

Menu
Accueil
Cours
Exercices
Contrôles
QCM
Quiz
Scratch
Problèmes
Brevet
Bac
Post-bac
LIVRES

Algorithmique et programmation en Python

> Python sert ici d'outil mathématique : automatiser un calcul répétitif, tester une conjecture, simuler une expérience aléatoire.

L'ESSENTIEL EN TROIS POINTS

1. L'affectation $=$ n'est pas une égalité : elle range une valeur dans une variable.
2. La boucle `for` répète un nombre connu de fois, la boucle `while` tant qu'une condition est vraie.
3. `range(n)` parcourt les entiers de 0 à $n-1$, jamais n.

Variables et affectation

À SAVOIR — LE SIGNE ÉGAL

`x = x + 1` n'est pas une équation : l'instruction lit la valeur de x, ajoute 1, et range le résultat dans x.

PIÈGE À ÉVITER

En mathématiques, $x = x+1$ n'a pas de solution. En Python, c'est une instruction parfaitement valide. Les deux $=$ n'ont pas le même sens.

Les boucles

COMMENT FAIRE — FOR ET WHILE

`for i in range(5):` exécute le bloc cinq fois, avec i valant $0,1,2,3,4$.

`while condition:` répète tant que la condition reste vraie.

PIÈGE À ÉVITER

`range(5)` s'arrête à 4. Pour parcourir 1 à 10, il faut écrire `range(1, 11)`.

EXEMPLE GUIDÉ

Somme des entiers de 1 à 100 :

```
s = 0
for i in range(1, 101):
  s = s + i
print(s)
```

Les conditions

À SAVOIR — IF, ELIF, ELSE

`if` teste une condition, `elif` en teste une autre si la précédente était fausse, `else` couvre tous les cas restants. L'égalité se teste avec ==, deux signes.

Les fonctions

COMMENT FAIRE — DÉFINIR ET APPELER

`def f(x):` puis, en dessous et indenté, `return 2*x + 3`. On l'utilise ensuite en écrivant `f(5)`.

ASTUCE

Une fonction Python sans `return` ne renvoie rien : elle peut afficher une valeur sans qu'on puisse la réutiliser dans un calcul.

Simuler le hasard

À SAVOIR — LE MODULE RANDOM

`from random import randint` puis `randint(1, 6)` tire un entier au hasard entre 1 et 6 inclus. En répétant beaucoup de tirages, la fréquence observée s'approche de la probabilité.

VÉRIFIEZ QUE VOUS SAVEZ FAIRE

- distinguer affectation et égalité mathématique
- écrire une boucle for avec le bon range

- écrire une boucle while qui se termine
- définir et appeler une fonction Python
- simuler une expérience aléatoire et comparer fréquence et probabilité

On nous pose souvent ces questions

Pourquoi peut-on écrire x = x + 1 en Python ?

Parce que le signe égal n'est pas une égalité mais une affectation : l'instruction lit la valeur de x, ajoute 1, et range le résultat dans la même variable.

Que parcourt range(5) ?

Les entiers de 0 à 4, jamais 5. Pour parcourir 1 à 10, il faut écrire range(1, 11).

Quand utiliser une boucle while plutôt qu'une boucle for ?

Quand le nombre de répétitions n'est pas connu d'avance et dépend d'une condition à atteindre.