

1 Boucle Tant que

Créez le script suivant :

```
N = 100
i = 0
while (2**i < N) :
    print(i)
    i = i + 1
print('fin')
```

Exécutez ce script. Que fait-il ? Essayez d'en déduire le fonctionnement de la boucle `while`.

Le déroulement de la boucle est le suivant :

1. On teste la condition (`2**i < N`).
2. Si la condition est vérifiée (c'est-à-dire que l'évaluation de l'expression booléenne renvoie `True`), Python effectue les instructions `print(i)` et `i=i+1` puis recommence le test.
3. Si la condition n'est pas vérifiée (c'est-à-dire que l'évaluation de l'expression booléenne renvoie `False`), on sort de la boucle et Python effectue l'instruction `print('fin')`.

Remarques :

- La variable `i` est un compteur, sa valeur finale permet de savoir combien de fois on a répété la boucle.
- En Python, pour incrémenter un compteur on peut utiliser l'instruction `i += 1` à la place de `i = i + 1`.
- Quand on écrit une boucle, il faut impérativement s'assurer qu'elle va réellement s'arrêter sinon l'exécution ne peut pas se terminer.

2 Exercices

Exercice 1

On considère la fonction suivante :

```
def boucle(A,B) :
    i = 0
    while (A > B) :
        i += 1
        A = A - i
        B = B + i**2
    return A,B
```

Exécuter à la main cette fonction avec les paramètres `A=100` et `B=0`. On pourra représenter dans un tableau les valeurs de `i`, de `(A>B)`, de `A` et de `B` à chaque étape.

Exercice 2

Écrire une fonction `depasse` qui prend comme paramètre un nombre `n` et renvoie le plus grand entier `i` tel que $i^2 + 3i - \frac{1}{3} < n$.

Exercice 3

Écrire une fonction qui prend en entrée un paramètre `x` et un paramètre `n` et renvoie, grâce à une boucle, la valeur de x^n

3 Boucle Pour

Quand on sait à l'avance le nombre de répétitions dont on aura besoin dans une boucle, on peut utiliser une boucle pour (for avec Python) qui dispose d'un compteur intégré.

Créez le script suivant :

```
N = eval(input('Entrez un nombre entre 1 et 10 : '))
U = 1
V = 1
for i in range(0,N):
    U = U + 2
    V = V * 2
    print('valeur de U : ',U)
    print('valeur de V : ',V)
```

Exécutez plusieurs fois ce script (en testant différentes valeurs pour N).

Comment fonctionne la boucle for ?

Remarque 1 : avec une boucle for, on n'a pas besoin d'un compteur, c'est la variable `i` qui joue le rôle de compteur.

Remarque 2 : Il est possible de mentionner un pas autre que 1. En ce cas, on écrira `in range(a,b,pas)`.

4 Exercices

Exercice 1

Écrire une fonction `table` qui prend en paramètre un entier n et affiche la table de multiplication de n .

Exercice 2

Écrire une fonction `puissance` qui prend deux paramètres `x` et `n` et qui calcule x^n en utilisant une boucle `for`.

Exercice 3

On veut estimer expérimentalement le nombre moyen obtenu lors d'une série de lancers de dé (supposé équilibré).

1. Quelle est la valeur théorique de la moyenne obtenue lorsqu'on lance un dé équilibré ?
2. Créer une fonction `experimental` qui prend en entrée le nombre `n` de lancers à effectuer et calcule le total de ces `n` lancers.
3. Modifier la fonction pour qu'elle renvoie la valeur moyenne des lancers.
4. Tester `experimental(10)`, `experimental(100)`, `experimental(1000)`, `experimental(10000)`. Que constate-t-on ?

Exercice 4

On veut créer le programme suivant : l'ordinateur choisit un nombre entier au hasard entre 1 et 100 puis demande à l'utilisateur de rentrer des nombres entiers jusqu'à deviner le nombre choisi par l'ordinateur.

1. Créez un script effectuant cette démarche
2. Modifiez le script pour que le programme affiche, à chaque proposition de l'utilisateur, si le nombre donné est trop grand ou trop petit par rapport à celui cherché.
3. Modifiez le script pour que le programme n'accepte que 8 propositions au maximum.