Scratch et algorithmes

DURÉE ESTIMÉE

25 minutes

Un algorithme est une suite d'instructions permettant de résoudre un problème. Le logiciel Scratch permet de coder ces algorithmes sous forme de programmes visuels, en assemblant des blocs colorés.

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
- Le multiplier par 2

- Ajouter 7 au résultat
- Annoncer le résultat

Pour le nombre de départ 5 : on obtient $5 \times 2 + 7 = 17$.
Pour le nombre de départ 10 : on obtient $10 \times 2 + 7 = 27$.

Remarque

Un algorithme se compose toujours de trois étapes :

- les **données d'entrée** (les informations de départ) ;
- le **traitement** (les instructions à exécuter) ;
- la **sortie** (le résultat obtenu).

2 - L'environnement Scratch

Scratch

Scratch est un logiciel de programmation visuelle. On y construit des programmes en assemblant des **blocs** colorés dans une zone de script.

L'interface de Scratch se compose de plusieurs zones :

- la **scène** : zone où le lutin exécute le programme ;
- la **zone de script** : espace où l'on assemble les blocs ;

- la **palette de blocs** : liste des blocs disponibles, classés par catégories.

Catégories de blocs

Les blocs sont classés en catégories identifiées par leur couleur :

Remarque

Pour démarrer un programme, on utilise un bloc chapeau du menu Evénements. Le plus courant est :

On clique ensuite sur le drapeau vert au-dessus de la scène pour lancer le programme.

3 - Déplacements et tracés

Dans Scratch, le lutin se déplace sur la scène et peut dessiner en utilisant le **stylo**.

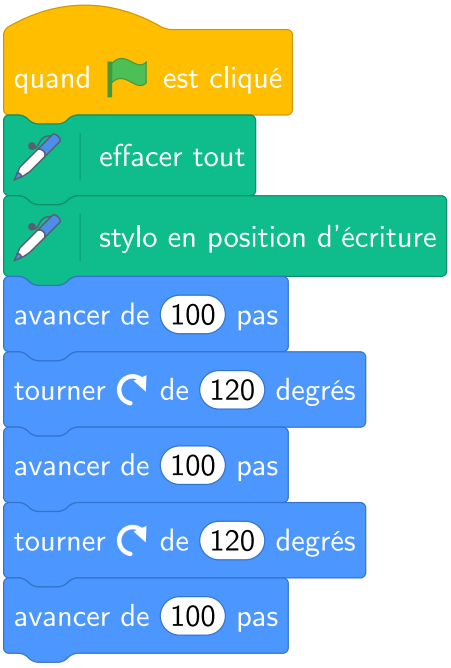
Blocs de déplacement et de tracé

Les principaux blocs de déplacement et de tracé sont :

- **avancer de ... pas** : fait avancer le lutin dans la direction courante ;
- **tourner de ... degrés** : fait pivoter le lutin (vers la droite ou vers la gauche) ;
- **stylo en position d'écriture** : le lutin laisse une trace quand il se déplace ;
- **relever le stylo** : le lutin se déplace sans dessiner ;
- **effacer tout** : efface tous les tracés de la scène.

Exemple

Ce programme trace un triangle équilatéral de 100 pas de côté :



Télécharger le projet Scratch .sb3 ?

L'angle extérieur d'un triangle équilatéral vaut $\frac{360}{3} = 120$ degrés. Le lutin tourne donc de 120 degrés après chaque côté.

Attention

Dans Scratch, l'angle de rotation correspond à l'**angle extérieur** de la figure, pas à l'angle intérieur. Pour un polygone régulier à n côtés, l'angle de rotation est $\frac{360}{n}$ degrés.

4 - 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.

Dans Scratch, les blocs de variables permettent de :

- **créer** une variable (lui donner un nom) ;
- **mettre ... à ...** : affecter une valeur à la variable ;
- **ajouter ... à ...** : augmenter la valeur de la variable.

Exemple

Le programme suivant crée une variable « score », lui donne la valeur 0, puis l'augmente de 10 :

Télécharger le projet Scratch .sb3 ?

À la fin du programme, le lutin affiche 10.

