

---

## Boucle for

## Exercices

---

### Exercice 1

Complétez le programme qui affiche tous les nombres pairs entre 1 et 20 (inclus).

**Instructions :**

- Utilise une boucle for pour parcourir les nombres de 1 à 20.
- Vérifie si un nombre est pair (un nombre est pair s'il est divisible par 2 sans reste).
- Affiche uniquement les nombres pairs.

```
for i in range(..., ...): # Parcours des nombres de 1 à 20
    if i % 2 == 0: # Vérifie si le nombre est pair
        print(...) # Affiche le nombre pair
```

## Exercice 2

Ecrire un programme qui demande un premier entier  $a$ , puis demande un entier  $b$ . Puis calcule  $(a + b)^4$ . **Ne pas utiliser** la fonction puissance en Python.

## Exercice 3

Ecrire un programme qui demande un entier  $n$  à l'utilisateur, puis calcule et affiche le résultat de la multiplication  $2 \times 2 \times 2 \dots \times 2$  (où on a  $n$  occurrences de 2).

## Exercice 4

Ecrire un programme qui calcule et affiche  $1 \times 2 \times \dots \times 100$ .

## Exercice 5

### Calcul de la somme des cubes des entiers pairs

#### Enoncé :

Ecrire un programme qui calcule la somme des cubes des entiers pairs compris entre 1 et un nombre entier  $n$  donné par l'utilisateur. Utilisez une boucle for avec un accumulateur pour obtenir le résultat final.

#### Instructions :

1. Le programme demande à l'utilisateur d'entrer un nombre entier positif  $n$ .
2. Utilise une boucle for pour parcourir tous les nombres entiers pairs entre 1 et  $n$ .
3. Calcule le cube de chaque entier pair et ajoute ce cube à une variable accumulateur.
4. À la fin de la boucle, affiche la somme totale des cubes des entiers pairs.

#### Exemple :

Si l'utilisateur entre  $n = 6$ , le programme doit afficher **288**, car la somme des cubes des entiers pairs entre 1 et 6 est :  $2^3 + 4^3 + 6^3 = 8 + 64 + 216 = 288$ .

#### Complément :

Elever une variable  $i$  à la puissance 3 s'écrit : `i ** 3`

## Exercice 6

### Compte à rebours

Écrire un programme qui affiche un compte à rebours à partir d'un nombre donné par l'utilisateur jusqu'à zéro.

#### Instructions :

- Demande à l'utilisateur de saisir un nombre entier positif.
- Utilise une boucle for pour afficher le compte à rebours en partant de ce nombre jusqu'à zéro inclus.
- Chaque nombre doit être affiché sur une ligne.

## Exercice 7

### Triangle de nombres

Écrire un programme qui affiche un triangle de nombres en fonction d'un nombre entier donné par l'utilisateur. Le triangle doit avoir autant de lignes que la valeur entrée, et chaque ligne doit afficher des nombres en ordre croissant, en commençant toujours par 1.

#### Instructions :

- Demandez à l'utilisateur de saisir un nombre entier positif n.
- Utilisez des boucles for imbriquées pour afficher un triangle de nombres.
- Chaque ligne contient un nombre croissant d'éléments, la première ligne affiche 1, la deuxième ligne affiche 1 2, la troisième affiche 1 2 3, etc.

Si l'utilisateur entre 5 :

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

## Exercice 8

### Triangle inversé

Énoncé : Écris un programme qui affiche un triangle d'étoiles inversé. Le nombre d'étoiles commence à n et diminue de 1 à chaque ligne.

**Exemple de sortie :** Si l'utilisateur entre 5, le programme devra afficher :

```
*****
****
***
**
*
```

## Exercice 9

### Générer un tableau de multiplication

#### Enoncé :

Écrire un programme qui affiche un tableau de multiplication pour des nombres allant de 1 à un entier  $n$  donné par l'utilisateur. Utilisez deux boucles for imbriquées pour afficher le tableau.

#### Instructions :

1. Le programme demande à l'utilisateur d'entrer un nombre entier positif  $n$ .
2. Le programme doit afficher un tableau où chaque case contient le produit des indices de ligne et de colonne. Par exemple, la première ligne du tableau affiche les produits de 1 à  $n$  multipliés par 1, la deuxième ligne affiche les produits de 2 à  $n$  multipliés par 2, etc.
3. Utilisez deux boucles imbriquées pour calculer et afficher le tableau.

#### Exemple :

Si l'utilisateur entre  $n = 4$ , le programme doit afficher :

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

Coup de pouce :

- Comment afficher la valeur d'une variable *ma\_variable* sans retour à la ligne ?

```
print(ma_variable, end="\t")
```

- Comment passer à la ligne suivante après chaque ligne ?

```
print()
```