I6 - Utiliser une boucle while

Syntaxe de la boucle while

```
while condition:
    instructions
```

Python exécute alors les instructions suivantes :

- ⋆ Si la condition est fausse, il n'exécute pas les instructions indentées et il sort de la boucle while (il passe directement aux éventuelles instructions qui suivent la boucle while).
- ⋆ Si la condition est vraie, il exécute les instructions indentées puis
  - si la condition est devenue fausse, il sort de la boucle while ;
  - si la condition est encore vraie, il exécute de nouveau les instructions indentées puis recommence de la même manière en vérifient de nouveau la condition.

Si la condition reste tout le temps vraie, on parle de **boucle infinie** : en général ce n'est pas une bonne chose mais dans de rares cas ça peut être utile.

Dérouler une boucle while

Comme pour une boucle `for`, on peut « dérouler une boucle `while` » pour comprendre ce qu'il se passe exactement et repérer d'éventuelles erreurs.

Par exemple pour la boucle suivante :

```
u = 2
n = 1
while u < 100:
    u = 2 * u
    n = n + 1
```

Étape	`u`	`n`
0	2	1
1	$4 = 2^2$	2
2	$8 = 2^3$	3
3	$16 = 2^4$	4
4	$32 = 2^5$	5
5	$64 = 2^6$	6
6	$128 = 2^7$	7

La boucle s'arrête alors car la condition `u < 100` n'est plus vraie. On peut identifier un invariant de boucle : à chaque étape, on a $u = 2^n$.

Ici on a pu faire le tableau entier mais souvent on ne peut faire que les premières lignes pour vérifier que tout se passe comme on l'avait prévu.

Trouver la première/dernière valeur de n vérifiant une propriété

Soit $(u_n)_{n\geqslant 0}$ la suite définie par

$$u_0 = 2 \qquad \text{et} \qquad \forall n \in \mathbb{N}, \quad u_{n+1} = u_n * *2$$

Trouvons la plus grande valeur de n telle que $u_n < 10^6$.

```
u = 2
n = 0
while u < 10**6:
    u = u**2
    n = n + 1
print(n)
```

Déroulons la boucle pour vérifier :

Étape	u	n
0	$2 = u_0$	0
1	$4 = u_1$	1
2	$16 = u_2$	2
3	$16^2 = u_3$	3
⋮	⋮	⋮
k	u_k	k
⋮	⋮	⋮

Lorsque la boucle s'arrête, on a $u = u_n \geqslant 10^6$: il faut donc afficher $n-1$ au lieu de n :

```
u = 2
n = 0
while u < 10**6:
    u = u**2
    n = n + 1
print(n-1)
```

Erreurs fréquentes

- **Confondre if et while**
  Les syntaxes sont les mêmes (seul le mot-clé change) mais dans le cas d'un `if`, les instructions indentées sont exécutées au plus une seule fois. Cela arrive souvent lorsqu'on veut combiner une boucle `for` et une instructions conditionnelle. Par exemple on écrit

  ```
  u = 2
  for i in range(n):
      while u % 2 == 0:
          u = u + 1
  ```

  au lieu de

  ```
  u = 2
  for i in range(n):
      if u % 2 == 0:
          u = u + 1
  ```

- **Ne pas modifier la variable utilisée dans la condition**
  C'est la boucle infinie assurée ! Si la condition est vraie au début mais qu'on ne change jamais la variable concernée par cette condition, elle restera vraie pour toujours ! Par exemple

```
u = 2
n = 0
while u < 10**6:
    n = n + 1
print(n-1)
```