

I- Introduction

Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat.

La **programmation** consiste à traduire un algorithme dans un langage que comprend l'exécutant, qu'il s'agisse d'une personne, d'un ordinateur, d'un robot...

Pour ce cours, nous utiliserons un « robot » (exécutant) qui produira ce genre de dessins :



II - Qu'est-ce qu'une instruction ?

1°) Définition

Une instruction est un ordre qu'on donne à l'exécutant.

Pour notre robot, on utilisera les instructions suivantes :

- Droite** → déplace le robot une case à droite ;
- Gauche** → déplace le robot une case à gauche ;
- Colorie** → colorie la case sous laquelle est le robot.

Exemple :

Le programme ci-contre réalise le dessin suivant



Colorie
Droite
Colorie
Droite
Droite
Droite
Colorie
Droite
Colorie

2°) Instruction à effet global

Il est possible d'utiliser des instructions qui ont un effet durable (plus que juste la ligne où elles sont écrites).

Par exemple :

Stylo en écriture → colorie automatiquement la où le robot se trouve, même après déplacement, jusqu'à ce qu'on donne l'instruction :

Relever le style → arrête le coloriage automatique.

Exemple :



Stylo en écriture
Droite
Relever le stylo
Droite
Droite
Droite
Stylo en écriture
Droite

3°) Instruction à paramètre

Certaines instructions peuvent prendre un (ou des) paramètres.

Droite(<nombre>) → déplace le robot de <nombre> cases à droite

Exemple :



Stylo en écriture
Droite
Relever le stylo
Droite(3)
Stylo en écriture
Droite

→ Exercices 1 et 2 de la fiche

→ Activité à plusieurs : sur le cahier, dessiner un motif (maximum 12 cases). Sur un quart de feuille, écrire le programme de votre motif. Échanger ensuite ce programme avec un autre élève et exécuter le programme de cet élève. Comparer vos résultats.

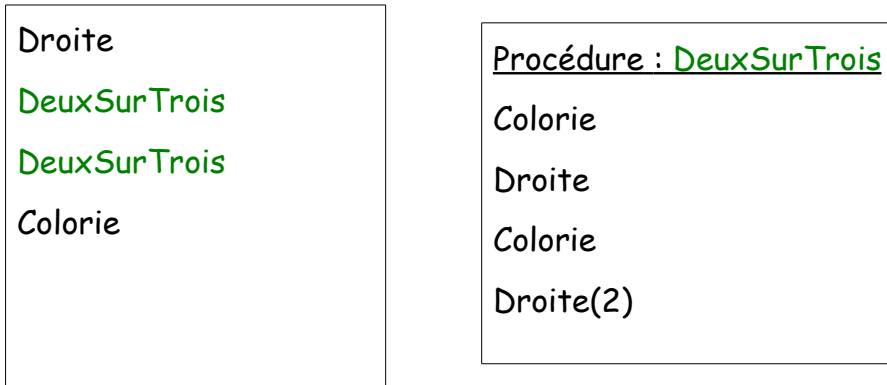
III - Procédures, répétitions

Il existe des méthodes pour éviter d'écrire plusieurs fois la même chose dans un programme :

1°) Créer ses propres instructions : une procédure

Une procédure est un « sous-programme » qui a un nom et qui regroupe plusieurs instructions pour en créer une nouvelle.

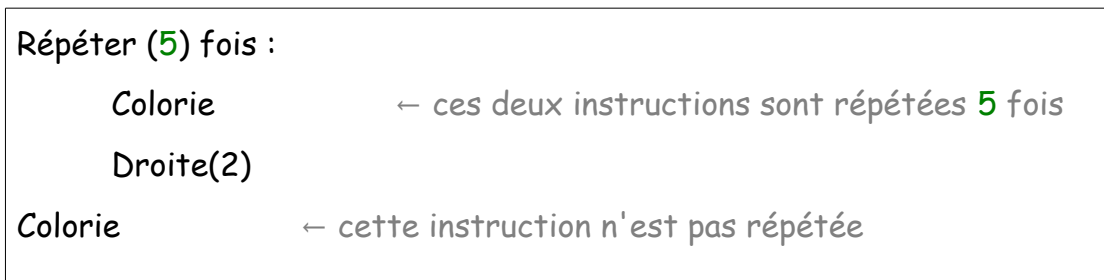
Exemple :



