



# les exercices de python

## Exercice 1 :

Ecrire une fonction permettant de calculer et retourner le factoriel d'un nombre entier positif, donné en paramètre.

(NB :  $0! = 1$ )

## Exercice 2 :

Ecrire une procédure permettant de définir si une chaîne de caractère donnée en paramètre est palindrome ou non.

Une chaîne palindrome est une chaîne qui se lit de la même façon dans les deux sens. (ex : OPPO)

## Exercice 3 :

Écrivez une fonction `validate_email` qui vérifie si une chaîne de caractères donnée en paramètre est une adresse email valide. La fonction doit vérifier que l'adresse contient un symbole "@" suivi par un point "." et qu'il y a au moins un caractère avant le "@" et après le point.

La fonction retourne `true` si l'email est valide ou `false` le cas échéant.

## Exercice 4 :

Ecrire une fonction `dessin_triangle()` qui prend en argument un entier `n` strictement positif. La fonction doit afficher un triangle isocèle de taille `n`.

Exemple:

```
dessin_triangle(3) -->
  *
 * *
* * *
```

## Exercice 5 :

Ecrire une fonction `calcul_occurences()` ayant deux paramètres : un caractère et une chaîne de caractères.

La fonction doit chercher et retourner le nombre des occurrences du caractère à l'intérieur de la chaîne de caractères.

exemple:

```
calcul_occurences('A','CASABLANCA') ---> 4
```

**Exercice 6 :**

Ecrire une fonction `moyenne_liste()` en python ayant comme argument une liste L composée de nombres réels.

La fonction doit retourner la moyenne des nombres appartenant à la liste L

Exemple :

`L=[-2, 5.3, 6.7]`

`moyenne_liste(L) --> 3.33333`

**Exercice 7:**

Ecrire une procédure en python permettant deux arguments :

Un élément i et une liste L.

La procédure doit afficher :

Le nombre des occurrences de l'élément i dans la liste L.

Puis, si le nombre des occurrences est  $\geq 1$ , la procédure doit afficher également :

- La position de la première occurrence.

- La position de la dernière occurrence.

**Exercice 8 :**

Ecrire une procédure `trier_liste()` en python ayant comme argument une liste L composée de nombres réels.

La procédure doit afficher les éléments de la liste L triés dans le sens croissant.

Vous êtes invités à utiliser l'algorithme de TRI par SELECTION.

Exemple :

`L=[-7.2,3.4,-9.5,1.2]`

`trier_liste(L) --> -9.5 -7.2 1.2 3.4`

**Exercice 9:**

Ecrire une procédure nommée `supp_voyelles()` qui accepte une chaîne de caractères comme paramètre.

Cette procédure doit afficher la chaîne en éliminant les voyelles.

**Exercice 10 :**

Ecrire une fonction nommée `isPerfect()` qui admet un entier comme paramètre.

La fonction doit retourner `True` si le paramètre est parfait, sinon `False` dans le cas inverse.

Un nombre parfait est un nombre qui est égal à la somme de ses diviseurs propres.

Ex:

6 est parfait.

Les diviseurs propres de 6 sont : 1, 2 et 3.

La somme des diviseurs propres est :  $1+2+3 = 6$

**Exercice 11:**

Ecrire une fonction nommée `insert_String()` qui admet trois paramètres. Une chaîne de caractères, un entier et un caractère.

Cette fonction doit retourner une nouvelle chaîne incluant le caractère indiqué dans le 3ème paramètre à la position indiquée dans le deuxième paramètre.

Exemple :

`insert_String("Casablanca",2,'m') ----> "Camsablanca"`