

Langage de base : Atelier n°5 Fonctions

Exercice 1 (clé) :

- 1) Créer une fonction qui prend en argument 2 nombres et retourne leur somme multiplié par 2.
- 2) Créer une fonction qui retourne la moyenne de 2 nombres.

Exercice 2 (clé) :

- 1) Créer une fonction qui prend en entrée 2 nombres et qui retourne le plus grand de ces 2 nombres.
- 2) Créer une fonction qui prend en entrée 2 nombres et qui retourne le plus petit de ces 2 nombres.
- 3) Créer un programme qui demande à l'utilisateur de saisir un premier nombre puis un deuxième nombre et qui retourne le plus grand et le plus petit de ces nombres
- 4) Modifier le programme pour que celui-ci prenne en compte le cas où les nombres sont égaux.

Exercice 3 (clé) :

- 1) Créer une fonction permettant de calculer l'aire d'un triangle
- 2) Créer une fonction retournant la valeur de l'hypoténuse d'un triangle rectangle en prenant en entrée les 2 plus petits côtés du triangle (théorème de pythagore).

Exercice 4 (clé) :

- 1) Créer une fonction qui retourne la plus grande valeur d'une liste d'entiers
Exemple : `[1, 8, 4, 2] => 8`
- 2) Créer une fonction qui retourne la somme des valeurs d'une liste d'entiers
Exemple : `[1, 8, 4, 2] => 15`
- 3) Créer une fonction qui retourne la valeur moyenne d'une liste d'entiers (pour aller plus vite vous avez le droit d'utiliser la fonction de la question 2)
Exemple : `[1, 8, 4, 2] => 3.75`

Exercice 5 (secondaire) :

- 1) Créer une fonction qui prend en entrée un nombre et qui retourne un booléen True si ce nombre est pair (indice : pour vérifier si un nombre est pair on utilise le modulo : **`nbr % 2 == 0`**)
- 2) Créer une fonction qui prend en entrée un nombre et qui retourne un booléen True si ce nombre est impair (indice : pour vérifier si un nombre est pair on utilise le modulo : **`nbr % 2 == 1`**)
- 3) Vérifier le bon fonctionnement de ces fonctions

Exercice 6 (secondaire) :

- 1) Créer une fonction qui compte le nombre de caractères d'un mot.
- 2) Créer une fonction qui compte le nombre de mots d'une ligne.

Exercice 7 (secondaire) :

- 1) Créer un programme qui prend en entrée un nombre et qui retourne une liste de nombre entier aléatoire dont l'addition vaut le nombre en entrée.

Exercice 8 (secondaire) :

Un programme principal saisit une chaîne d'ADN valide et une séquence d'ADN valide (valide signifie qu'elles ne sont pas vides et sont formées exclusivement d'une combinaison arbitraire de "a", "t", "g" ou "c").

Écrire une fonction **valide** qui renvoie vrai si la saisie est valide, faux sinon.

Écrire une fonction **saisie** qui effectue une saisie valide et renvoie la valeur saisie sous forme d'une chaîne de caractères.

Écrire une fonction **proportion** qui reçoit deux arguments, la chaîne et la séquence et qui retourne la proportion de séquence dans la chaîne (c'est-à-dire son nombre d'occurrences).

Le programme principal appelle la fonction **saisie** pour la chaîne et pour la séquence et affiche le résultat.

Exemple d'affichage :

```
chaîne : attgcaatggtgtacatg
```

```
séquence : ca
```

```
Il y a 10.53 % de "ca" dans votre chaîne.
```