Python 6 – **Boucles TANT QUE / WHILE : non bornées**

CC BY NC SA

I – Boucle NON BORNEE : principe

La boucle TANT QUE est dite **non bornée** car on n'a pas besoin de connaître **à l'avance** le nombre de fois où elle va se dérouler.

Le principe est simple : lorsqu'on tombe sur le début de la boucle, on évalue l'expression fournie.

Si elle est vraie : on _____ de boucle et on _____ au début.

Si elle est fausse : on arrête la boucle en cherchant la première instruction hors boucle (non indentée).

```
L1 début du programme
L2
L3 while condition de poursuite:
L4     instruction A
L5     instruction B
L6     instruction C
L7
L8 suite du programme
```

```
Deux possibilités de séquence :
Soit L3(poursuite = True)  – L4 – L5 – L6 puis L3
Soit L3(poursuite = False) - L8
```

II – Exemple classique : le seuil

Imaginons qu'on double notre argent tous les mois. On veut savoir combien de temps nous aurons 500 euros.

```
1   epargne = 10
2   nb_mois = 0
3
4   while epargne < 500:
5       epargne = epargne * 2
6       nb_mois = nb_mois + 1
7
8   print(epargne)
9   print(nb_mois)
```

Traduction de la ligne 4 : Tant que notre épargne est inférieur à 500 euros, réalise les lignes 5 et 6.

La **tabulation** (4 espaces) permet à l'interpréteur de savoir où se trouvent les instructions à réaliser en boucle.

L'interpréteur Python exécute les lignes dans cet ordre :

L01(epargne=10) - L02(nb_mois=0)

L04(True) - L05(epargne=_____) - L06(nb_mois=____)

L04(True) - L05(epargne=_____) - L06(nb_mois=____)

L04(True) - L05(epargne=_____) - L06(nb_mois=____)

L04(True) - L05(epargne=_____) - L06(nb_mois=____)

L04(True) - L05(epargne=_____) - L06(nb_mois=____)

L04(True) - L05(epargne=_____) - L06(nb_mois=____)

L04(False car epargne = _____) - L08 - L09

III – Première évaluation de la condition

Les variables nécessaires à l'évaluation du TANT QUE doivent être définies **avant** la ligne du TANT QUE.

Si la condition est basée sur une variable qui est calculée dans le bloc de la boucle elle-même, comment faire ? **Initialiser cette variable avec une "fausse première valeur"** de façon à rentrer dans la boucle au départ. Il suffit de mettre 0 en ligne 3.

```
01  from random import randint
02
03  valeur = 0
04
05
06  while valeur != 6:
07      valeur = randint(1, 6)
08      print(valeur)
09
10  print("Bravo, vous avez tiré un 6 !")
```

Exemple de déroulé

Imaginons qu'on tire 4, 2 et 6 en ligne 7.

01 - L03(**valeur=0**)

L06(True car valeur est différent de 6) - L07(**valeur=4** maintenant) - L08

L06(True car valeur est différent de 6) - L07(**valeur=2** maintenant) - L08

L06(True car valeur est différent de 6) - L07(**valeur=6** maintenant) - L08

L06(False car valeur n'est pas différent de 6) – L10