

Definitions

Dans un algorithme, une boucle est une suite d'instructions que l'on répète (en boucle) un certain nombre de fois. On dit que la boucle est bornée lorsque l'on sait exactement combien de fois on va répéter la boucle. En python, pour effectuer N répétitions, on utilise l'instruction:

```
for i in range(N):
```

Exemple n°1 : range(N)

On souhaite obtenir la table de multiplication par 3 à l'aide d'un algorithme. On sait que :

$3 \times 0 = 0$	$3 \times 4 = 12$	$3 \times 8 = 24$
$3 \times 1 = 3$	$3 \times 5 = 15$	$3 \times 9 = 27$
$3 \times 2 = 6$	$3 \times 6 = 18$	$3 \times 10 = 30$
$3 \times 3 = 9$	$3 \times 7 = 21$	$3 \times 11 = 33$

On remarque que l'opération effectuée est toujours la même : $3 \times i$ où i est un entier compris entre 0 et 11. Voici alors l'algorithme et le programme associé :

```
# Algorithme
1. Pour i allant de 0 à 11 faire:
2.     Afficher 3 * i
3. Fin Pour
```

```
# Python
1. for i in range(12):
2.     print(3 * i)
```

Dans l'algorithme, les instructions qui seront répétées sont celles dans le bloc compris entre la ligne "Pour" et la ligne "Fin Pour". Une répétition du bloc est aussi appelé une **ITERATION**.

A chaque itération, le variant i augmente d'une unité: 0, puis 1, puis 2, ...

C'est ce qui permet d'afficher les termes de la table de 3, avec l'instruction $3 * i$

Lorsque l'on arrive à $i = 12$, la boucle s'arrête **sans exécuter** le bloc.

Les affichage successifs seront: **0, 3, 6, 9 ... 33**

Exemple n°2 : range(1,N)

On peut souhaiter démarrer la boucle avec, comme première valeur 1 pour le variant i .

Dans ce cas, on ajoute un argument à la fonction range.

```
# Algorithme
1. Pour i allant de 1 à 11 faire:
2.     Afficher 3 * i
3. Fin Pour
```

```
# Python
1. for i in range(1,12):
2.     print(3 * i)
```

Ici, la boucle va commencer avec $i = 1$, puis $i = 2$ à la deuxième itération ... $i = 11$ à la dernière.

Elle va donc s'exécuter 11 fois., et afficher: **3, 6, 8, ... 33**

Exercices

Exercice 1 : On donne le programme python qui affiche les valeurs des entiers:

- 1) Quelle est la dernière valeur affichée par le programme?
- 2) Combien y-a-t-il d'itérations?

Exercice 2: La fonction python ci-contre affiche toutes les valeurs paires entre les bornes [a,b], où a et b sont des entiers.

```
# Python
1. def nombres_paires(a,b):
2.     for i in range(a,b):
3.         if i%2==0:
4.             print(i)
```

- 1) Qu'est ce qui est affiché lorsque l'on execute l'instruction:

```
nombres_paires(0,7)
```

- 2) Quelle instruction, utilisant cette fonction, va afficher les nombres pairs entre 10 et 100?

Exercice 3:

- 1) Écrire un programme qui affiche la table de multiplication par 4 depuis 4×0 jusqu'à 4×10 .
- 2) Modifier ce programme pour qu'il affiche la table de multiplication par 4 depuis 4×3 jusqu'à 4×8 .
- 3) Écrire une fonction TableMultiplication(n) qui affiche la table de multiplication par n depuis $n \times 0$ jusqu'à $n \times 10$.

Exercice 4: fonction avec valeur de retour (return)

La fonction ci-contre retourne la chaine de caractères "ha" un nombre n de fois, où n est le paramètre de la fonction.

```
# Python
1. def affiche(n):
2.     return n * "ha"
```

- 1) Qu'est ce qui est affiché lorsque l'on execute les instructions:

```
mot = affiche(3)
print(mot)
```

- 2) Quelle instruction, utilisant cette fonction, va afficher le mot "hahahahahahahahahahaha" ?

Exercice 5 : On considère la fonction f définie sur l'ensemble des réels par $f: x \rightarrow x^2$

- 1) Définir cette fonction avec Python.
- 2) Écrire un programme qui retourne les images des entiers entre -3 et 3 par la fonction .
- 3) Modifier ce programme pour n'afficher que les images des entiers pairs entre -3 et 3. (sans modifier f).

Exercice 6: On considère le programme suivant où deux boucles "for" sont emboîtées :

```
# Python
1. for i in range(5,10):
2.     for j in range(4):
3.         print(i - j)
```

- 1) Quelle est la première valeur des variants i et j au début du programme ?
- 2) Après l'affichage de la première valeur, quel variant va changer en premier : i ou j ? Quelle est alors la deuxième valeur affichée ?
- 3) Exécuter ce programme pour visualiser les résultats