

# Apprendre à programmer avec Scratch

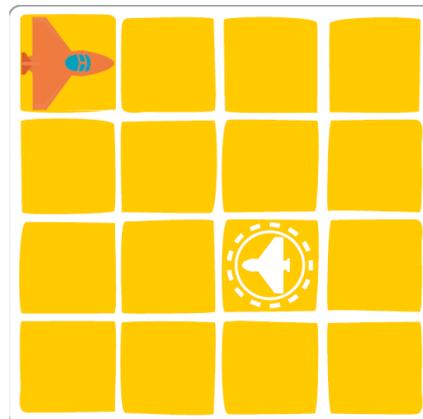
## 1. Le job d'un programmeur, c'est quoi au juste ?

Nous faisons connaissance avec la programmation sans ordinateur...

### Activité 1 : L'algorithme <sup>1</sup>, c'est comme une recette.

Un petit vaisseau doit parcourir une trajectoire simple sur une grille pour atteindre une destination donnée.

Tu es le capitaine tu dois donner un plan de route (des instructions) au pilote du vaisseau pour qu'il atteigne l'objectif orienté comme indiqué.



#### Le plan de route.

Au crayon, note ci-dessous une liste d'instructions simples que tu vas fournir au pilote. Tu n'es pas obligé de remplir toutes les lignes mais tu dois t'assurer que le pilote arrivera bien à destination à l'aide de ton plan de route.

.....

.....

.....

.....

.....

.....

Compare ce que tu as noté avec tes camarades et discutes-en avec eux. Corrige et améliore ta liste d'instructions si nécessaire. Cette liste d'instructions, c'est un **algorithme**.

**Un algorithme**, c'est un ensemble d'instructions qui décrivent toutes les étapes à réaliser pour atteindre un objectif.

Voici d'autres exemples d'algorithme :

- Une recette pour faire de crêpes.
- La procédure pour tracer un cercle à l'aide d'un compas.
- Les instructions de montage d'un meuble fourni par une marque suédoise bien connue.

Trouve au moins trois autres exemples d'algorithmes.

- ...
- ...
- ...

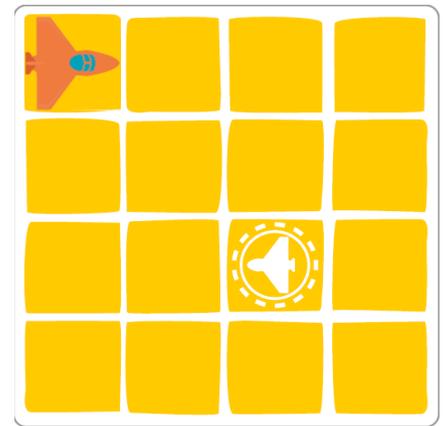
<sup>1</sup> Les grilles et graphismes sont empruntés à <https://cdn.reseau-canope.fr/archivage/valid/contenus-associes-activites-pedagogiques-N-18069-29384.pdf>

## Activité 2 : en pilote automatique.

On remplace le pilote du vaisseau par un ordinateur qui va piloter le vaisseau.

L'ordinateur ne connaît que quatre instructions :

1. *DROITE* = Pivoter de 90° à droite
2. *GAUCHE* = Pivoter de 90° à gauche
3. *AVANCER* = Avancer d'une case
4. *RÉPÉTER 4 x* = répéter 4 x (tu peux changer le nombre...)



Pour l'exemple précédent, voici le programme :

