

Répéter comme un perroquet avec un Python

Exercice 1 Les répétitions □

La commande `range(n)` où n est un entier fait prendre à la variable les valeurs entières de ... à ...

- 1. Exécuter le programme ci-dessous :

```
for i in range(4):  
    print(i)
```

- 2. La commande `range(n, m)` où n et m sont des entiers fait prendre à la variable les valeurs entières de n à $m - 1$.

Recopier et exécuter le programme ci-dessous :

```
for i in range(12, 42):  
    print(i)
```

- 3. La commande `range(n, m, k)` où n , m et k sont des entiers fait prendre à la variable les valeurs entières de n et $m - 1$ avec un pas de k .

Exécuter le programme ci-dessous :

```
for i in range(12, 42, 3):  
    print(i, end='-')
```

- 4. Recopier et exécuter le programme ci-dessous :

```
for i in range(1, 11):  
    print("5 x", i, '=', 5 * i)
```

Que fait ce programme ?

- 5. En utilisant une boucle `for`, écrire un programme permettant d'obtenir l'affichage ci-dessous :

```
Le carré de 3 est 9  
Le carré de 4 est 16  
Le carré de 5 est 25  
Le carré de 6 est 36  
Le carré de 7 est 49  
Le carré de 8 est 64  
Le carré de 9 est 81
```

- 6. Écrire un programme permettant d'afficher dans cet ordre à l'aide d'une boucle les nombres :

7 9 11 13 15 17

- 7. Écrire un programme permettant d'afficher dans cet ordre et à la suite, à l'aide d'une boucle, les nombres : 10 9 8 7 6 5 4 3 2 1 0

Exercice 2 □

1. Calculer la somme S des n termes :

$$S = 1 + 2 + 3 + \dots + n$$

2. Modifier votre programme pour calculer la somme des cubes des n premiers termes.

Exercice 3 □

L'objectif de cet exercice est d'écrire un programme qui compte le nombre de mots dans une phrase simple.

Astuce : Il faut compter le nombre de caractère ... +1.

- 1. Compléter tester le programme suivant :

```
phrase = "Bonjour, il fait beau"  
...  
for lettre in phrase:  
    if .....:  
        ...  
    else:  
        ...  
print('Le nombre de mots est :',  
      NbMots)
```

- 2. Créer un programme qui compte le nombre de voyelle dans une phrase.

- 3. Compléter : dans la phrase, il y a ... mots et ... voyelles.

Exercice 4 Motifs □

1. Réaliser les programmes des motifs 1 et 2, expliquer succinctement.

2. Réaliser les motifs 3, 4 et 5 pour une taille choisie en utilisant des programmes les plus courts possibles.

□ Motif 1

```
for i in range(11):  
    print("*" * (6 - abs(5 - i)))  
.....
```

□ Motif 2

```
n = int(input('taille'))  
for i in range(n):  
    for j in range(n):  
        if i == j or j == n - 1 or i == 0:  
            print('*', end='')  
        else:  
            print(' ', end='')  
    print()      #Passage à la ligne  
.....
```

Motif 3

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

Motif 4

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

Motif 5 : Bonus

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

Lors de vos programmes, vous avez rencontré beaucoup d'erreurs que vous avez su résoudre. Ces erreurs font parties de l'apprentissage ! Mais comment y remédier ? Pour cela, il faut une rigueur absolue pour écrire un code Python. C'est pourquoi, on a défini les règles d'écriture suivantes :

- Mettre un espace après et avant les signes.
- Mettre un espace après une virgule.
- Respecter l'alignement.
- Choisir des noms de variables très parlants et en accord avec le type.
- #Ajouter des commentaires comme ceci OU `"""Comme cela pour expliquer votre code"""`
- Privilégier l'anglais dans le choix des variables et les commentaires afin qu'un maximum de personnes puisse le lire. `My_Name = "spyder"`.
- DRY : ne vous répétez pas, Don't Repeat Yourself en anglais.

Exercice 5 Être rigoureux en Python et pourquoi

1. Modifier pour expliquer ce code afin de le rendre très lisible et concis.
2. Préciser ce que fait ce programme.

.....
.....
.....

```
from math import sqrt  
Nbre_Rentre_Utilisateur=int(input('nombre'))  
a=True  
for i in range(2,Nbre_Rentre_Utilisateur-1):  
    if Nbre_Rentre_Utilisateur%i==0:  
        a=False  
print(a)
```

Exercice 6 Bonus : Code ISBN

Le code ISBN : L'International Standard Book Number (ISBN) à 13 chiffres est un numéro international qui permet d'identifier de manière unique, chaque édition de chaque livre publié. Les trois premiers chiffres sont 978, suivis de 9 chiffres. Le dernier chiffre est une clef obtenue par le calcul suivant :

1. Additionner les 12 chiffres du code ;
2. Y ajouter le double de la somme des chiffres de rang pairs (2^e , 4^e , 6^e , 8^e , 10^e , 12^e) ;
3. Prendre le dernier chiffre de cette somme ;
4. Calculer la différence entre 10 et le résultat obtenu.

1. Vérifier à la main la clé de l'exemple.
2. Écrire un programme Python qui vérifie qu'un code ISBN est correct.

