et « répéter jusqu'à » ?

Si le nombre de répétitions est connu à l'avance, on utilise « répéter N fois ». Si on ne sait pas combien de fois il faudra répéter (on attend qu'une condition soit vérifiée), on utilise « répéter jusqu'à ».

Voir la fiche méthode : Utiliser une boucle dans Scratch

4. Comment traduire un programme de calcul en Scratch ?

On demande le nombre de départ avec « demander … et attendre », on le stocke dans une variable, puis on traduit chaque opération par un bloc Scratch. On peut aussi écrire l'expression algébrique correspondante en notant x le nombre de départ.

Voir la fiche méthode : Traduire un programme de calcul en Scratch

5. Comment dessiner un polygone régulier avec Scratch ?

On utilise le stylo et une boucle « répéter n fois » avec les blocs « avancer » et « tourner ». L'angle de rotation est $\frac{360}{n}$ degrés, où n est le nombre de côtés.

Voir la fiche méthode : Dessiner une figure géométrique avec Scratch

6. Comment créer et utiliser un bloc personnalisé dans Scratch ?

On crée le bloc dans la catégorie « Mes blocs » avec « Ajouter un bloc », on place les instructions sous le chapeau « définir », puis on

appelle le bloc dans le programme principal. On peut ajouter un paramètre pour adapter le bloc à chaque appel.

Voir la fiche méthode : Créer et utiliser un bloc personnalisé dans Scratch

Télécharger en PDF