

Boucle non bornée while

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.

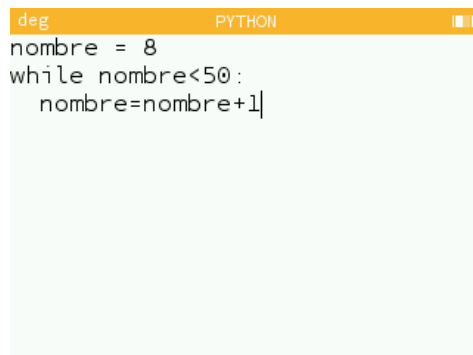
On connaît alors une condition d'arrêt, c'est-à-dire un test qui permet de savoir si l'instruction va être répétée ou non.

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 :

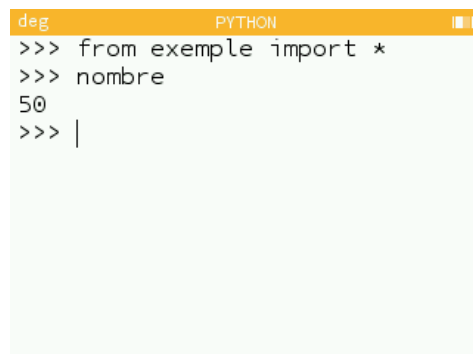
```
while condition:  
    instruction
```

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

A screenshot of a Python code editor window titled 'deg' and 'PYTHON'. The code inside is:

```
nombre = 8  
while nombre < 50:  
    nombre = nombre + 1
```

A la fin de la boucle, on aura :

A screenshot of a Python REPL window titled 'deg' and 'PYTHON'. The code inside is:

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

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 sur les instructions conditionnelles 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>

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 .

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.