

Python: Les boucles for (exercices 🦊)

Voici des **exercices progressifs**, directement alignés avec l'article sur les boucles **for**. Ils vont du **très guidé** vers le **raisonnement autonome**, sans introduire de notions non vues.

5GMS

3TTR



Exploration

Ces exercices te permettent de t'entraîner à utiliser les boucles **for** pour répéter des actions et parcourir des séquences (**range** , chaînes de caractères).

01 – Dire bonjour plusieurs fois

`01_bonjour_5_fois.py`

Afficher le mot **"Bonjour"** **5 fois** à l'aide d'une boucle **for**.

02 – Compter de 0 à 4

`02_compter_0_a_4.py`

Afficher les nombres de **0** à **4**, chacun sur une ligne.

03 – Compter de 1 à 10

`03_compter_1_a_10.py`

Afficher les nombres de **1** à **10** inclus, chacun sur une ligne.

04 – Afficher les nombres pairs

`04_nombres_pairs_0_20.py`

Afficher tous les nombres pairs entre **0** et **20** (inclus).

05 – Table de multiplication de 5

`05_table_5.py`

Afficher la table de multiplication de `5` (de `5 × 1` à `5 × 10`).

06 – Table de multiplication (nombre de l'utilisateur)

`06_table_nombre_utilisateur.py`

Demander un nombre à l'utilisateur et afficher sa table de multiplication (de `0` à `10`).

07 – Carrés des nombres

`07_carres_1_a_10.py`

Afficher le carré des nombres de `1` à `10`.

Exemple attendu :

```
1 → 1
2 → 4
3 → 9
```

08 – Parcourir un mot

`08_parcourir_python.py`

À partir du mot `"Python"`, afficher chaque lettre sur une ligne différente.

09 – Parcourir un mot donné par l'utilisateur

`09_parcourir_mot_utilisateur.py`

Demander un mot à l'utilisateur et afficher chaque lettre sur une nouvelle ligne.

10 – Compter les lettres d'un mot

10_compter_lettres_mot.py

Demander un mot à l'utilisateur et afficher le **nombre total de lettres**, en comptant avec une boucle `for` (sans utiliser `len()`).

11 – Afficher une phrase caractère par caractère

11_phrase_caractere_par_caractere.py

À partir de la phrase `"Je code en Python"`, afficher chaque caractère (y compris les espaces), un par ligne.

12 – Compter les lettres "a"

12_compter_a.py

Demander un mot à l'utilisateur et compter combien de fois la lettre `"a"` apparaît.

13 – Somme des nombres

13_somme_1_a_100.py

Calculer la somme des nombres de `1` à `100` à l'aide d'une boucle `for`.

14 – Afficher les positions d'un mot

14_positions_programmation.py

Afficher les nombres de `0` jusqu'à la longueur du mot `"Programmation"` moins 1.

15 – Mot vertical numéroté

15_mot_vertical_numerote.py

Demander un mot à l'utilisateur et afficher chaque lettre précédée de son numéro de position (à partir de 1).

Exemple :

```
1 : P
2 : y
3 : t
4 : h
5 : o
6 : n
```

16 – Exercice bonus : Le jeu du 21 (version `for`)

`16_bonus_jeu_21_for.py`

Le joueur commence avec un score de `0`. Le jeu se déroule en **5 tours maximum** à l'aide d'une boucle `for`.

À chaque tour :

- un nombre aléatoire entre **1 et 6** est généré ;
- ce nombre est ajouté au score ;
- le tirage et le score actuel sont affichés.

À la fin du jeu :

- si le score est **exactement 21** → afficher `"Bravo, 21 !"`
- si le score est **inférieur à 21** → afficher `"Pas assez"`
- si le score est **supérieur à 21** → afficher `"Trop élevé"`

Contraintes :

- utiliser une boucle `for`
 - utiliser une variable de cumul `score`
 - pas de boucle `while`
 - pas de `break`
-