Programmer et utiliser un tableur

Seconde professionnelle · Chapitre 08

Variables et expressions

`x = 3` affecte une valeur ; `x == 3` teste une égalité. `**` désigne une puissance, `/` une division, `//` un quotient entier et `%` le reste. `x = x + 2` augmente l'ancienne valeur de 2.

Choisir avec une condition

`if`, `elif`, `else` choisissent un bloc selon des tests. L'indentation délimite chaque bloc. `and` exige deux conditions, `or` au moins une, `not` inverse un booléen. Vérifie les valeurs limites : $<$ et $\leqslant$ ne donnent pas le même résultat à l'égalité.

Boucles comptées

`for k in range(a, b)` parcourt les entiers de a inclus à b exclu, par pas de 1. Initialise l'accumulateur : souvent 0 pour une somme, 1 pour un produit. Une trace indique l'état des variables après chaque passage.

Boucles conditionnelles

`while` répète tant que la condition est vraie. Pour atteindre un terme strictement supérieur à S, continue tant que `u <= S`. Mets à jour les variables et justifie l'arrêt. Distingue la valeur atteinte du nombre de passages.

Fonctions et listes

`def` définit une fonction ; `return` renvoie une valeur et termine l'appel. `print` affiche mais ne remplace pas `return`. Une liste de longueur n possède les indices 0 à n−1. `L[-1]` est son dernier élément si elle n'est pas vide. `L.append(x)` ajoute une valeur en fin de liste.

Tableur

Une formule commence par `=`. Une référence relative comme `A2` s'adapte lors d'une recopie ; `$A$2` fixe ligne et colonne. Dans `$A2`, seule la colonne est fixe ; dans `A$2`, seule la ligne est fixe. Contrôle la première formule et une cellule après recopie.

Exemple de trace

Initialise `u = 2`, puis répète trois fois `u = 2*u` : les valeurs sont 2, 4, 8, 16. Trois mises à jour donnent 16 ; pour une suite commençant en u_0, c'est u_3.

Avant de valider

Teste un cas simple et un cas limite. Vérifie les bornes de `range`, l'indentation, les indices et les unités. Explique le calcul : un résultat plausible ne suffit pas.