Attention

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

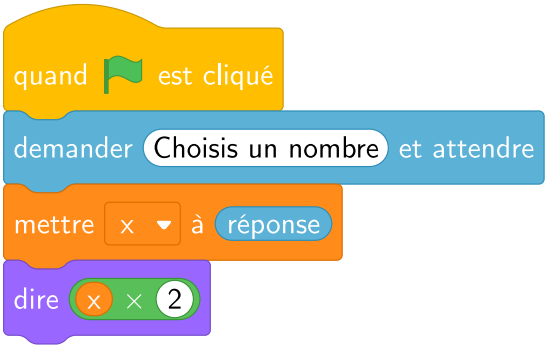
Entrée et sortie

Pour communiquer avec l'utilisateur :

- **Entrée** : le bloc « demander … et attendre » (menu Capteurs) affiche une question et stocke la réponse dans la variable spéciale « réponse » ;
- **Sortie** : le bloc « dire … » (menu Apparence) affiche un message dans une bulle.

Exemple

Ce programme demande un nombre à l'utilisateur et affiche son double :



Projet : double d'un nombre .sb3

Si l'utilisateur entre 7, le lutin affiche 14.

Tableau de valeurs pas à pas

Pour suivre l'exécution d'un programme, on peut construire un **tableau de valeurs** : on note la valeur de chaque variable après chaque instruction.
Soit le programme suivant :

Projet : évolution de deux variables .sb3

Ligne	A	B
2	3	
3		5
4	8	5

À la fin, la variable A contient $3 + 5 = 8$.

5 - Instructions conditionnelles

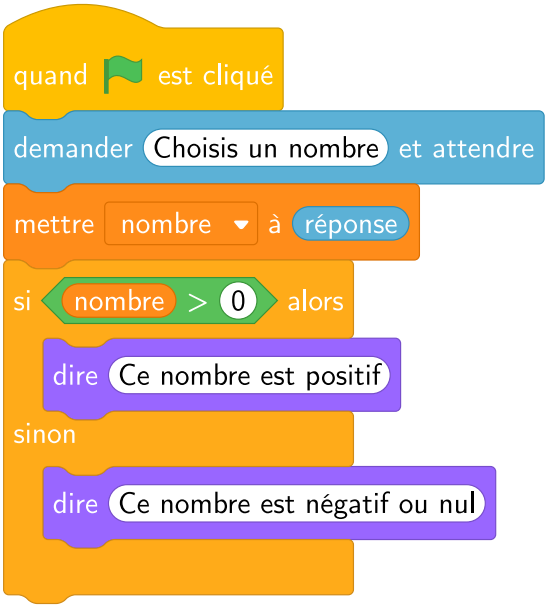
Instruction conditionnelle

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 s'exécutent que si la condition est vraie ;
- **si ... alors ... sinon** : un premier bloc s'exécute si la condition est vraie, un second si elle est fausse.

Exemple

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



Projet : positif ou négatif .sb3 ?

Remarque

Une condition est un test qui renvoie « vrai » ou « faux ». Les opérateurs de comparaison les plus courants sont : $=$, $<$ et $>$. On peut aussi combiner des conditions avec les opérateurs logiques « et », « ou » et « non ».

6 - 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 fixé 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 en utilisant une boucle :

Télécharger le projet Scratch .sb3

Un carré a 4 côtés et chaque angle extérieur mesure $\frac{360}{4} = 90$ degrés. La boucle « répéter 4 fois » évite de recopier les mêmes instructions quatre fois.

Exemple

Ce programme calcule la somme $1 + 2 + 3 + \ldots + 10$:

Projet : somme de 1 à 10 .sb3

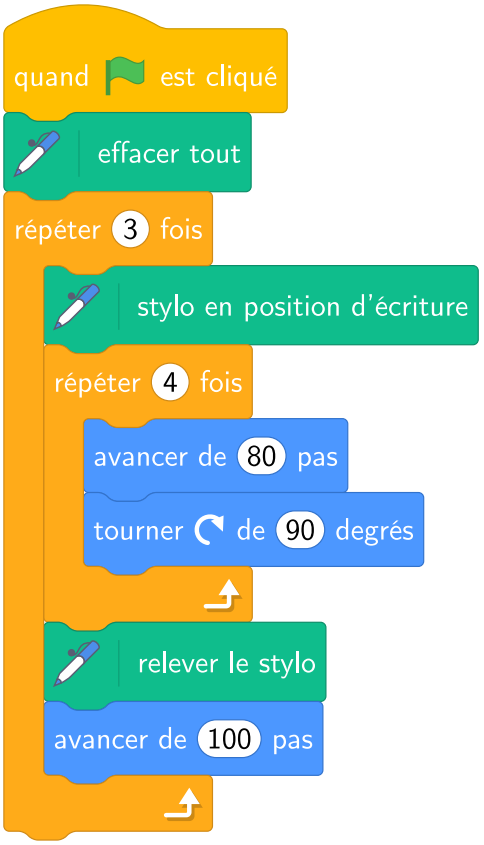
La boucle s'arrête quand le compteur dépasse 10. La variable « somme » contient alors $1 + 2 + \ldots + 10 = 55$.

