

Python 7 - Les fonctions dans Python



1 - Principe des fonctions

Les fonctions sont des constructions particulières :

- qui reçoivent des données d'entrée et
- qui renvoient leur réponse en sortie

Version mathématique : $x \mapsto (a \cdot x + b)$

Ceci définit la fonction affine.

Version schématique :

5, 10, 50, 0, 20 $\Rightarrow$ Fonction MAXIMUM $\Rightarrow$ 50

2 - Fonctions natives en Python

Une fonction native est une fonction présente de base dans Python. On en **fait l'appel** en notant son nom et en fournissant entre parenthèses les entrées séparées par des virgules.

Fonction MAXIMUM : se nomme **max()** en Python.

```
>>> max(10, 0, 50, 40)
50
```

Fonction VALEUR ABSOLUE : se nomme **abs()**.

```
>>> abs(-12)
12
>>> abs(12)
12
```

3 - Définition de fonctions

Définition / déclaration : Définir une fonction veut dire la mettre en mémoire, pas à l'utiliser réellement. Une fonction se déclare en début de programme à l'aide du mot-clé **def**. On place ensuite le **nom** de la fonction, les **paramètres entre parenthèses** et un double-point : .

Exemple :

```
def fois2(a):
    b = a * 2
    return b
```

→ **Déclaration** de la fonction qui recevra un paramètre nommé localement **a**
 → On crée une variable locale **b** qui sera supprimée à la fin de la fonction
 → On **renvoie** le résultat et **on sort définitivement**.

Notez bien que ce programme ne fait RIEN à part mettre la fonction fois2() en mémoire.

4 - Appel d'une fonction

Appel : Faire un appel de fonction veut dire qu'on l'utilise concrètement en lui fournissant des valeurs d'entrée précises. La fonction va alors effectuer son travail et fournir sa réponse avec **return**.

Notez qu'une fonction fournit sa réponse sur la ligne d'où elle a été appelée dès qu'elle exécute **return**.

```
1 def fois2(a):
2     b = a * 2
3     return b
4
5 x = fois2(10)
6 print(x)
7
8 y = fois2(100)
9 print(y)
```

Déroulé

L01(définition de la fonction)

L05(appel à la fonction en envoyant 10) -

L01(réception du 10) - L02 - L03 (envoi de 20)

L05 (retour de la réponse 20) - L06

L08 (appel à la fonction en envoyant 100) -

L01(réception du 100) - L02 - L03 (envoi de 200)

L08 (retour de la réponse 200) - L09

Comprendre où vont les entrées

Il faut comparer la ligne de la définition et celle de l'appel.

Avec l'appel de la ligne 5, on a 10 qui est stocké dans a lors de cet appel :

Définition	def fois2(a):
Appel	x = fois2(10)

5 - Fonction à plusieurs paramètres

On peut créer des fonctions possédant 0, 1, 2 ou plus paramètres. Le tout est de le déclarer.

```
1 def mystere(a, b, c):
2     reponse = a + b*2 + c
3     return reponse
4
5 x = mystere(10, 20, 30)
```

Lors de l'appel en ligne 5, on va donc envoyer 10 dans a, 20 dans b et 30 dans c.

Remarque : même sans rien envoyer, il faut placer les parenthèses. Exemple : **input()** ou **int(input())**