
Boucle for

Exercices corrigés

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
```

Correction :

```
for i in range(1, 21): # Parcours des nombres de 1 à 20
    if i % 2 == 0: # Vérifie si le nombre est pair
        print(i) # 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.

Correction :

```
a = int(input("Entrez un nombre entier positif : "))
b = int(input("Entrez un nombre entier positif : "))

calcul = (a + b)
for _ in range(3) :
    calcul = calcul * (a + b)

print("Resultat :", calcul)
```

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).

Correction :

```
p = 1
for _ in range(n):

    p = 2 * p

print(p)
```

Exercice 4

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

Correction :

```
f = 1
for n in range(1, 101):

    f = f * n

print(f)
```

Exercice 5

Calcul de la somme des cubes des entiers pairs

Énoncé :

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. Utilisez une boucle for pour parcourir tous les nombres entiers pairs entre 1 et n .
3. Calcule le cube de chaque entier pair et ajoutez ce cube à une variable accumulateur.
4. À la fin de la boucle, affichez 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 :

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

Correction :

```
# Demander à l'utilisateur d'entrer un nombre entier positif
```

```
n = int(input("Entrez un nombre entier positif : "))
```

```
# Initialiser l'accumulateur
```

```
somme = 0
```

```
# Boucle pour parcourir les entiers pairs de 2 à n
```

```
for i in range(2, n+1, 2):
```

```
    # Ajouter le cube de l'entier pair à la somme
```

```
    somme += i ** 3
```

```
# Afficher la somme des cubes des entiers pairs
```

```
print("La somme des cubes des entiers pairs est :", somme)
```

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.

Correction :

```
# Demander à l'utilisateur de saisir un nombre entier positif

n = int(input("Entrez un nombre entier positif pour commencer le compte à rebours : "))

# Boucle for pour faire un compte à rebours de n à 0

for i in range(n, -1, -1): # Départ de n jusqu'à 0 inclus, en décrémentant de 1
    print(i)
```

Exercice 7

Carré d'étoiles

Écrire un programme qui demande à l'utilisateur d'entrer un nombre entier représentant un nombre de lignes, puis un nombre de colonnes et qui affiche un carré d'étoiles (*) avec les dimensions données précédemment.

Aides :

Pour afficher sur une même ligne : `print("*", end="")`

Pour sauter de ligne : `print()`

```
lignes = int(input("Entrez un nombre entier pour les lignes : "))
colonnes = int(input("Entrez un nombre entier pour les colonnes : "))

for i in range(0, lignes):
    for j in range(0, colonnes) :
        print("*", end="")
    print()
```

Exercice 8

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
```

Correction :

```
# Demander à l'utilisateur de saisir un nombre entier positif
```

```
n = int(input("Entrez un nombre entier positif : "))
```

```
# Boucle for pour créer le triangle de nombres
```

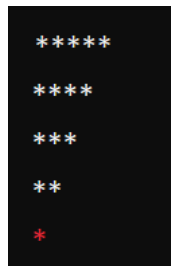
```
for i in range(1, n + 1): # Contrôle le nombre de lignes (de 1 à n)
```

```
    for j in range(1, i + 1): # Affiche les nombres de 1 à i sur chaque ligne
```

```
        print(j, end=" ") # Affiche les nombres sur la même ligne avec un espace
```

```
    print() # Passe à la ligne suivante
```

Exercice 9 : Triangle inversé



```
# Demander à l'utilisateur d'entrer un nombre entier
n = int(input("Entrez un nombre entier : "))

# Boucle for pour afficher chaque ligne du triangle inversé
for i in range(n, 0, -1):
    # Afficher i étoiles sur la ligne
    print("*" * i)
```


Exercice 10

Générer un tableau de multiplication

Énoncé :

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()
```

Correction :

```
# Demander à l'utilisateur d'entrer un nombre entier positif
n = int(input("Entrez un nombre entier positif : "))

# Boucle pour les lignes
for i in range(1, n + 1):
    # Boucle pour les colonnes
    for j in range(1, n + 1):
        # Afficher le produit de i et j, sans retour à la ligne
        print(i * j, end="\t")
    # Passer à la ligne suivante après chaque ligne
    print()
```