

Partie V : input(), if, for et while.

À partir de maintenant, nous rédigerons nos programmes dans la fenêtre éditeur de texte afin de pouvoir les sauvegarder.

Boucle for et while (tant que)

For :

À vous !

Tester les différentes lignes de code suivantes afin de comprendre le fonctionnement de la boucle for :

```
>>> for i in range (1,4) :  
    print(i)
```

```
>>> for i in range (5) :  
    print(i)
```

```
>>> for i in range (1,10,2) :  
    print(i)
```

Vous pouvez ainsi paramétrer totalement for boucle for...

Exemple de programme à tester : Table de multiplication

```
>>> def table_de_multiplication_de(a):    #Nous définissons la fonction table_de_multiplication_de(a)  
    for i in range(11):  
        print(a*i)  
  
>>> table_de_multiplication_de(4)      #Nous la testons pour a=4.  
0                                       #Réponse.  
4  
8  
12  
16  
20  
24  
28  
32  
36  
40
```

À vous ! M'appeler pour validation à chaque fin de projet

- Écrivez un programme qui affiche les 20 premiers termes de la table de multiplication par 7.
- Créer un programme permettant d'afficher les 20 premiers termes de la suite de Fibonacci.

(La suite de Fibonacci est une suite de nombre ans laquelle chaque terme est la somme des deux termes qui le précèdent. Elle commence par les termes 0 et 1)

- Écrivez un programme avec la boucle for qui permet de calculer la somme des entiers de 1 à n. n sera le paramètre de votre fonction.
- Écrivez un programme permettant de calculer le produit des nombres impaires de 1 à 23.
- Balayage :
Recherche de façon expérimentale une valeur approchée du minimum de la fonction

$f(x) = x^3 - x^2 - \frac{1}{4}x + 1$ sur l'intervalle $[0, 1]$.

Indications.

- Construisez une boucle dans laquelle une variable i balaye les entiers de 0 à 100.
- Définissez $x = \frac{i}{100}$. Ainsi $x = 0.00$, puis $x = 0.01$, $x = 0.02$...
- Calculez $y = x^3 - x^2 - \frac{1}{4}x + 1$
- Affichez les valeurs à l'aide de `print("x =", x, "y =", y)`.
- Cherchez à la main pour quelle valeur de x on obtient un y le plus petit possible.
- N'hésitez pas à modifier ton programme pour augmenter la précision.
- Pour les plus fort : Trouver le plus petit entier $n > 1$ tel que la moyenne de $1^2, 2^2, 3^2, \dots, n^2$ soit un carré parfait. (d'après *British Mathematical Olympiad 1994*)

While :

À vous !

Tester les différentes lignes de code suivantes afin de comprendre le fonctionnement de la boucle while :

```
>>> a = 0           #initialisation de la variable
>>> while (a < 7): # (n'oubliez pas le double point !)
    a = a + 1      # (n'oubliez pas l'indentation qui permet de faire évoluer la variable a !)
    print(a)
```

Remarque : Il n'est pas obligatoire de faire évoluer la variable de « 1 en 1 », mais il faut la faire évoluer pour ne pas avoir une boucle infinie.

À vous !

- Écrivez un programme qui affiche une table de conversion de sommes d'argent exprimées en euros, en dollars canadiens. La progression des sommes de la table sera « géométrique », comme dans l'exemple ci-dessous :

1 euro(s) = 1.65 dollar(s)
2 euro(s) = 3.30 dollar(s)
4 euro(s) = 6.60 dollar(s)
8 euro(s) = 13.20 dollar(s)
etc. (S'arrêter à 16 384 euros)

M'appeler pour validation.

- Écrivez un programme qui affiche une suite de 12 nombres dont chaque terme soit égal au triple du terme précédent.

M'appeler pour validation.