

09. Fonctions

Nous utiliserons dans ce TP l'environnement de développement Spyder.

Exemple 1. Fonction factorielle : algorithme et programme en Python

Fonction: $factorielle(n)$
Action: Calcul de la factorielle f d'un entier n
Début
 $f \leftarrow 1$
 Pour k allant de 1 à n faire
 | $f \leftarrow kf$
 FinPour
 Renvoyer f
Fin

```
def factorielle(n):
    """Calcul de la factorielle f d'un entier n"""
    f=1
    for k in range(1,n+1):
        f=k*f
    return f
```

Exercice 1. Tester le programme précédent en calculant $factorielle(6)$ dans l'interpréteur Python.

Exercice 2.

- Écrire sous forme d'algorithme une fonction permettant de calculer le nombre d'années nécessaire pour doubler un capital placé à intérêts composés avec un taux annuel de $t\%$.
- Traduire cet algorithme en programme Python puis tester ce dernier.

Exercice 3.

- Écrire sous forme d'algorithme une fonction qui calcule le produit des entiers de m à n .
- Traduire cet algorithme en programme Python puis tester ce dernier.

Exercice 4.

- Écrire sous forme d'algorithme une fonction permettant de calculer le n -ième terme de la suite de Fibonacci définie par
$$\left\{ \begin{array}{l} u_1 = 1 \\ u_2 = 1 \\ u_{n+2} = u_{n+1} + u_n, n \in \mathbb{N}^* \end{array} \right. .$$
- Traduire cet algorithme en programme Python puis tester ce dernier.

Exercice 5.

- Écrire sous forme d'algorithme une fonction qui calcule la somme des carrés inférieurs ou égaux à un entier n .
- Traduire cet algorithme en programme Python puis tester ce dernier.

Exercice 6.

- Écrire sous forme d'algorithme une fonction permettant de déterminer si un entier $n \geq 2$ est un nombre premier.
- Traduire cet algorithme en programme Python puis tester ce dernier.

Exercice 7.

- Écrire sous forme d'algorithme une fonction permettant de déterminer le nombre de diviseurs d'un entier n non nul.
- Traduire cet algorithme en programme Python puis tester ce dernier.