

TP6 - Les fonctions - Corrigé

Exercice 1

Définir une fonction **EurosToLivres(PE)** qui, étant donné un nombre PE correspondant à un prix en euros, renvoie le prix associé PL en livres Sterling. 1 EUR = 0.86 GBP (cours relevé le 28/10/2022).

```
In [1]: def EurosToLivres(PE):  
        PL = 0.86*PE  
        return PL
```

Testez votre fonction.

```
In [3]: EurosToLivres(100)
```

```
Out[3]: 86.0
```

Écrire la fonction de conversion inverse, puis tester.

```
In [4]: def LivresToEuros(PL):  
        PE = PL/0.86  
        return PE
```

```
In [5]: LivresToEuros(100)
```

```
Out[5]: 116.27906976744187
```

Exercice 2

1. Écrire une fonction **somme(liste)** qui, étant donnée une liste de nombres, renvoie la somme des éléments de la liste.

```
In [8]: def somme(liste):  
        S = 0  
        for x in liste:  
            S = S + x  
        return S
```

On n'oublie pas de tester la fonction.

```
In [9]: somme([1,2,3,4])
```

```
Out[9]: 10
```

2. En déduire une fonction **moyenne(liste)** qui, étant donnée une liste de nombres, renvoie la moyenne des éléments de la liste.

```
In [10]: def moyenne(liste):  
         return somme(liste)/len(liste)
```

Et on teste, bien sûr.

```
In [11]: moyenne([1,2,3,4])
```

```
Out[11]: 2.5
```

Exercice 3

Écrire une fonction factorielle(n) qui renvoie n! pour $n \in \mathbb{N}$ puis calculer la somme

$$S = \sum_{k=0}^{20} \frac{2^k}{k!}.$$

On vérifiera que $S \approx e^2$.

```
In [12]: def factorielle(n):  
        p = 1  
        for k in range(1,n+1):  
            p = p*k  
        return p
```

```
In [13]: factorielle(5)
```

```
Out[13]: 120
```

```
In [14]: S = 0  
        for k in range(0,21):  
            S = S + 2**k/factorielle(k)  
        print(S)
```

```
7.389056098930604
```

```
In [15]: import numpy as np  
        np.exp(2)
```

```
Out[15]: 7.38905609893065
```

Exercice 4

1. Définir une fonction **suiteU(n)** qui, étant donné $n \in \mathbb{N}$ renvoie la valeur de u_n où

$$u_0 = 1 \text{ et } u_{n+1} = \sqrt{u_n + 2n}.$$

On vérifiera que $u_{10} \approx 4,7432\dots$

```
In [16]: import numpy as np  
        def suiteU(n):  
            u = 1  
            for k in range(n):  
                u = np.sqrt(u+2*k)  
            return u
```

```
In [17]: suiteU(10)
```

```
Out[17]: 4.743269000795927
```

2. Écrire un programme qui utilise cette fonction pour déterminer le plus petit entier n tel que $u_n \geq 60$.

```
In [23]: n = 0  
        while suiteU(n) < 60:  
            n = n+1  
        print(n)
```

```
1772
```

3. Même question sans utiliser la fonction suiteU. Appréciez la différence de temps d'exécution.

```
In [24]: n = 0
u = 1
while u < 60:
    u = np.sqrt(u+2*n)
    n = n+1
print(n)
```

1772

4. Utiliser la fonction `suiteU` pour créer la liste $[u_0, u_1, \dots, u_{20}]$.

```
In [25]: [suiteU(n) for n in range(21)]
```

```
Out[25]: [1,
1.0,
1.7320508075688772,
2.3941701709713277,
2.8972694336169926,
3.301101245587144,
3.647067485746205,
3.9556374310275464,
4.237409282926013,
4.498600813911589,
4.743269000795927,
4.974260648658847,
5.193675061905476,
5.403117161593433,
5.603848424216472,
5.796882647097186,
5.9830496109506885,
6.163038991516335,
6.337431576870581,
6.506722030090926,
6.671335850494332]
```

Exercice 5

Écrire une fonction qui détermine le plus grand élément d'une liste.

```
In [28]: def max(L):
m = L[0]
for x in L:
    if x > m:
        m = x
return m
```

```
In [29]: max([1,2,47,8,5,1,5,87,4,5])
```

```
Out[29]: 87
```

```
In [ ]:
```