Boucles imbriquées

On peut placer une boucle à l'intérieur d'une autre boucle : on parle de **boucles imbriquées**. La boucle intérieure s'exécute entièrement à chaque passage de la boucle extérieure.

Exemple

Ce programme trace 3 carrés de 80 pas de côté, espacés de 100 pas :



Télécharger le projet Scratch .sb3

La boucle extérieure se répète 3 fois (un passage par carré). A chaque passage, la boucle intérieure trace un carré complet (4 côtés), puis le lutin

avance sans dessiner pour se placer au début du carré suivant.

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 sera faux.

7 - Programmes de calcul

Programme de calcul

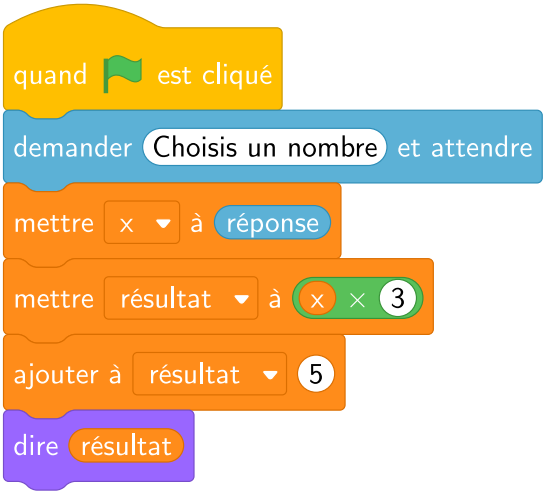
Un **programme de calcul** est une suite d'opérations mathématiques appliquées à un nombre de départ. Il peut se traduire en programme Scratch ou en expression algébrique.

Exemple

Soit le programme de calcul :

- Choisir un nombre x
- Le multiplier par 3
- Ajouter 5 au résultat

Sa traduction en Scratch :



Projet : programme 3x + 5 .sb3

L'expression algébrique correspondante est $3x + 5$.

Vérification pour $x = 4$: $3 \times 4 + 5 = 12 + 5 = 17$.

Remarque

Pour traduire un programme de calcul en Scratch :

- demander le nombre de départ et le stocker dans une variable ;
- appliquer chaque opération dans l'ordre, en utilisant une variable « résultat » ;
- afficher le résultat final avec le bloc « dire ».

Les questions essentielles

1. Comment construire une figure avec Scratch ?

Il faut utiliser les blocs du stylo (effacer tout, stylo en position d'écriture) et les blocs de mouvement (avancer, tourner). Pour un polygone régulier à n côtés, on utilise une boucle « répéter n fois » avec un angle de rotation de $\frac{360}{n}$ degrés.

Voir la fiche méthode : Construire une figure géométrique avec Scratch

2. Comment utiliser une variable dans Scratch ?

Il faut créer la variable dans le menu Variables, l'initialiser avec « mettre ... à ... », puis la modifier avec « ajouter ... à ... » ou en lui affectant une nouvelle valeur. On peut suivre l'évolution des variables avec un tableau de valeurs pas à pas.

Voir la fiche méthode : Utiliser une variable dans Scratch

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

On utilise « si ... alors » quand il n'y a rien de particulier à faire lorsque 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

4. Comment choisir le bon type de boucle ?

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 et qu'on attend qu'une condition soit vérifiée, on utilise « répéter jusqu'à ».

Voir la fiche méthode : Utiliser une boucle dans Scratch

5. 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 du programme par le bloc Scratch correspondant (opérateurs et variables).

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

Télécharger en PDF