*AVANCER*  
*AVANCER*  
*DROITE*  
*AVANCER*  
*AVANCER*  
*DROITE*

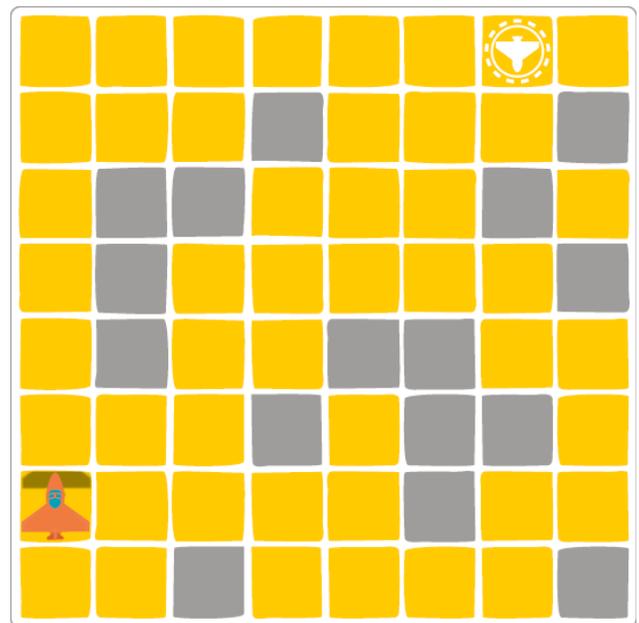
Le même, avec une boucle :

*RÉPÉTER 2 x*  
*AVANCER*  
*AVANCER*  
*DROITE*

Un **programme**, c'est une série d'instructions qui peuvent être comprises et exécutées par une machine, par exemple par un ordinateur.

Voici une nouvelle mission. À toi d'écrire le programme en utilisant le moins d'instructions possible.

.....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....



Combien d'instructions utilises-tu ? Compare avec tes camarades.

Il y a moyen d'écrire le programme avec seulement 4 instructions. Essaie de trouver comment !

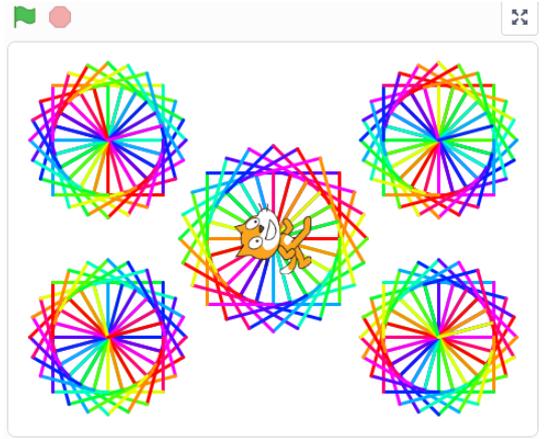
Alors, au final, comment décrirais-tu le travail d'un programmeur ?

.....  
 .....  
 .....  
 .....  
 .....  
 .....

## 2. Prêt à programmer ?

Avec Scratch<sup>2</sup>, nous avons appris à un personnage à tracer un carré. Ensuite, nous avons demandé à l'ordinateur d'en dessiner plusieurs pour obtenir ce genre d'effet . Nous avons aussi dessiné d'autres polygones.

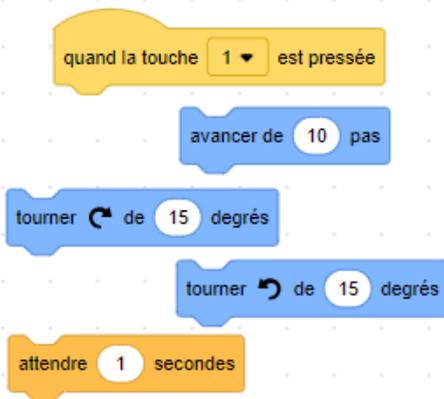
### Activité 3 : un petit scarabée effectue une trajectoire carrée



Scratch propose un **langage de programmation**.

Ce langage se compose d'instructions codées sous forme de blocs que l'ordinateur est capable de comprendre et d'exécuter.

Avec ces blocs,



nous avons composé ce programme,



ou mieux encore, celui-ci avec une boucle « **répéter 4 fois** »:



Nous avons encore utilisé d'autres blocs pour placer correctement le petit scarabée au milieu de l'écran et pour dessiner un trait de couleur.

Le programme complet est à la page suivante.

Tu peux retrouver tous ces programmes ici :

<https://scratch.mit.edu/projects/1072097200>

Le langage Scratch est très accessible. Les blocs sont faciles à manipuler et chaque programme peut se lire comme on lit un texte.

