

I

Fonction à un ou plusieurs arguments

Définition I.1

Une **fonction** est un bloc d'instructions à qui on a donné un nom. Une fonction reçoit des arguments en entrée et renvoie un résultat.

Le bloc d'instructions d'une fonction n'est exécuté que si la fonction est appelée.

Commande pour définir une fonction en Python :

```
Python
def fonction(argument1, argument2, ...):
    Instructions
    return resultat
```

Pour exécuter une fonction, on utilise l'instruction suivante :

```
Python
fonction(argument1, argument2, ...)
```

Exemple I.1 — On donne la fonction suivante :

```
Python
def mafonction(x, y, z):
    s = x + y + z
    return s
```

- Que va renvoyer cette fonction si on prend pour arguments $x = 3$, $y = 2$ et $z = -1$?
- Que vaut `mafonction(0,0,10)` ?

→ À rédiger

Exemple I.2 — Écrire une fonction qui renvoie l'aire d'un triangle. Cette fonction aura deux arguments que l'on appellera `base` et `hauteur`.

→ À rédiger

II

Utilisation des fonctions en statistiques et probabilités

1. Fonction renvoyant un nombre aléatoire

Pour générer un nombre aléatoire, on utilise la bibliothèque `random`. L'instruction `random.randint(a,b)` donne un nombre entier aléatoire entre `a` et `b` et l'instruction `random()` donne un nombre flottant aléatoire entre 0 et 1.

Exemple II.1 — Écrire une fonction `simuler` afin simuler le lancer d'un dé cubique, qui revoie `True` lorsque le nombre obtenu est un multiple de 3 et qui renvoie `False` sinon.

→ À rédiger

Exemple II.2 — Écrire une fonction qui simule le lancer d'une pièce biaisée qui tombe sur Pile dans 40% des cas. Cette fonction renverra `True` si la pièce tombe sur Pile et `False` sinon.

→ À rédiger

2. Calcul de moyenne et d'écart-type

Exemple II.3 — On donne le programme suivant :

```
Python
import math

def f(a,b,c,d,e):
    m = (a+b+c+d+e)/5
    return m
```

```
Python
def g(a,b,c,d,e):
    m = f(a,b,c,d,e)
    v = (a-m)**2 + (b-m)**2 + (c-m)**2 + (d-m)**2 + (e-m)**2)/5
    e = math.sqrt(v)
    return e
```

Expliquer le fonctionnement des fonctions `f` et `g`. Quel est leur rôle ?

→ À rédiger

Solutions

Exemple I.1

1. La fonction va renvoyer la valeur 4.
2. 10.

Exemple I.2

Voici la fonction demandée :

Python

```
def aire(hauteur, base):  
    aire = hauteur * base / 2  
    return aire
```

Exemple II.1

Voici une fonction possible :

Python

```
import random  
  
def simuler():  
    resultat = random.randint(1,6)  
    if resultat%3==0:  
        return True  
    else:  
        return False
```

Autre possibilité :

Python

```
import random  
  
def simuler():  
    resultat = random.randint(1,6)  
    return resultat%3==0
```

Exemple II.2

Voici une fonction possible :

Python

```
import random  
  
def simuler():  
    resultat = random.random()  
    if resultat<=0.40:  
        return True  
    else:  
        return False
```

Autre possibilité :

Python

```
import random  
  
def simuler():  
    resultat = random.random()  
    return resultat<=0.40
```

Exemple II.3

La fonction `f` prend cinq arguments en entrée et renvoie leur moyenne.

La fonction `g` prend cinq arguments en entrée et commence par calculer leur moyenne et stocke le résultat dans la variable `m`. Ensuite, elle calcule l'écart-type `e` puis elle renvoie ce résultat.

Algorithmique (3ème partie)

A savoir faire à la fin du chapitre.

- Savoir ce qu'est une fonction en Python.
- Savoir écrire des fonctions simples.
- Savoir comment appeler une fonction.
- Savoir lire, comprendre, modifier, compléter des fonctions plus complexes.
- Savoir écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire ou d'une répétition d'expériences aléatoires indépendantes.
- Savoir lire et comprendre une fonction renvoyant une moyenne.

Algorithmique (3ème partie)

A savoir faire à la fin du chapitre.

- Savoir ce qu'est une fonction en Python.
- Savoir écrire des fonctions simples.
- Savoir comment appeler une fonction.
- Savoir lire, comprendre, modifier, compléter des fonctions plus complexes.
- Savoir écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire ou d'une répétition d'expériences aléatoires indépendantes.
- Savoir lire et comprendre une fonction renvoyant une moyenne.

Algorithmique (3ème partie)

A savoir faire à la fin du chapitre.

- Savoir ce qu'est une fonction en Python.
- Savoir écrire des fonctions simples.
- Savoir comment appeler une fonction.
- Savoir lire, comprendre, modifier, compléter des fonctions plus complexes.
- Savoir écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire ou d'une répétition d'expériences aléatoires indépendantes.
- Savoir lire et comprendre une fonction renvoyant une moyenne.