

Boucle non bornée

NUMWORKS

On utilise généralement la boucle **while** lorsqu'on souhaite répéter un nombre de fois une même instruction et qu'on ne sait pas combien de fois cette instruction va être répétée.

On connaît alors une condition d'arrêt, c'est-à-dire un test qui permet de savoir si l'instruction va être répétée ou non.

Par exemple, si on veut ajouter 1 à un nombre tant que ce nombre est inférieur à 50, on utilisera une boucle **while** : "TANT QUE nombre < 50, ajouter 1 à nombre".

En Python, la structure de la boucle **while** est la suivante :

```
while condition:
    instruction
```

Dans notre exemple, pour un nombre initial égal à 8, on écrira donc :

A la fin de la boucle, on aura :

En effet, lorsque nombre contient 49 la condition est toujours vérifiée et l'instruction s'exécute. **nombre** est donc incrémenté de 1 et vaut alors 50.

La condition n'est plus vérifiée et la boucle **while** se termine.

La fiche 2 contenait la syntaxe des conditions possibles en Python. Elles sont répétées ci-dessous.

Condition	Syntaxe Python
x est égal à y	<code>x == y:</code>
x est différent de y	<code>x != y:</code>
x est strictement supérieur à y	<code>x > y:</code>
x est strictement inférieur à y	<code>x < y:</code>
x est supérieur ou égal à y	<code>x >= y:</code>
x est inférieur ou égal à y	<code>x <= y:</code>

1 Exercice

Écrire une fonction **max_cube(n)** qui prend un entier naturel n en argument et qui renvoie le plus grand entier dont le cube est inférieur ou égal à n.

2 Autre exercice

Écrire une fonction **facteurs(n)** qui prend un nombre entier en argument et renvoie la liste des facteurs de sa décomposition en facteurs premiers.

```
deg PYTHON
nombre = 8
while nombre<50:
    nombre=nombre+1|
```

FIGURE 1 - *

image 1

```
deg PYTHON
>>> from exemple import *
>>> nombre
50
>>> |
```

FIGURE 2 - *

image 2