

## P5 - Les listes

Corrigé

### Exercice 1 :

1. Créer une liste L1 contenant les éléments 4, 7, 12, 11 et 8 dans cet ordre.

In [1]: `L1 = [4,7,12,11,8]`

2. Afficher le nombre d'éléments dans la liste L1.

In [2]: `len(L1)`

Out[2]: 5

3. Afficher le premier élément de L1

In [3]: `L1[0]`

Out[3]: 4

4. Afficher le dernier élément de L1, de deux façons différentes.

In [4]: `L1[-1]`

Out[4]: 8

In [5]: `L1[4]`

Out[5]: 8

5. Afficher le troisième élément de L1 (le 12) de deux façons différentes.

In [6]: `L1[2]`

Out[6]: 12

In [7]: `L1[-3]`

Out[7]: 12

6. Afficher la liste [7,12] à partir de L1.

In [8]: `L1[1:3]`

Out[8]: [7, 12]

7. Ajouter un 13 à la fin de L1.

In [9]: `L1.append(13)`  
`L1`

Out[9]: [4, 7, 12, 11, 8, 13]

\*Attention, si on exécute deux fois cette cellule, cela ajoute deux fois le 13.

8. Vérifier, avec un booléen, que 13 est dans la liste.

```
In [10]: 13 in L1
```

```
Out[10]: True
```

9. Supprimer le 13.

```
In [11]: del L1[-1]
L1
```

```
Out[11]: [4, 7, 12, 11, 8]
```

\*Attention, si on exécute deux fois cette cellule, cela supprime deux fois l'élément numéro -1 (le dernier)

10. Vérifier, avec un booléen, que 13 n'est plus dans la liste.

```
In [12]: 13 in L1
```

```
Out[12]: False
```

11. Modifier le deuxième élément (le 7) pour qu'il soit égal à -1.

```
In [13]: L1[1] = 7
L1
```

```
Out[13]: [4, 7, 12, 11, 8]
```

## Exercice 2

1. Créer la liste L2 = [3,6,9,12,...,60] en une instruction.

```
In [14]: L2 = [3*k for k in range(1,21)]
L2
```

```
Out[14]: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60]
```

2. Définir maintenant la liste L2 en partant d'une liste vide [] et en utilisant des append successifs.

```
In [15]: L2 = []
for k in range(1,21):
    L2.append(3*k)
L2
```

```
Out[15]: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60]
```

3. Enfin, définir L2 en partant d'une liste nulle puis modifier successivement les éléments.

```
In [16]: L2 = [0]*20 #la taille de L2 voulue est 20
for i in range(len(L2)):
    L2[i] = 3*(i+1)
L2
```

```
Out[16]: [3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48, 51, 54, 57, 60]
```

**Exercice 3**

On considère la suite  $(u_n)_{n \in \mathbb{N}}$  définie par

$$\begin{cases} u_0 = 3 \\ \forall n \in \mathbb{N}, u_{n+1} = \ln(nu_n + 2) \end{cases}$$

Écrire un programme qui, étant donné  $n \in \mathbb{N}$ , crée la liste

$$L3 = [u_0, u_1, u_2, \dots, u_n]$$

On donnera deux façons de procéder. Vérifier que  $u_{20} \approx 4,444$ .

```
In [17]: import numpy as np

# Méthode 1 : avec append
n = 20
u = 3
L3 = [u]
for k in range(n):
    u = np.log(k*u+2)
    L3.append(u)
print(L3)

# OU, sans la variable u
L3 = [3]
for k in range(n):
    L3.append(np.log(k*L3[k]+2)) # ou L3.append(np.log(k*L3[-1]+2))
print(L3)
```

```
Out[17]: [3,
0.6931471805599453,
0.9907104653475315,
1.3816387733432482,
1.8156251253548543,
2.225974044756554,
2.5748898043193824,
2.859301758298302,
3.0917291406295497,
3.2859299214798408,
3.4523140214022803,
3.597946038175374,
3.727556904821344,
3.8444009694422068,
3.950805403357433,
4.048497387031752,
4.138799323136612,
4.222749344941559,
4.301179025270597,
4.3747656178645595,
4.444068448733263]
```

```
In [18]: # Méthode 2, en partant d'une liste nulle
# L3[i] = u_i
n = 20
L3 = [0]*(n+1) # la taille de L3 est n+1
L3[0] = 3
for i in range(n):
    L3[i+1] = np.log(i*L3[i]+2) #ici, on voit bien apparaître la relation de récurrence
print(L3)

# OU, avec une variable u
L3 = [0]*(n+1) # la taille de L3 est n+1
u = 3
L3[0] = u
for k in range(n):
    u = np.log(k*u+2)
    L3[k+1] = u #u_(k+1)
print(L3)
```

```
Out[18]: [3,
0.6931471805599453,
0.9907104653475315,
1.3816387733432482,
1.8156251253548543,
2.225974044756554,
2.5748898043193824,
2.859301758298302,
3.0917291406295497,
3.2859299214798408,
3.4523140214022803,
3.597946038175374,
3.727556904821344,
3.8444009694422068,
3.950805403357433,
4.048497387031752,
4.138799323136612,
4.222749344941559,
4.301179025270597,
4.3747656178645595,
4.444068448733263]
```

#### Exercice 4

On considère la suite  $(u_n)_{n \in \mathbb{N}}$  définie par

$$\begin{cases} v_0 = 1, v_1 = 1 \\ \forall n \in \mathbb{N}, v_{n+2} = v_{n+1} + v_n \end{cases}$$

Écrire un programme qui, étant donné  $n \in \mathbb{N}$ , crée la liste  $L4 = [v_0, v_1, v_2, \dots, v_n]$ .

```
In [19]: n = 20
L4 = [0]*(n+1)
L4[0] = 1
L4[1] = 1
for i in range(n-1):
    L4[i+2] = L4[i+1]+L4[i]
print(L4)
```

```
Out[19]: [1,
1,
2,
3,
5,
8,
13,
21,
34,
55,
89,
144,
233,
377,
610,
987,
1597,
2584,
4181,
6765,
10946]
```

### Exercice 5 - Sommes et produits (pas de listes ici)

1. Écrire un programme calculant  $\sum_{k=1}^{1000} \frac{1}{k^2}$ .

```
In [20]: S = 0
         for k in range(1,1001):
           S = S + 1/k**2
         S
```

Out[20]: 1.6439345666815615

2. Écrire un programme calculant 50!.

```
In [21]: p = 1
         for k in range(1,51):
           p = p*k
         p
```

Out[21]: 3041409320171337804361260816606476884437764156896051200000000000