

# COURS 5 : Programmer une boucle non bornée – 1 heure

Une **boucle non bornée** ou **boucle while** ou **boucle tant que** permet de répéter un bloc d'instruction tant qu'une certaine condition est vérifiée.

## Programme 21 :

```
1 import random
2 nombre_aleatoire = random.randint(1,10)
3
4 print("Un entier vient d'être tiré au sort entre 1 et 10 inclus.")
5
6 reponse = 0
7 compteur = 0
8
9 while reponse != nombre_aleatoire:
10     reponse = int(input("Devinez l'entier tiré au sort :"))
11     compteur = compteur+1
12
13 print("Bravo... ")
14 print("Vous avez trouvé en",compteur, "fois")
```

## Étude du programme :

Le programme est un jeu qui choisit un nombre de façon aléatoire entre 1 et 10.

Vous le joueur devait trouver le nombre.

Le programme indique le nombre de coup pour trouver le nombre

### **Ligne 1 :**

Un tirage aléatoire demande l'appel d'un module nommé **random**.

Le programme doit démarrer par la ligne : **import random**

### **Ligne 2 :**

La variable se nomme **nombre\_aleatoire**. (les espaces entre les mots sont remplacés par « \_ » lorsque l'on souhaite mettre du texte comme une variable)

Elle sera donc aléatoire (=random)

La variable est comprise entre les valeurs 1 et 10 (=randint(1,10))

### **Ligne 4 :**

Le programme affiche le message

### **Ligne 6 :**

La réponse de l'utilisateur sera une variable portant le nom de **reponse**.

Elle est égale à 0 mais peut importe, car l'utilisateur sera amenée à la changer (voir ligne 10).

### **Ligne 7 :**

On introduit une variable qui porte le nom de compteur démarrant à 0.

Ce compteur permettra de compter les lignes exécutées à partir de la boucle (voir ligne 11).

### **Ligne 9 :**

Démarrage de la boucle tant que (=while) la variable **reponse** est différente de la variable **nombre\_aleatoire**

#### **Ligne 10 :**

La boucle while vérifie la valeur entrée par l'utilisateur dans la variable **reponse** en posant la question (=input)

#### **Ligne 11 :**

Le compteur est indenté dans la boucle while. La variable nommée compteur ajoute +1 à chaque ligne tant les deux variables sont différentes.

### **Ligne 13 et 14 :**

Affichage de ces deux lignes si les variables **reponse** et **nombre\_aleatoire** sont égales

## QUESTIONS :

- 1- Recopier le programme 21 et l'exécuter
- 2- Modifier la ligne 6 du programme en entrant une autre valeur numérique différente à la variable et vérifier si l'exécution du programme en est modifiée.  
Tester avec plusieurs valeurs.

Que remarquez-vous ? :

**Aucune différence car la valeur de la variable change de valeur à la demande du programme (ligne 10)**

- 3- Modifier la ligne 7 du programme en entrant « **compteur = 1** » et vérifier si l'exécution du programme en est modifiée.

Que remarquez-vous ? :

**Le compteur a démarré à 1 au lieu de 0, il affiche donc 1 tentative en plus.**

- 4- Modifier la ligne 11, par « **compteur+=1** »

Que remarquez-vous ? :

N'oublier pas de remettre la variable « **compteur=0** » à la ligne 7.

**Rien n'a changé, `compteur = compteur + 1` est semblable à `compteur+=1`**

- 5- Soit le programme suivant

```
1 p=1
2 while p<=10:
3     p=p*2
4     print(p)
```

Sans utiliser l'ordinateur, expliquer le fonctionnement de ce programme.

**Ligne 1 :**

**La variable p prend la valeur 1**

**Ligne 2 :**

**Tant que  $p \leq 10$  le calcul  $p=2p$  est réalisé**

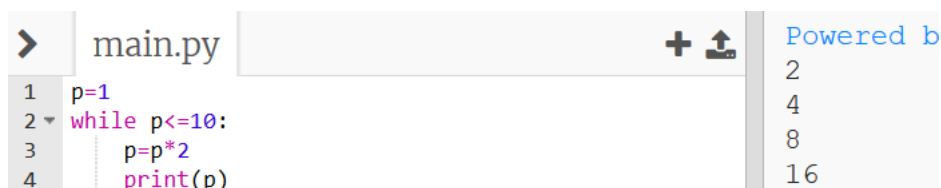
**On démarre à  $P=1 \leq 10 \Rightarrow$  le programme affiche le résultat de  $2xp \Rightarrow 2$**

**Maintenant  $P=2 \leq 10 \Rightarrow$  le programme affiche le résultat de  $2xp \Rightarrow 4$**

**Maintenant  $P=4 \leq 10 \Rightarrow$  le programme affiche le résultat de  $2xp \Rightarrow 8$**

**Maintenant  $P=8 \leq 10 \Rightarrow$  le programme affiche le résultat de  $2xp \Rightarrow 16$**

**Maintenant  $P=16 \leq 10$  n'est plus vérifié donc le programme n'affiche plus rien.**



```
> main.py + [icon] Powered b
1 p=1
2 while p<=10:
3     p=p*2
4     print(p)
2
4
8
16
```