

## 1) Qu'est-ce qu'un algorithme ?

C'est une **méthode** pour résoudre systématiquement un type de problème.

Un **algorithme** est dit correct s'il résout à chaque fois le problème posé.

Un algorithme comporte en général une **suites d'instructions** à effectuer dans un ordre précis et se termine en un nombre fini d'étapes.

Un algorithme s'écrit en **langage naturel**.

Ce n'est pas un programme.

Un **programme** est la traduction d'un algorithme dans un **langage informatique** particulier.

Les instructions possibles sont :

Des **affectations de variables**

Des **conditions**

Des **boucles**

## 2) Affectations de variables :

On écrira par exemple :

x prend la valeur 5

pi ← 3.14

prenom = 'inconnu'

Une **variable** est un **emplacement** dans la **mémoire** de l'ordinateur (adresse mémoire).

Python associe au **nom** d'une variable une **adresse** dans sa **mémoire**, dans laquelle est stockée la **valeur** de la variable.

Remarque : Pour faire augmenter une variable de 1, on peut écrire :

x prend la valeur x + 1

x ← x + 1

x = x + 1

**A gauche du signe égal se trouve la nouvelle valeur de la variable x,**  
**et à droite du signe égal se trouve la valeur précédente de la variable x.**

### 3) Instruction conditionnelle :

<p><b>Si une condition est vraie</b>, faire ce qui suit : des instruction ici...</p> <p><b>Sinon</b> faire ce qui suit : d'autres instructions ici...</p> <p>Des instructions qui seront <b>toujours faites</b> ici...</p>	<p>Exemple :</p> <p>n est un nombre entier</p> <p><b>Si le nombre n est pair</b> faire : <math>n \leftarrow n / 2</math></p> <p><b>Sinon</b> faire : <math>n \leftarrow 3 * n + 1</math></p> <p>Afficher n</p>
--	--

Remarque : Le bloc d'instructions **Sinon** n'est pas obligatoire.  
Dans ce cas, rien n'est fait lorsque la condition n'est pas vraie.

#### Exercice : Valeur absolue d'un nombre réel

La valeur absolue d'un réel  $x$  est définie de la manière suivante :  $|x| = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$ .

La valeur absolu de  $x$  est donc toujours un nombre positif.

Elle correspond à la **distance** entre  $x$  et 0 :  $d(x, 0) = |x|$ .

La distance entre deux réels  $x$  et  $y$  est alors donnée par :  $d(x, y) = |x - y|$ .

Ecrire un algorithme qui renvoie la valeur absolu d'un réel  $a$ .

Voici une forme plus complète de l'instruction conditionnelle :

<p><b>Si une condition est vraie</b> faire :</p> <p>instructions1</p> <p><b>Sinon si une autre condition est vraie</b> faire :</p> <p>instructions2</p> <p><b>Sinon</b> faire :</p> <p>instructions3</p> <p>Des instructions qui seront <b>toujours faites</b> ensuite...</p>
---

Exercice : On considère une fonction  $f$  définie sur  $\mathbb{R}$  de la manière suivante :

$$f(x) = \begin{cases} x+2 & \text{si } x \leq 0 \\ -x+2 & \text{si } 0 < x \leq 2 \\ 2x-4 & \text{si } x > 2 \end{cases}$$

Ecrire un algorithme qui renvoie la valeur de  $f(x)$  quelle que soit la valeur entrée pour  $x$ .

#### 4) Boucle bornée :

On souhaite répéter un **bloc** contenant **plusieurs instructions** un nombre de fois **connu à l'avance**.

Pour cela, on utilise ce qu'on appelle une **boucle bornée**, c'est-à-dire avec un début et une fin connus.

<p><b>Répéter n fois</b> ce qui suit dans le bloc :</p> <p>instruction1 instruction2 ... Les instructions qui suivent ici...</p>	<p>Exemple : On souhaite afficher la table de 9</p> <p><b>Pour i allant de 1 à 10</b> faire : afficher <math>9 * i</math></p> <p>La variable <math>i</math> s'appelle un <b>compteur</b></p>
--	--

Remarque : Pour créer une telle boucle, on utilise toujours une instruction utilisant un **compteur**, en précisant son début, sa fin, et son **pas** s'il est différent de 1. Ce compteur est une **variable** qui contient un nombre correspondant à l'étape actuelle.

On peut alors éventuellement utiliser cette variable dans le bloc contenant les instructions à répéter.

**Exercice** : On donne l'algorithme suivant :

<p><math>s \leftarrow 0</math> pour <math>k</math> allant de 1 à <math>n</math> faire : <math>s \leftarrow s + k</math></p>
---

- Combien vaut la variable  $s$  lorsqu'on choisit  $n=5$  ?
- Que calcule cet algorithme ?

**Exercice** : Ecrire un algorithme qui renvoie la valeur de la somme suivante en utilisant un compteur de valeur initiale zéro :

$$S_n = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2}$$

**Exercice** : Ecrire un algorithme qui renvoie la somme des entiers impairs de 1 à  $n$ .

## 5) Boucle non bornée :

On souhaite répéter un **bloc** contenant **plusieurs instructions** un nombre de fois **non connu à l'avance**.

Pour savoir quand arrêter de parcourir ce bloc d'instructions, on fixe **dès le départ** une **condition** qui doit être respectée.

Lorsque la condition n'est plus respectée, on sort de la boucle.

Pour cela, on utilise ce qu'on appelle une **boucle non bornée**.

Tant qu' <b>une condition est vraie</b> faire : instruction1 instruction2 ... Les instructions suivantes	Pour afficher le quotient d'un entier dans la division par 2 :  n est un nombre entier positif donné quotient $\leftarrow$ 0 <b>tant que</b> $n \geq 2$ faire : quotient $\leftarrow$ quotient + 1 $n \leftarrow n - 2$ afficher quotient
--	--

**Exercice** : Ecrire un algorithme affichant un compte à rebours à partir de 10.

**Exercice** : **Partie entière d'un nombre réel**

La partie entière d'un réel  $x$  est définie comme le plus grand entier inférieur ou égal à  $x$ . La partie entière de  $x$  se note  $E(x)$  en mathématiques.

Ainsi,  $n = E(x)$  est équivalent à dire que  $n \in \mathbb{Z}$  et que  $n \leq x < n + 1$ .

- Ecrire un algorithme qui renvoie la partie entière d'un réel positif  $x$ .
- Ecrire un algorithme qui renvoie la partie entière d'un réel négatif  $x$ .
- Ecrire un algorithme qui renvoie la partie entière d'un réel  $x$ .

Remarque :

On peut toujours remplacer une boucle bornée par une boucle non bornée en créant un compteur à **initialiser** au préalable.

Pour afficher la table de 9 on aurait pu faire ainsi :

```
i ← 1
tant que i ≤ 10 faire :
    afficher 9 * i
    i ← i + 1
```

Il ne faut pas oublier d'**augmenter le compteur** à chaque étape, **sinon** la boucle ne s'arrête jamais.