MATHÉMATIQUES · SECONDE

Algorithmique avec Python

6 notions, l'essentiel à retenir et les pièges à éviter.

1 Variables et affectation

nom = valeur

- x = 5 range la valeur 5 dans la variable x : c'est une affectation, pas une égalité.
- Types : int (entier), float (nombre à virgule), str (texte), bool (True ou False).
- ** puissance, // quotient entier, % reste de la division.

2 Instructions conditionnelles

if condition:

- == compare, = affecte. Deux-points, puis bloc indenté de 4 espaces.
- Les tests if, elif sont lus dans l'ordre : seul le premier vrai s'exécute.
- and, or et not combinent des conditions.

3 Boucle bornée for

for i in range(n):

- range(n) donne 0 à n − 1, soit n tours ; range(a, b) va de a à b − 1.
- range(a, b, p) avance de p.
- Accumuler : initialiser avant la boucle, mettre à jour à chaque tour.

4 Boucle non bornée while

while condition:

- Répète tant que la condition est vraie, quand le nombre de tours est inconnu.
- Initialiser, modifier la variable testée, vérifier que la condition deviendra fausse.

5 Fonctions

def f(x): return résultat

- Les paramètres reçoivent les valeurs de l'appel, dans l'ordre.
- return renvoie une valeur et arrête la fonction ; print affiche seulement.
- Sans return, la fonction renvoie None.

6 Listes

L[0] … L[len(L) − 1]

- Indices à partir de 0 ; len(L) donne le nombre d'éléments.
- L.append(x) ajoute x à la fin ; for v in L: parcourt les valeurs.
- Maximum : partir de L[0], pas de 0. Moyenne : liste non vide.

Pièges à éviter

- Écrire = au lieu de == dans un test.
- Oublier que la borne de fin de range est exclue.
- Oublier les deux-points ou l'indentation d'un bloc.
- Laisser inchangée la variable testée par while : la boucle ne s'arrête pas.
- Confondre return, qui renvoie, et print, qui affiche.