2°) Répéter plusieurs instructions



Une « boucle Répéter » permet de l'écrire facilement :



Pour indiquer quelles instructions doivent être répétées, on les « décale » vers la droite. Ce décalage s'appelle l'indentation. C'est un concept TRES IMPORTANT en programmation.

→ Exercice 3 de la fiche

IV - Variables, entrées/sorties et conditions

1°) La mémoire : les variables

Un exécutant (humain, ordinateur, robot,...) peut mémoriser une information dans une **variable**. Une variable possède un **nom** et une **valeur** qui est généralement un nombre. En programmation on privilégiera des noms de variable explicites.

Par exemple la variable « age » peut contenir le nombre 14, et la variable « prénom » peut contenir le texte « Ludovic ».

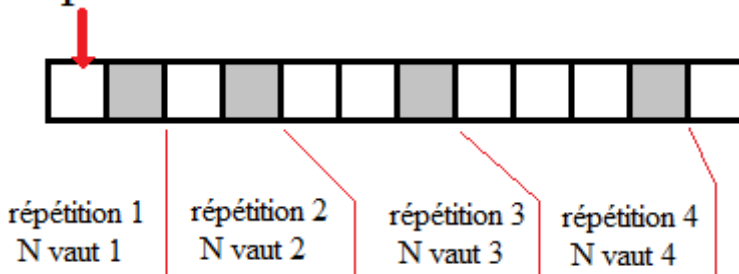
On utilisera l'instruction :

Mettre (<variable>) à (<valeur>) → La variable <variable> contient maintenant la valeur <valeur>.

Exemple :

Mettre N à 1	← la variable N vaut 1
Répéter (4) fois	
Droite(N)	← on avance de N cases
Colorie	
Mettre N à N+1	← la variable N augmente de 1

départ



→ Exercice 4 de la fiche

2°) Instruction d'entrée / sortie

Définition : l'utilisateur est celui à qui l'exécutant s'adresse (donc vous).

Une instruction peut demander une information à l'utilisateur (entrée) :

Demander la valeur de (<variable>) → l'utilisateur choisi la valeur de <variable>

Une instruction peut donner une information à l'utilisateur (sortie) :

Dire (texte ou <variable>) → écrit le texte indiqué ou la valeur de <variable>

Dans notre cas, on écrira ce texte à côté du motif du robot.

Exemple:

```
Dire("Combien de cases ?")
Demander la valeur de (nombre)
Sylo en écriture
Droite(nombre - 1)
Relever style
Droite
Dire("J'ai colorié " + nombre + "cases.")
```

Motif obtenu :



Messages obtenus :

```
Combien de cases ?
4 ← choix de l'utilisateur
J'ai colorié 4 cases.
```

→ Exercice 5 de la fiche

3°) Conditions, instruction conditionnelle

On introduit une nouvelle structure de programme : Si (...) alors ...
et sa variante : Si (...) alors ... sinon ...

```
Dire("Quel âge as-tu ?")
Demander la valeur de (age)
Si (age est plus petit que 18) alors      ← on écrira age < 18 par la suite
    Dire("Tu es mineur.")      ← ces lignes ne seront exécutées que si age < 18
    Dire("Car tu as moins de 18 ans.")
Sinon
    Dire("Tu es majeur.")      ← sinon c'est cette ligne qui sera exécutée
```

NE PAS OUBLIER L'INDENTATION ! C'est à dire l'alinéa qui permet de savoir jusqu'où s'étend la condition.

→ Exercice 6 de la fiche

→ Problèmes de la fiche