

# Python : listes et statistiques



Répondre aux questions dans son cahier !



Lorsqu'il est demandé de compléter un morceau de code, il est souvent pratique d'utiliser le panneau de *gauche* de Spyder. Tout ce qui y est écrit peut alors être lancé dans l'interpréteur (fenêtre de droite) en appuyant sur la touche F5.

## 1 Listes

Nous allons utiliser des listes en Python.

En Python, une liste est définie en ouvrant des crochets [], et en plaçant des éléments à l'intérieur, séparés par des virgules.

```
maliste = [6, 9, 10, 5 + 2]
```

on peut accéder aux éléments d'une liste en indexant :

```
print(maliste[0]) # Affiche le premier élément de la liste
```

Rentrer le code ci-dessus dans l'interpréteur.

1. Afficher le 4<sup>ème</sup> élément.

2. Que se passe-t-il si on essaie d'afficher le 5<sup>ème</sup> élément de cette liste ?

Il est possible d'obtenir la longueur d'une liste en utilisant la fonction `len`. On peut également obtenir tous les éléments de la liste l'un après l'autre, en utilisant une boucle `for ... in liste`.

```
autreliste = [2, 5, 7, 11]
for x in autreliste:
    print(x)
print("la longueur de autreliste est ", len(autreliste))
```

3. Qu'affiche la fonction ci-dessus ?

Enfin, il est possible de générer des listes d'une autre manière, dite 'impérative'

```
liste1 = [i for i in range(50)]
liste2 = [2 * i for i in range(50)]
```

4. Que contient la liste `liste1` ?

5. Que contient la liste `liste2` ?

6. Compléter le code ci-dessous pour obtenir la somme de tous les éléments de la liste `liste1` :

```
total = 0
for x in ...:
    ...
print(total)
```

## 2 Lancers de dés



Cette activité est une reproduction du TP 3 page 297 du manuel.

1. (a) Reproduire la fonction ci-dessous puis la compléter pour qu'elle renvoie l'effectif d'une valeur d'une liste donnée.

```
from random import randint
import matplotlib.pyplot as plt

def nombre_de(valeur, liste):
    nb=0
    for x in liste:
        if x==...:
            ...
    return nb
```

- (b) Que renvoie `nombre_de(1, [2,1,4,3,1,3,6,1,7,12])` ?
2. On considère l'expérience aléatoire suivante : on lance deux dés équilibrés à six faces et on fait la somme des deux faces obtenues.

Le programme suivant écrit à la suite de la fonction `nombre_de`, permet de représenter par un nuage de points le nombre d'obtentions de chaque issue lorsqu'on répète 1 000 fois l'expérience précédente.

```
L = [randint(1,6)+randint(1,6) for i in range(1000)]
for k in range(...):
    plt.plot(k,nombre_de(k,L), '*')
plt.show()
```

- (a) Compléter ce programme et commenter le graphique obtenu.
- (b) Modifier ce programme de façon à répéter 10 000 fois l'expérience aléatoire.
- (c) Modifier ce programme pour qu'il calcule la moyenne des sommes obtenues après 10 000 lancers des deux dés.