

1 Fonctions

1.1 Description

En informatique, une fonction est un « sous-programme ». Il existe des fonctions déjà définies dans Python, par exemple dans le module `math`, on trouvera les fonctions `sqrt`, `sin`, etc...

On peut aussi définir ses propres fonctions. Pour définir une fonction, on utilise le mot clé `def` suivi du nom de la fonction.

```
def nom():  
    Instructions
```

■ Exemple 1:

```
def hello():  
    print("Bonjour Toulmonde")
```

Remarque 1.

Les parenthèses sont obligatoires. Les deux points à la fin également. Les instructions sont indentées.

Si on exécute le programme précédent, rien ne s'affiche dans la console. La fonction `hello` a seulement été ajoutée à la collection déjà connue par Python.

1.2 Appeler une fonction

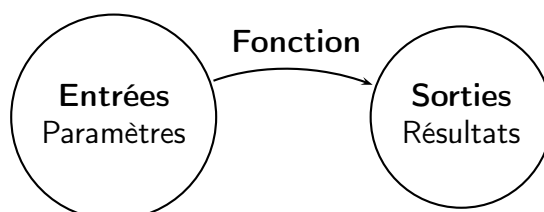
Pour qu'une fonction s'exécute, il suffit de l'appeler.

■ Exemple 2:

```
def hello():  
    print("Bonjour Toulmonde")  
  
hello()
```

2 Entrées/Sorties

Une fonction peut être schématisée comme ceci :



2.1 Les arguments (ou paramètres)

Une fonction admet zéro, un ou plusieurs arguments, séparés par des virgules.

■ Exemple 3:

```
def hello(nom, prenom):
    print("Bonjour", prenom, nom, "!")

hello("Zetofré", "Mélanie")
hello("Onskai", "Sandra")
```

2.2 Les résultats

Une fonction permet de « renvoyer » ou « retourner » la valeur d'une variable. Comme nous avons vu dans le premier TP, il existe plusieurs type de variables. Ainsi, une fonction peut retourner un entier, un flottant, un booléen, une chaîne de caractère ou une liste, selon les besoins.

On utilise le mot clé `return` suivi d'un espace, pas de parenthèses

■ Exemple 4:

```
def f(x):
    return 4*x**2-5*x+1

print(f(-3))
print(f(1))
```

```
def v(distance, temps):
    return distance/temps

print(v(22,4))
```

► Exercice 1

On calcule l'IMC (Indice de Masse Corporelle) d'une personne adulte par la formule suivante : $I = \frac{m}{t^2}$, où la masse m est donnée en kilogrammes et la taille t est donnée en mètres. Construire une fonction IMC qui prendra comme arguments la masse et la taille et renverra la valeur de l'IMC.

► Exercice 2

On définit la *factorielle* d'un nombre entier par le produit de tous ses prédécesseurs : $n! = n(n-1)(n-2)\dots 3 \times 2 \times 1$.

1. Créer une fonction `fact` qui prend en argument n et renvoie $n!$.
2. Faire afficher $0!$, $1!$, $2!$, ..., $10!$. (Utiliser une boucle `for`).

► Exercice 3

On définit le coefficient binomial $\binom{n}{k}$ par la formule $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ qui permet de connaître le nombre de façons de choisir k éléments parmi n .

1. Créer une fonction `binom` qui prend en argument n et k et qui renvoie $\binom{n}{k}$.

On pourra utiliser dans cette deuxième fonction la fonction `fact` créée précédemment !

2. Afficher les coefficients $\binom{5}{0}, \binom{5}{1}, \binom{5}{2}, \dots, \binom{5}{5}$

► Exercice 4

Quelle est le rôle de la fonction du programme suivant ?

```
1 def test(n):
2     a=True
3     for i in range(2,n):
4         if n%i==0:
5             a=False
6     return a
7
```

► Exercice 5

Soit la suite (u_n) définie par $u_0 = 1$ et $u_{n+1} = 0,8 * u_n + 5$.

Construire une fonction `seuil` qui prend en argument une valeur réelle et qui déterminera le premier rang de la suite qui dépassera cette valeur.

Donner les valeurs de `seuil(20)`, `seuil(24)`, `seuil(24.9999)`.

► Exercice 6

En utilisant la fonction `IMC` définie dans l'exercice 1, créer une fonction qui renvoie un message suivant le tableau suivant :

IMC	Interprétation
+ de 40	obésité morbide ou massive
35 à 40	obésité sévère
30 à 35	obésité modérée
25 à 30	surpoids
18,5 à 25	corpulence normale
16,5 à 18,5	maigreur
- de 16,5	famine

► Exercice 7

On définit une suite géométrique de raison $q = 1,04$ et de premier terme 1. Rédiger un algorithme permettant de connaître le rang à partir duquel la valeur du terme dépasse le double du premier terme. Faire varier la valeur du premier terme. Que constate-t-on ?

► Exercice 8

Interpréter le programme suivant. La fonction `random()` permet de générer un nombre aléatoire entre 0 et 1. Donc la commande `floor(6*random() + 1)` génère un nombre entier aléatoire entre 1 et 6.

```
from random import random
de=1
q=0
while de != 6:
    de=floor(6*random() + 1)
    q=q+1
print(q)
```

► Exercice 9

On définit la suite de Syracuse par le procédé suivant. Étant donnée une valeur initiale u_0 fixée, pour

$$\text{tout } n \in \mathbb{N}, u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } n \text{ est pair} \\ 3u_n + 1 & \text{sinon} \end{cases}$$

Écrire un programme qui permet de déterminer le premier rang à partir duquel la suite atteint la valeur 1, en prenant des valeurs initiales entre 2 et 20.