

## COURS 4 : Programmer une boucle bornée – 1 heure

La **boucle bornée** ou **boucle for** ou **boucle pour** est communément utilisée lorsque l'on souhaite répéter plusieurs fois une même instruction. Il faut connaître le nombre de fois que vous souhaitez répéter l'instruction.

Vous aurez besoin d'une variable permettant de compter le nombre de fois que vous répétez l'instruction.

### Programme 20 :

```
1 ▾ for i in range(1,6) :  
2   | print ("je répète 6 fois un message")
```

### Étude du programme :

Ligne 1 : **Pour (=for)** une variable **i allant (=in) de (=range)** 1 à 6

Ligne 2 : affiche le message

### QUESTIONS :

- 1- Écrire et exécuter le programme 20.
- 2- Indiquer la cohérence qu'il semble y avoir : **Le message est répété 5 fois, le programme fait [1,6[**
- 3- Modifier le programme en tenant compte des lignes ci-dessous :

```
1 ▾ for i in range(6) :  
2   | print ("Je répète 6 fois le message")
```

- 4- Comparer la réponse du programme avec la question 2. **Le message est répété 6 fois**
- 5- Modifier le programme 20 (voir ci-dessous) et exécuter le.

```
1 ▾ for i in range(1,6) :  
2   | print ("je répète 3 fois un message",i)
```

- 6- Que remarquez-vous ? : **La variable i a compté chaque ligne**
- 7- Remplacer la variable **i** par **a** et exécuter de nouveau le programme. 

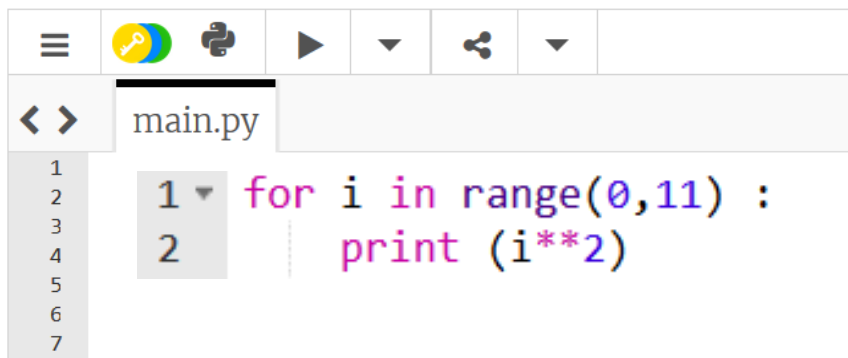
```
1 ▾ for a in range(1,6) :  
2   | print ("je répète 6 fois un message",a)
```
- 8- Que remarquez-vous ? : **Le résultat est le même, on peut donner le nom que l'on souhaite au variable.**
- 9- Modifier et exécuter votre programme pour avoir en troisième ligne `print ("Au revoir")` indentée dans la boucle. 

```
1 ▾ for a in range(1,6) :  
2   | print ("je répète 6 fois un message",a)  
3   | print ("Au revoir")
```
- 10- Même question si l'on souhaite cette ligne en non indentée. 

```
1 ▾ for a in range(1,6) :  
2   | print ("je répète 6 fois un message",a)  
3 print ("Au revoir")
```

## Exercices :

- 1- Écrire un programme qui affiche tous les nombres carrés des nombres entiers de 0 à 10.



```
1
2
3
4
5
6
7
1 for i in range(0,11) :
2     print (i**2)
```

- 2- Écrire un programme qui affiche tous les nombres pairs compris entre 0 et 10, avec à la fin de ce programme le texte Au revoir.

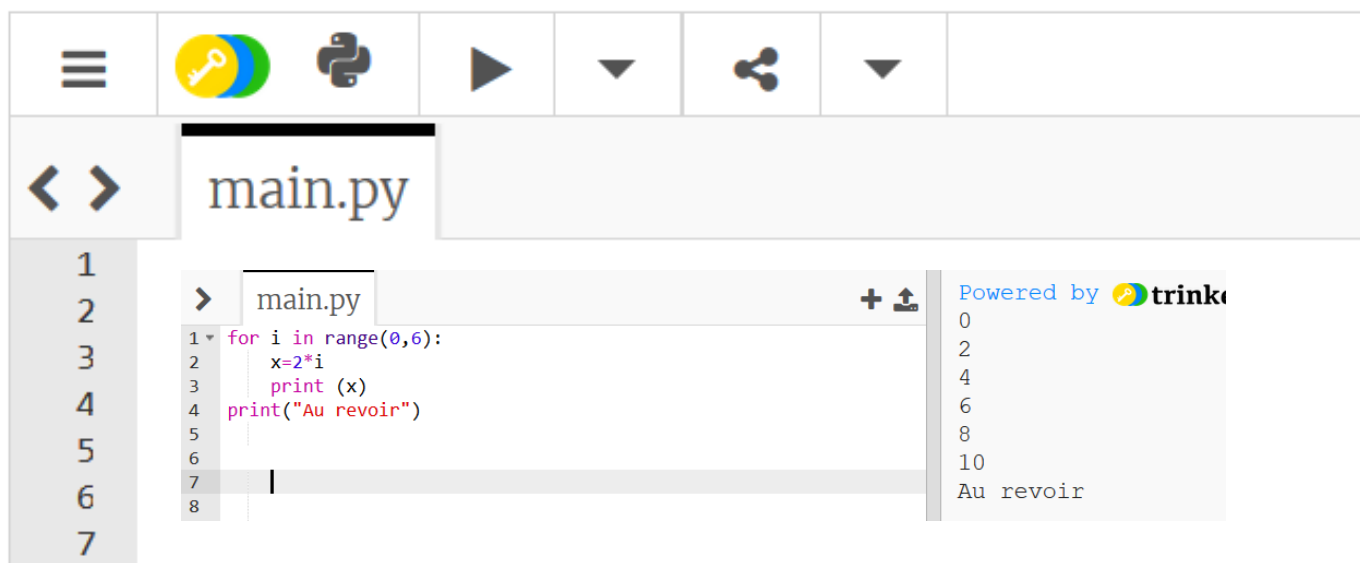
Votre programme contiendra en :

Ligne 1 : la boucle for


Ligne 2 : la formule permettant d'obtenir les nombres pairs dans une variable

Ligne 3 : l'affichage des nombres

Ligne 4 : l'affichage de « au revoir » à la fin du programme



```
1
2
3
4
5
6
7
> main.py
1 for i in range(0,6):
2     x=2*i
3     print (x)
4     print("Au revoir")
5
6
7
8
```

Powered by  trinket

0  
2  
4  
6  
8  
10  
Au revoir

Aide :

Si dans un ensemble  $N \in [a,b]$  l'on souhaite obtenir les :

- Nombres pairs d'un ensemble, on fait  $2xN$
- Nombres impairs d'un ensemble, on fait  $2*N+1$

3- Modifier ce programme pour qu'il affiche tous les nombres impairs compris entre 0 et 10.

The screenshot shows a Python IDE interface. At the top, there is a toolbar with icons for a menu, a key, a Python logo, a play button, a dropdown arrow, a share icon, and another dropdown arrow. Below the toolbar, the file name 'main.py' is displayed. On the left side, there is a vertical line of numbers from 1 to 7, indicating line numbers. The main code area contains the following Python code:

```
1 for i in range(0,5):  
2     x=2*i+1  
3     print (x)  
4     print("Au revoir")  
5  
6  
7
```

On the right side, there is a console area with the text 'Powered by trinket' and the output of the program:

```
1  
3  
5  
7  
9  
Au revoir
```