

## □ La Boucle for

**Objectif :** Comprendre et utiliser la boucle **for** pour répéter un bloc un nombre **connu** de fois.

### □ Concept

La boucle **for** parcourt une séquence (liste, chaîne, **range**) et exécute le bloc pour chaque élément.

python

```
for i in range(debut, fin+1, pas):  
    # traitement
```

| Opération                | Code Python                        |
|--------------------------|------------------------------------|
| Répéter N fois (0 à N-1) | for i in range(N):                 |
| De début à fin inclus    | for i in range(debut, fin+1):      |
| Avec un pas              | for i in range(debut, fin+1, pas): |

□□ La valeur finale est **exclue** dans range() — utiliser **fin+1** pour l'inclure.

## □ Bloc 1 — Séquences et comptages

### □□ Exercice 1 — Afficher les nombres de 1 à 10

□ **Objectif :** Afficher les nombres de 1 à 10 à l'aide d'une boucle for.

terminal

```
1  
2  
3  
...  
10
```

### □□ Exercice 2 — Afficher les nombres pairs de 2 à 20

□ **Objectif :** Afficher uniquement les nombres pairs de 2 à 20.

 terminal

```
2
4
6
...
20
```

### □□ Exercice 3 — Compte à rebours

□ **Objectif** : Afficher les nombres de 10 à 1, puis afficher "Décollage !".

 terminal

```
10
9
...
1
Décollage !
```

## □ Bloc 2 — Accumulations

### □□ Exercice 4 — Somme de 1 à n

□ **Objectif** : Lire un entier **n** puis afficher la somme des nombres de 1 à n.

 terminal

```
Entrez n :
> 5
15
```

### □□ Exercice 5 — Somme de 1 à 100

□ **Objectif** : Calculer et afficher la somme des nombres de 1 à 100 sans saisie.

 terminal

```
Somme = 5050
```

### □□ Exercice 6 — Calcul de moyenne des notes

□ **Objectif** : Demander combien de notes l'utilisateur veut entrer, récupérer chaque note, les afficher une à une, puis afficher la moyenne.

```
terminal

Combien de notes ?
> 3
Note 1 :
> 14
Note 2 :
> 17
Note 3 :
> 15
Moyenne : 15.33
```

## □ Bloc 3 — Chaînes et parcours

### □□ Exercice 7 — Afficher les lettres d'un mot

□ **Objectif** : Afficher chaque lettre du mot "Python" sur une ligne différente.

```
terminal

P
y
t
h
o
n
```

### □□ Exercice 8 — Inverser une phrase

□ **Objectif** : Demander une phrase à l'utilisateur et l'afficher lettre par lettre à l'envers.

```
terminal

Entrez une phrase :
> Bonjour
r
u
```

```
o
j
n
o
B
```

### □□ Exercice 9 — Compter une lettre dans une phrase

□ **Objectif** : L'utilisateur entre une phrase et une lettre. Afficher combien de fois la lettre apparaît (ignorer majuscules/minuscules). Ne pas utiliser `.count()` !

```
terminal
Phrase :
> Python est populaire
Lettre :
> p
La lettre 'p' apparaît 2 fois.
```

## □ Bloc 4 — Formes et motifs

### □□ Exercice 10 — Pyramide d'étoiles

□ **Objectif** : Demander `n` et afficher une pyramide d'étoiles de `n` lignes.

□ Construire chaque ligne en concaténant des étoiles dans une variable chaîne, puis afficher avec `print()`.

```
terminal
Entrez n :
> 4
*
**
***
****
```

### □□ Exercice 11 — Triangle inversé

□ **Objectif** : Demander `n` et afficher un triangle inversé d'étoiles avec `n` lignes.

 terminal

Entrez n :

&gt; 4

\*\*\*\*

\*\*\*

\*\*

\*

## □□ Exercice 12 — Carré d'étoiles

□ **Objectif** : Demander un entier **n** et afficher un carré d'étoiles de n lignes et n colonnes.

□ Pour chaque ligne, construire une chaîne de n étoiles puis l'afficher avec print().

 terminal

Entrez n :

&gt; 4

\*\*\*\*

\*\*\*\*

\*\*\*\*

\*\*\*\*

## □□ Exercice 13 — Table de multiplication d'un nombre

□ **Objectif** : Demander un nombre à l'utilisateur puis afficher sa table de multiplication jusqu'à 10.

 terminal

Entrez un nombre :

&gt; 3

3 x 1 = 3

3 x 2 = 6

...

3 x 10 = 30

## □ Bloc 5 — Boucles imbriquées

Une boucle **imbriquée** est une boucle à l'intérieur d'une autre. Pour chaque itération de la boucle externe, la boucle interne s'exécute entièrement.

python

```
for i in range(n) :      # boucle externe – lignes
    for j in range(n) :  # boucle interne – colonnes
        # traitement de la case (i, j)
```

□ Pour afficher plusieurs valeurs sur une même ligne : construire une chaîne **ligne** dans la boucle interne, puis faire **print(ligne)** après la boucle interne.

## □□ Exercice 14 — Rectangle de chiffres

□ **Objectif** : Afficher un rectangle 4×5 où chaque cellule contient le numéro de colonne (1 à 5).

terminal

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

## □□ Exercice 15 — Matrice d'indices

□ **Objectif** : Afficher les coordonnées ligne-colonne d'une matrice 3×3.

terminal

```
(1,1) (1,2) (1,3)
(2,1) (2,2) (2,3)
(3,1) (3,2) (3,3)
```

## □□ Exercice 16 — Tableau de multiplication n×n

□ **Objectif** : Demander un entier **n** et afficher le tableau de multiplication de 1 à n.

Complétez le code :

python

```
n = int(input('Entrez n : '))
for i in range(1, n + 1) :
    ligne = ''
    for j in range(1, n + 1) :
```

```
ligne = ligne + str(i * j) + '\t'  
print(ligne)
```

terminal

Entrez n :

> 4

1 2 3 4

2 4 6 8

3 6 9 12

4 8 12 16

### □□ Exercice 17 — Diagonale principale □

□ **Objectif** : Même tableau que l'exercice 16, mais remplacer les éléments de la diagonale principale ( $i == j$ ) par le symbole □.

terminal

Entrez n :

> 4

★ 2 3 4

2 ★ 6 8

3 6 ★ 12

4 8 12 ★

## □ Récapitulatif

## □ Boucle for

|                      |                                                 |
|----------------------|-------------------------------------------------|
| N fois (0 à N-1)     | <code>for i in range(N):</code>                 |
| Début à fin inclus   | <code>for i in range(debut, fin+1):</code>      |
| Avec un pas          | <code>for i in range(debut, fin+1, pas):</code> |
| Parcourir une chaîne | <code>for lettre in mot:</code>                 |
| Accumuler            | <code>somme = somme + i</code>                  |
| Concaténer une ligne | <code>ligne = ligne + str(val) + ' '</code>     |
| Boucle imbriquée     | <code>for i in ...: for j in ...:</code>        |
| Diagonale            | <code>if i == j : # case spéciale</code>        |