

## Boucle WHILE (tant que)

La boucle "tant que" est l'instruction conditionnelle qui permet de répéter un bloc d'instructions tant qu'une condition donnée reste **vraie**.

**Algorithm:**

```
tant-que une condition reste vraie faire
    ce bloc d'instructions
fin du tant-que
```

**Syntaxe:**

```
while <condition> :
    instruction 1
    instruction 2
    .
    .
STOP: fin indentation
```

**Remarque 1** Toujours les deux points au bout de la ligne du `while` et l'indentation du bloc d'instructions...

**Remarque 2** En lançant une telle boucle, on obtient l'exécution répétée du bloc d'instructions tant que *condition* reste une variable booléenne vraie. Dès que *condition* devient fausse, Python s'arrête. A retenir:

On sort de la boucle quand la condition est FAUSSE

**Remarque 3 : ATTENTION AUX BOUCLES INFINIES!!**

La condition est réévaluée à chaque passage. Il faut qu'elle évolue, sinon l'itération peut ne pas s'arrêter. On dit alors que l'on boucle indéfiniment.

En effet, même si le nombre d'itérations dans une boucle `while` est inconnu, il doit rester fini afin d'éviter de boucler indéfiniment. Pour cela, il faut s'assurer qu'une instruction provoque l'arrêt de la boucle: le bloc d'instructions agit sur la condition de sorte que celle-ci, vraie au départ, devienne fausse (après une ou plusieurs itérations).

**Exercice 1 (algorithme de seuil)**

Écrire une fonction `premier(M)` qui prend en argument un réel *M* (positif strictement) et qui renvoie le premier entier *n* tel que  $n! \geq M$ .

**Remarque 4** Une boucle `while` est donc plus compliquée à mettre en place qu'une boucle `for` car c'est à l'utilisateur de "gérer" le compteur:

- initialisation **AVANT** la boucle,
- incrémentation **DANS** la boucle (`a=a+1` par exemple)

**Exercice 2** On considère la suite  $(u_n)$  définie par:

$$u_0 = 1 \text{ et } \forall n \geq 1, u_{n+1} = u_n + \frac{1}{u_n}.$$

Montrer que la suite  $(u_n)$  est bien définie, qu'elle est croissante et qu'elle diverge vers  $+\infty$ .  
En déduire une fonction `seuil(M)` qui , pour tout réel  $M$ , renvoie le premier  $n$  tel que  $u_n \geq M$ .