Le petit programme ci-dessus, en langage Python, cela donnerait par exemple ceci :

```
import turtle

# Créer une fenêtre
fenetre = turtle.Screen()
fenetre.setup(width=500, height=500)

# Créer une tortue
tortue = turtle.Turtle()
tortue.speed(1) # Ajuste la vitesse (0: le plus rapide, 10: le plus lent)

# Définir la couleur du tracé
tortue.color("red")

# Définir la longueur des côtés
cote = 100 # 100 pixels équivalent à 10 cm (vous pouvez ajuster cette va

# Tracer le carré
for _ in range(4):
    tortue.forward(cote)
    tortue.right(90)

# Garder la fenêtre ouverte
turtle.done()
```

<sup>2</sup> <https://scratch.mit.edu>

# 3. On récapitule !

## La mission

Le petit scarabée est placé au milieu de l'écran. Il doit décrire une trajectoire carrée et la tracer.

## L'algorithme

Pour parcourir une trajectoire carrée, le personnage devra :

1. Se déplacer de 10cm en ligne droite
2. Tourner sur lui-même de 90° dans un sens déterminé
3. ... et répéter ces deux opérations à 4 reprises afin d'obtenir une figure fermée.

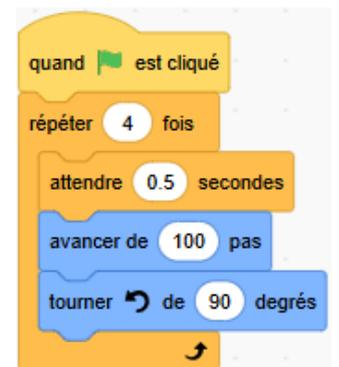
Pour présenter un algorithme, on peut utiliser un **logigramme** comme celui-ci.



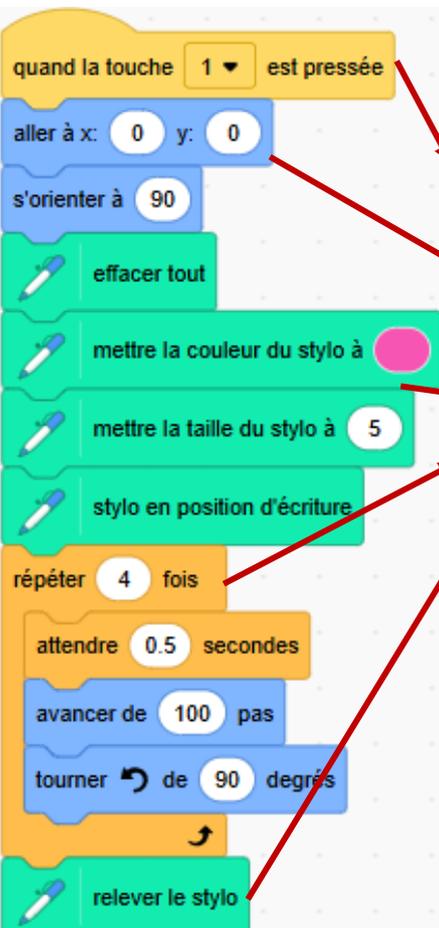
## Le langage de programmation

Une fois qu'on passe sur l'ordinateur, il faut utiliser un langage de programmation qu'il peut comprendre. Scratch propose des blocs comme ceux-ci.

## Le programme dans sa version la plus simple



Dans ce cas, le petit personnage (sprite) se déplace en décrivant un carré.



## Le programme avec initialisation et le tracé en couleur

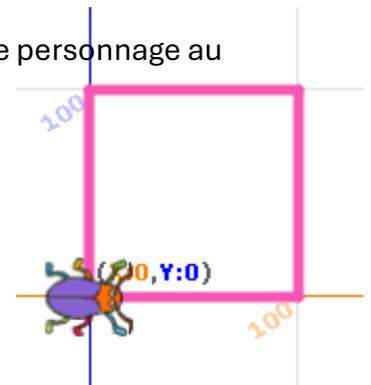
**Bloc « événement »** pour démarrer le programme

**Initialisation** : pour placer correctement le personnage au démarrage.

Préparation du **tracé**

**Boucle principale**

**Fin** du tracé



Tu peux continuer chez toi, créer de chouettes applications.

Avec un navigateur, sur tablette ou sur ordinateur, rends-toi sur <https://scratch.mit.edu>. Clique sur « Nous rejoindre » pour te créer un compte et tes projets seront mémorisés dans le Cloud.