

Comment programmer une fonction en Python ?

LA RÉPONSE COURTE

Une fonction Python se déclare avec **def**, un nom, des paramètres entre parenthèses et deux-points. Le corps est **indenté**, et la valeur est renvoyée par **return** : sans lui, la fonction ne rend rien.

La méthode en 5 étapes

1. Écrire l'en-tête

`def f(x) :` — le nom, les paramètres entre parenthèses, et surtout les deux-points à la fin de la ligne.

2. Indenter le corps

Toutes les lignes de la fonction sont décalées de quatre espaces. L'indentation est la syntaxe en Python : elle délimite le bloc.

3. Calculer puis renvoyer

`return` renvoie le résultat et termine la fonction. Sans `return`, la fonction affiche éventuellement quelque chose mais ne renvoie rien d'utilisable.

4. Appeler la fonction

`f(3)` exécute la fonction avec $x = 3$. Pour voir le résultat dans la console, on écrit `print(f(3))`.

5. Tester sur des cas connus

On vérifie le programme sur des valeurs dont on connaît le résultat de tête, notamment 0 et un nombre négatif.

Un exemple entièrement résolu

Énoncé

Écrire une fonction Python qui renvoie l'image de x par $f(x) = 3x^2 - 5x + 2$, puis une fonction qui renvoie le prix TTC d'un prix hors taxes avec une TVA de 20 %.

La fonction mathématique

```
def f(x):  
    return 3*x**2 - 5*x + 2
```

L'étoile est obligatoire pour la multiplication, et `**` note la puissance.

Test

`print(f(2))` affiche 4, car $3 \times 4 - 10 + 2 = 4$. Le calcul à la main confirme le programme.

La fonction TTC

```
def ttc(prix):
    return prix * 1.2
```

Attention au séparateur

En Python, le séparateur décimal est le **point** : on écrit 1.2, jamais 1,2 qui serait interprété comme deux valeurs.

Conclusion. `f(2)` renvoie 4 et `ttc(50)` renvoie 60.0 : la fonction Python calcule exactement la même chose que la fonction mathématique.

Les erreurs à ne pas faire

Oublier les deux-points en fin d'en-tête, ou l'indentation du corps : le programme ne s'exécute pas.

Écrire `3x` au lieu de `3*x` : le signe de multiplication est obligatoire.

Utiliser la virgule comme séparateur décimal.

Confondre `print` et `return` : `print` affiche, `return` renvoie une valeur réutilisable dans un autre calcul.

3 exercices pour s'entraîner

Cherche d'abord sur ton cahier, sans regarder l'exemple ; le corrigé se déploie ensuite.

1. Exercice 1

Écrire une fonction Python qui renvoie le périmètre d'un rectangle de longueur `L` et de largeur `l`.

Voir le corrigé

```
def perimetre(L, l):
    return 2*(L + l)
```

Test : `perimetre(5, 3)` renvoie 16.

2. Exercice 2

Écrire une fonction qui renvoie le prix soldé d'un article après une remise de `r` pour cent.

Voir le corrigé

```
def solde(prix, r):
    return prix * (1 - r/100)
```

Test : `solde(60, 30)` renvoie 42.0.

3. Exercice 3

Que renvoie ce programme ?
`def g(x): /      y = x + 1 /      y = 2*y /      return y`
 — appelé avec $g(4)$.

Voir le corrigé

Ligne par ligne : y prend la valeur $4+1=5$, puis y prend la valeur $2 \times 5 = 10$.
 $g(4)$ renvoie 10. La fonction calcule $2(x+1)$.

Questions fréquentes

Quelle différence entre print et return ?

print affiche un texte dans la console : le résultat est perdu ensuite. return renvoie une valeur qu'on peut stocker ou réutiliser, par exemple dans $f(2) + f(3)$. Une fonction sans return renvoie None, ce qui provoque une erreur si on essaie de calculer avec.

Pourquoi Python en cours de maths ?

Parce qu'il permet de traiter des calculs répétitifs impossibles à la main : parcourir des centaines de valeurs pour chercher un seuil, simuler dix mille lancers de dé et comparer la fréquence à la probabilité théorique. C'est un outil du programme, présent dans les exercices de contrôle comme au baccalauréat.