



Table des matières

1	Les boucles bornées.....	1
2	Les boucles non bornées.....	1
3	Les boucles imbriquées.....	2

1 Les boucles bornées

Une boucle permet de répéter une ou plusieurs instructions.

Lorsque l'on connaît le nombre de répétitions, on utilise une **boucle bornée**.

L'instruction Python correspondant à une boucle bornée est « **for indice in range()** » :

- « **in range()** » permet d'énumérer à une boucle de passage dans la boucle bornée
- L'**indice** *i* de la boucle prend les valeurs entières de 0 à *n*-1, donc *n* valeurs. Il est possible de préciser la première et la dernière valeur de l'indice dans la boucle en définissant des bornes. Lorsque l'indice ne doit pas varier de 1 en 1, on peut également modifier le nombre de pas qui sépare un indice du suivant.

Exemple : compte à rebours du déclenchement d'un appareil photo, la boucle s'effectuera 5 fois et l'indice *i* prend les valeurs entières de 0 à 4

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

affichera :

5
4
3
2
1

Exemple : on cherche à afficher une ligne sur deux d'un tableau de la 10^e ligne à la 19^e ligne. Voici le programme qui donne les numéros de lignes à afficher

```
for i in range(10,20 ,2) :  
    print(i)
```

affichera :

10
12
14
16
18

2 Les boucles non bornées

Lorsque le nombre de répétitions n'est pas connu à l'avance, on utilise une **boucle non bornée**. Elle permet de répéter un bloc d'instruction tant qu'une condition est vérifiée.

La structure correspondant à une boucle non bornée est « **while condition** : »

Si la condition est toujours réalisée, la boucle se répétera indéfiniment. Il est donc important de vérifier que la condition cesse d'être vraie à terme afin que la boucle se termine.

Le nombre de passage dans une boucle non bornée étant inconnu au départ, il peut être nécessaire de créer une variable comptant le nombre de tours. Elle peut indiquer à partir de quand une condition n'est plus vérifiée.



Une variable servant de **compteur** permet de compter le nombre de passages dans la boucle dans une boucle non bornée.

Exemple : Programme qui donne le nombre d'années pour que la consommation d'internet atteigne 6000TWh/an. Celle-ci est multipliée par 1,2 tous les ans. La variable « années » sert de compteur

```
consommation=1500
annees=0
while consommation<6000 :
    consommation=consommation*1.2
    annees=annees+1
print("Nombre d'années pour atteindre 6000 Wh : ", annees)
```

Affichera : Nombre d'années pour atteindre 6000 Wh : 8

Notes reprenez le caractère d'échappement d\'années pour faire apparaître l'apostrophe !

3 Les boucles imbriquées

Il est possible d'imbriquer des boucles. A chaque passage dans la première boucle, la deuxième boucle est effectuée entièrement.

Exemple : programme qui affiche les tables de multiplications de 7, 8 et 9. La première boucle fait varier i de 7 à 9 et pour chaque valeur de i, la variable j varie de 1 à 3 dans la deuxième boucle.

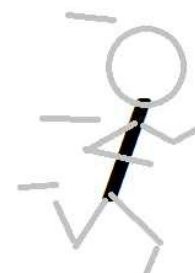
```
for i in range(7,10):
    print ("table de ",i)
    for j in range(1,4):
        print(i*j)
```

Affichera :

```
table de 7
7
14
21
table de 8
8
16
24
table de 9
9
18
27
```



Bumbling backslash falls backwards



Speedy slash sprints forwards