

Les boucles

i Les boucles font partie des grandes structures de base de l'algorithmique, et donc de la programmation.
Il s'agit de répéter une séquence d'instructions un certain nombre de fois.
On dit que l'on réalise une **itération** de la boucle à chaque fois que la séquence d'instruction est exécutée.

Il existe deux types de boucle :

- Les **boucles bornées ou itératives**, dont on connaît par avance le nombre de répétitions souhaitées
- Les **boucles conditionnelles**, dont on ne connaît pas le nombre de répétitions mais qui possèdent une condition d'arrêt.

Les boucles itératives

• structure d'une boucle itérative

En Python, lorsque que l'on connaît le nombre de répétitions à effectuer, on utilise généralement la boucle itérative **for**

A chaque itération, Python va prendre un élément de l'itérable et de la boucle à chaque fois que le corps de la boucle est exécuté. Le bloc d'instructions à répéter est délimité par l'**indentation**.

```
for variable in itérable :  
    bloc d'instructions indenté à répéter
```

Définition

Un itérable est un objet dont on peut parcourir les valeurs une par une.
Les types **list**, **tuple**, **dict**, **set** sont des exemples d'itérable.

• Répétition à l'aide de l'instruction **range**

- L'instruction **range**
-

Pour obtenir un itérable contenant **n** valeurs, on peut utiliser l'instruction **range**

Exemple

```
1 iterable = range(5)
```

A la suite de cette instruction, la variable nommée **iterable** contiendra les valeurs :

0, 1, 2, 3, 4
5 valeurs

- Les options
-

- Préciser le début `range(a,b)` : un itérable qui contient les nombres compris entre `a` inclus et `b` exclus.

Exemple : `range(3,8)` donnera 3, 4, 5, 6, 7.

- Préciser le pas `range(a,b, pas)` : un itérable qui contient les nombres compris entre `a` inclus et `b` exclus en incrémentant à chaque fois de `pas`.

Exemple : `range(3,11,2)` donnera 3, 5, 7, 9.

•• Construction d'une boucle avec l'instruction `range` _____

Comme l'instruction `range` nous fournit un itérable contenant un nombre de valeurs que l'on souhaite, il est fréquent de l'utiliser dans les boucles itératives.

La boucle sera donc construite à l'aide de l'instruction `for var in range(nb_de_repetition):` et le bloc d'instructions à répéter est délimité par l'**indentation**.

```
for i in range(n) :  
    bloc d'instructions indenté à répéter
```

Exemple :

L'exécution du code suivant

Affichera :

```
1 for _ in range(4):  
2     print("Hello")
```

Hello
Hello
Hello
Hello

i L'instruction `range(4)` crée un objet de type `range` qui contient les valeurs 0, 1, 2 et 3.

Quand on écrit l'instruction : `for i in range(4):`
A chaque répétition de la boucle, la valeur de la variable `i` change et prend successivement les valeurs de l'objet `range`

Exercice 1 ★

Quelle sera l'affichage, après l'exécution du code ci-dessous

```
1 for i in range(4):  
2     print("Hello")  
3     print("Word")
```

.....

.....

.....

.....

.....

.....

Exercice 2 ★ Debuguer

Retrouver les erreurs dans les codes ci-dessous.

```

1 for _ in range(4)           1 for _ range(4):           1 for _ in range(4):
2 print("Bla")                2 print("Hello")           2 print("Bonjour")

```

.....

.....

.....

Exercice 3 ★★

Après exécution du code ci-dessous, quelle sera la valeur de la variable **res**.

```

1 res = 1
2 for i in range(5):
3     res += i*3

```

.....

.....

.....

Exercice 4 ★

```

1 x = 2
2
3 for i in range(4):
4     x = x * 3
5     print(f"Étape {i} : x = {x}")

```

1. Recopier et complétez le tableau d'avancement suivant en mettant à jour la valeur de **x**.

Valeur de (i)	Valeur de x
0	2
1	
2	
3	

Exercice 5 ★

Voici le programme Python correspondant :

```

1 b = 3
2
3 for i in range(1, 6):
4     b = 2*b + 3*i
5     print(f"Étape {i} : b = {b}")

```

1. Recopier et complétez le tableau avancement suivant en mettant à jour la valeur de **x**.

Itération (i)	Valeur de b
1	
2	
3	
4	
5	

Exercice 6 ★

Voici le programme Python correspondant :

```

1  b = 0
2
3  for i in range(1, 10, 2):
4      b = b*i

```

1. Donner tableau suivant avancement en mettant à jour la valeur de **x**.

Exercice 7

Le nombre d'utilisateurs d'un nouveau réseau social est égal à 500 000 en janvier 2020. Ce nombre augmente de 5% par mois chaque mois.

On souhaite connaître le nombre d'utilisateur chaque mois, pour cela on aimerait utiliser une variable **utilisateurs**

1. Quelle instruction permet d'initialiser une variable **utilisateurs** qui représente le nombre d'utilisateurs du réseau social.
2. Quelle instruction permet de prendre en compte l'augmentation au bout d'un mois et de mettre à jour la variable **utilisateur**.
3. Écrivez un programme en Python utilisant pour calculer le nombre d'utilisateurs après 10 mois.
4. Modifiez votre programme pour demander à l'utilisateur de saisir le nombre de mois **n** et calculez le nombre d'utilisateurs après **n** mois.

Exercice 8 ★

Écrire un programme qui calcule et affiche la somme des nombres de 1 à 99.

.....

.....

.....

.....

.....

.....

● Itérer sur une chaîne de caractères

i Les objets itérables sont les objets que l'on peut parcourir à l'aide d'un `for`. Il possède généralement la méthode `__iter__`.

Les types `str`, `list`, `dict`, `tuple` sont les itérables que nous allons voir cette année.

Exemples :

Code

```
1 for mot in ["Hello", "Word"]:  
2     print(mot)
```

Sortie
Hello
Word

Code

```
1 for lettre in "Hello":  
2     print(mot)
```

Sortie
H
e
l
l
o

Exercice 9 ★★

Complete le programme ci-dessous pour qu'il affiche le nombre de caractères de la chaîne `ch`.

```
1 ch = "Vive les NSI"  
2 counter = .....  
3 for .....  
4 .....  
5 .....  
6
```

Exercice 10 ★★★

Écrire un programme qui affiche le nombre de `s` contenu dans un mot saisi par l'utilisateur.

Exemple :

- Si l'utilisateur saisit le mot `bonjour`, l'affichage sera 0
- Si l'utilisateur saisit le mot `salut`, l'affichage sera 1
- Si l'utilisateur saisit le mot `saisissant`, l'affichage sera 4