

Cette page présente les instructions usuelles découvertes au cours de cette formation, accompagnées de leur équivalent en pseudo-code algorithmique.

I – Affecter une variable

Interprétation	Pseudo-code	Python
<i>x prend la valeur 3</i>	$x \leftarrow 3$	<code>x = 3</code>

Les types usuelles des valeurs manipulées en Python sont :

- Les entiers (*type `int`*)
- Les flottants (*type `float`, assimilables aux décimaux*)
- Les booléens (*type `bool`*) de valeurs **True** ou **False**, résultats d'un test
- Les chaînes de caractères (*type `str`*) qui représentent les textes
- Les listes (*type `list`*) qui sont des séquences d'objets

II – Définir une fonction

Interprétation	Pseudo-code	Python
$f(x) = 2x + 3$	Définir $f(x)$ Renvoyer $2x + 3$	<code>def f(x):</code> <code> return 2*x+3</code>

Attention, les deux points et l'indentation sont indispensables...

III – Faire un test

Interprétation	Pseudo-code	Python
<i>x est-il égal à y ?</i>	$x = y$	<code>x == y</code>
<i>x est-il distinct de y ?</i>	$x \neq y$	<code>x != y</code>
<i>x est-il strictement supérieur à y ?</i>	$x > y$	<code>x > y</code>
<i>x est-il supérieur ou égal à y ?</i>	$x \geq y$	<code>x >= y</code>
<i>x est-il strictement inférieur à y ?</i>	$x < y$	<code>x < y</code>
<i>x est-il inférieur ou égal à y ?</i>	$x \leq y$	<code>x <= y</code>

IV – Instructions conditionnelle

Remarque	Pseudo-code	Python
<i>La partie « sinon » est optionnelle</i>	Si $A > 0$ Alors $A \leftarrow -A$ Sinon $A \leftarrow f(A)$ Fin Si	<code>if A > 0:</code> <code>A = -A</code> <code>else:</code> <code>A = f(A)</code>

A nouveau, les deux points et l'indentation sont indispensables. S'il y a plusieurs instructions imbriquées, on peut utiliser **elif**, contraction de « **else if** ».

V – Boucle itérateur

Pseudo-code	Python	Entiers parcourus
Répéter 20 fois $A \leftarrow A$ Fin répétition	<code>for i in range(20):</code> <code>A = A</code>	De 0 à 19...
Pour i allant de 1 à N : $A \leftarrow A+i$ Fin Pour	<code>for i in range(1, N+1):</code> <code>A = A + i</code>	De 1 à N ...

range(a, b, c) génère les entiers de **a** (*inclus*) à **b** (*exclus*), avec un pas de **c**. Par défaut, **a = 0** et **c = 1** donc écrire **range(n)** signifie **range(0, n, 1)**.

VI – Boucle conditionnelle

Remarque	Pseudo-code	Python
<i>Lorsqu'on passe sur machine, attention aux « boucles infinies »...</i>	Tant que $n > 1$ $n \leftarrow \text{reste de } n/q$ Fin Tant Que	<code>while n > 1:</code> <code>n = n%q</code>

VII – Modules complémentaires utiles

- **math** : contient les fonctions mathématiques les plus communes.
Pour importer les fonctions trigo. : `from math import sin, cos, tan`

- **random** : pour générer des nombres pseudo-aléatoires.
Pour un décimal dans $[0 ; 1[$: `from random import random`
Pour un entier dans $[a ; b]$: `from random import randint`