

TP8

CRÉER UNE LISTE AVEC UNE BOUCLE

*Corrigé***Exercice 1 Manipulation de listes.**

Tout cet exercice peut être fait dans la console.

1. Définir les listes $L_1 = [1, 2, 3, \dots, 10]$ et $L_2 = [4, 7, 12, 11, 8]$.

```
L1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
# ou
L1 = [i for i in range(1, 11)]

L2 = [4, 7, 12, 11, 8]
```

2. Dans la console, vérifier que L_1 contient 10 éléments et que L_2 contient 5 éléments.

```
>>> len(L1)
10

>>> len(L2)
5
```

3. Modifier le deuxième élément de L_2 (le 7 donc) pour qu'il soit égal à 6. On pensera à vérifier en affichant L_2 .

```
>>> L2[1]=6

>>> L2
[4, 6, 12, 11, 8]
```

4. Ajouter un 11 à la fin de L_1 .

```
>>> L1.append(11)

>>> L1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

5. Supprimer ce 11.

```
>>> L1.pop()
11

>>> L1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# OU
>>> L1.pop(len(L1)-1)
# L1.pop(i) renvoie terme de position i et l'enlève de la liste
11

>>> L1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

6. Créer $L_3 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 4, 6, 12, 11, 8]$ à partir de L_1 et L_2 .

```
>>> L1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

>>> L2
[4, 6, 12, 11, 8]

>>> L3=L1+L2

>>> L3
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 4, 6, 12, 11, 8]
```

7. Créer $L_4 = [1, 2, 3, 4, 5]$ à partir de L_1 .

```
>>> L1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> L4 = L1[ 0 : 5 ] # éléments numérotés 0 à 4 -- ou L1[ : 5 ]
>>> L4
[1, 2, 3, 4, 5]
```

8. On a $L_2 = [4, 6, 12, 11, 8]$.

Ajouter un 10 en troisième position pour obtenir $L_2 = [4, 6, 10, 12, 11, 8]$, sans recopier les autres valeurs. *On découpera la nouvelle liste en 3 morceaux, que l'on concaténera avec +.*

```
>>> L2
[4, 6, 12, 11, 8]

>>> L2[0:3]
[4, 6, 12]

>>> L2[3:]
[11, 8]

>>> L2 = L2[0:3] + [10] + L2[3:]

>>> L2
[4, 6, 12, 10, 11, 8]
```

9. Créer la liste $[0, 0, \dots, 0]$ avec 15 zéros en utilisant la répétition `*`.

```
>>> [0]*15
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Exercice 2 Définir une liste par compréhension

On peut définir une liste de valeurs suivant un schéma répétitif par

```
liste = [ element_k for k in range(...) ]
```

Par exemple, $L = [1, 4, 9, 16, 25]$ peut être obtenue avec

```
L = [ k**2 for k in range(1,6) ]
```

1. Définir de cette façon les listes suivantes :

$L1 = [1, 2, 3, 4, \dots, 100]$ $L2 = [\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{20}]$ $L3 = [3, 6, 9, 12, \dots, 99]$

```
>>> L1=[i for i in range(1,101)]
>>> L2=[1/i for i in range(2,21)]
>>> L3=[3*i for i in range(1,34)]
```

2. Importer la bibliothèque `random` puis créer une liste `L4` de 15 entiers aléatoires compris entre 1 et 6.

```
from random import randint
L4 = [randint(1,6) for i in range(15)]

>>> L4
[3, 1, 4, 1, 5, 5, 3, 2, 3, 2, 1, 5, 3, 4, 6]
```

3. Dans `L4`, *échanger* l'élément en position 0 et l'élément en position 3. Votre programme doit fonctionner quelle que soit la liste aléatoire `L4`.

```
a = L4[3]
L4[3] = L4[0]
L4[0] = a

>>> L4
[1, 1, 4, 3, 5, 5, 3, 2, 3, 2, 1, 5, 3, 4, 6]
```

Exercice 3 Créer une liste par ajouts successifs d'éléments

```
from math import *
u = 3
L = [u] # il faut mettre u0 dans la liste
for i in range(20):
    u = log( i*u+2)
    L.append(u) # on ajoute l'élément u à la fin de L
print(L)

>>> L[20]
4.444068448733263
```

```
# version 1
def liste_v(n):
    v = 4
    L=[v]
    for k in range(1,n):
        v = v + 1/k
        L.append(v)
    return L

# version 2
def liste_v(n):
    L=[4]
    for k in range(1,n):
        v = L[-1] + 1/k
        L.append(v)
    return L
```

Exercice 4

1. Demander à l'utilisateur de saisir un entier $n \geq 1$ puis créer une liste aléatoire L d'entiers entre 1 et 50 de longueur n comme à l'exercice 2.

```
n = int(input("donner un entier : "))
L = [randint(1,50) for i in range(n)]
print(L)

>>> (executing lines 15 to 16 of "TP6listes.py")
donner un entier : 5

>>> L
[39, 46, 6, 5, 3]
```

2. Écrire des instructions conditionnelles qui répondent au problème qui suit :
 - si la longueur de la liste est paire alors afficher le premier et le dernier élément ;
 - si la longueur est impaire alors afficher l'élément situé au milieu de la liste.

```
if len(L)%2 == 0:
    print("Le premier élément est ",L[0])
    print("Le dernier élément est ", L[len(L)-1] ) # ou L[-1]
else:
    print("L'élément du milieu est ", L[len(L)//2]) # quotient
```

Exercice 5 Soit (v_n) la suite définie par $v_0 = 2$, $v_1 = 5$ et

$$\forall n \in \mathbb{N}, v_{n+2} = v_n - v_{n+1}.$$

Écrire une fonction `liste_v(n)` qui renvoie liste $[v_0, v_1, \dots, v_n]$ pour $n \in \mathbb{N}$.

```
def liste_v(n):
    a = 2 # v0
    b = 5 # v1
    L = [a,b] # v0 et v1 sont dedans
```

```
    for i in range(2,n+1): # le prochain terme à mettre est u2
                          # for i in range(n-1) fonctionne aussi
        c = a-b
        a = b
        b = c
        L.append(c)
    return L

>>> liste_v(2)
[2, 5, -3]
```