

# Python : les boucles:

La notion de boucle est fondamentale en informatique. Une boucle permet d'exécuter plusieurs fois des instructions qui ne sont présentes qu'une seule fois dans le code.

## 1. La boucle for (pour):

La structure de la boucle for est la suivante:

```
0 for i variant de 0 a n:  
    instruction_1  
2    instruction_2
```

prgm.py

**Exercice 31:** Soit le programme suivant :

```
0 for i in range(10):  
    print(i)
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme ? Vérifiez votre réponse en testant le programme.

**Exercice 32:** "un générateur automatique de punition" (boucle for):

Écrire une fonction qui prendra 2 paramètres : une chaîne de caractère et un nombre entier. Par exemple :

Si on passe comme paramètres à notre fonction : "Je ne dois pas discuter en classe" et 3

La fonction devra permettre d'afficher :

```
Je ne dois pas discuter en classe  
Je ne dois pas discuter en classe  
Je ne dois pas discuter en classe
```

## 2. La boucle while (tant que):

La structure de la boucle while est la suivante:

```
0 while expression :
1     instruction_1
2     instruction_2
```

prgm.py

Tant que l'expression s'évalue à "True", les instructions à l'intérieur du bloc (partie indentée) seront exécutées.

**Exercice 33:** Soit le programme suivant :

```
0 i = 0
1 while i < 10:
2     print(i)
3     i = i+1
4 print('c est termine')
```

prgm.py

Quel est le résultat attendu après l'exécution de ce programme ? Vérifiez votre réponse en testant le programme.

**Exercice 34:** "un générateur automatique de punition" (boucle while):  
Même exercice mais avec une boucle while.

**Exercice 35:** Écrire une fonction permettant d'afficher une table de multiplication. Cette fonction devra prendre en paramètre la table désirée.

Par exemple si l'on passe le paramètre 3 à la fonction, la fonction devra permettre d'afficher :

```
1 x 3 = 3
2 x 3 = 6
...
...
10 x 3 = 30
```