

# exBouclesPython

October 8, 2019

## 1 Exercice

1. Créer une boucle FOR qui affiche 4 fois le message "Bienvenue"
2. Modifier votre boucle pour que le message soit affiché autant de fois qu'une valeur saisie avant la boucle.
3. Modifier enfin votre code pour que le message soit numéroté :

```
Bienvenue 1 fois\\
Bienvenue 2 fois\\
etc.
```

```
In [6]: for _ in range(4):
        print("Bienvenue")
```

```
Bienvenue
Bienvenue
Bienvenue
Bienvenue
```

```
In [7]: n=int(input("Nombre de répétitions:"))
        for _ in range(n):
            print("Bienvenue")
```

```
Nombre de répétitions:3
Bienvenue
Bienvenue
Bienvenue
```

```
In [8]: n=int(input("Nombre de répétitions:"))
        for i in range(n):
            print("Bienvenue",i+1,"fois")
```

Nombre de répétitions:5  
Bienvenue 1 fois  
Bienvenue 2 fois  
Bienvenue 3 fois  
Bienvenue 4 fois  
Bienvenue 5 fois

## 2 Exercice

1. Créer une boucle FOR qui affiche sur une même ligne les nombres de 0 à 9 séparés par un tiret : 0 - 1 - 2 - ...
2. Modifier votre boucle pour qu'elle affiche les nombres en commençant à 5 : 5 - 6 - 7 - ....
3. Modifier votre boucle qu'elle affiche les nombres progressant de 3 en 3 : 0 - 3 - 6 - ...
4. Modifier votre boucle pour afficher les nombres dans l'ordre décroissant de 9 à 0 : 9 - 8 - 7 - 6 - ...

```
In [9]: for i in range(10):  
        print(i,end="-")
```

0-1-2-3-4-5-6-7-8-9-

```
In [10]: for i in range(5,10):  
         print(i,end="-")
```

5-6-7-8-9-

```
In [11]: for i in range(0,10,3):  
         print(i,end="-")
```

0-3-6-9-

```
In [13]: for i in range(10):  
         print(9-i,end="-")
```

9-8-7-6-5-4-3-2-1-0-

## 3 Exercice

1. Écrire un programme qui affiche les nombres de 0 à 99 sur 10 lignes, chaque nombre étant séparé par un tiret.
2. Proposer une seconde version afin qu'il n'y ait pas de tiret à la fin de chaque ligne.

```
In [15]: for i in range(10):  
         for k in range(10):  
             print(k+10*i,end="-")  
         print()
```

0-1-2-3-4-5-6-7-8-9-  
10-11-12-13-14-15-16-17-18-19-  
20-21-22-23-24-25-26-27-28-29-  
30-31-32-33-34-35-36-37-38-39-  
40-41-42-43-44-45-46-47-48-49-  
50-51-52-53-54-55-56-57-58-59-  
60-61-62-63-64-65-66-67-68-69-  
70-71-72-73-74-75-76-77-78-79-  
80-81-82-83-84-85-86-87-88-89-  
90-91-92-93-94-95-96-97-98-99-

```
In [17]: for i in range(10):  
         for k in range(9):  
             print(k+10*i,end="-")  
         print(k+10*i+1)
```

0-1-2-3-4-5-6-7-8-9  
10-11-12-13-14-15-16-17-18-19  
20-21-22-23-24-25-26-27-28-29  
30-31-32-33-34-35-36-37-38-39  
40-41-42-43-44-45-46-47-48-49  
50-51-52-53-54-55-56-57-58-59  
60-61-62-63-64-65-66-67-68-69  
70-71-72-73-74-75-76-77-78-79  
80-81-82-83-84-85-86-87-88-89  
90-91-92-93-94-95-96-97-98-99

```
In [20]: for i in range(10):  
         for k in range(9):  
             print("%02d" % (k+10*i),end="-")  
         print("%02d" % (k+10*i+1))
```

00-01-02-03-04-05-06-07-08-09  
10-11-12-13-14-15-16-17-18-19  
20-21-22-23-24-25-26-27-28-29  
30-31-32-33-34-35-36-37-38-39  
40-41-42-43-44-45-46-47-48-49  
50-51-52-53-54-55-56-57-58-59  
60-61-62-63-64-65-66-67-68-69  
70-71-72-73-74-75-76-77-78-79  
80-81-82-83-84-85-86-87-88-89  
90-91-92-93-94-95-96-97-98-99

## 4 Exercice

1. On souhaite réaliser le motif en "X" ci-dessous. Il est composé de 10 lignes et 10 colonnes.
  - a) Créer une première boucle qui permet de créer une première ligne. Pour afficher du texte sur une même ligne, il faut ajouter le paramètre **end** à la fonction **print** : `print(message, end="")`.
  - b) Créer une boucle qui va répéter 10 fois la boucle précédente. On dit que ce sont des boucles imbriquées.
2. Modifier votre code pour que le nombre de lignes et colonnes soit saisi par l'utilisateur.
3. Réaliser les motifs suivants de 10 lignes et 10 colonnes:

```
In [24]: for _ in range(10):
         for _ in range(10):
             print("X",end="")
         print()
```

```
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
```

```
In [23]: for _ in range(10):
         print("X",end="")
         print()
         for _ in range(8):
             print("X",end="")
             for _ in range(8):
                 print(" ",end="")
             print("X",end="")
             print()
         for _ in range(10):
             print("X",end="")
         print()
```

```
XXXXXXXXXX
X          X
X          X
X          X
```

```

X      X
X      X
X      X
X      X
X      X
XXXXXXXXXX

```

```

In [5]: for _ in range(5):
        for _ in range(5):
            print("HI",end="")
        print()
        for _ in range(5):
            print("IH",end="")
        print()

```

```

HIHIHIHIHI
IHIIHIHIH
HIHIHIHIHI
IHIIHIHIH
HIHIHIHIHI
IHIIHIHIH
HIHIHIHIHI
IHIIHIHIH
HIHIHIHIHI
IHIIHIHIH

```

```

In [25]: n=int(input("n="))
        for _ in range(n):
            print("X",end="")
        print()
        for k in range(1,n//2):
            for _ in range(k):
                print(" ",end="")
            print("X",end="")
            for _ in range(k+1,n-k-1):
                print(" ",end="")
            print("X",end="")
            for _ in range(n-k,n//2):
                print(" ",end="")
            print()
        for k in range(n//2,n-1):
            for _ in range(n-k-1):
                print(" ",end="")
            print("X",end="")
            for _ in range(n-k,k):
                print(" ",end="")

```

```
n=10
XXXXXXXXXX
 X        X
  X      X
   X    X
    XX
    XX
   X  X
  X    X
 X      X
XXXXXXXXXX
```

```

        P=P*(k+1)
    print("S=",S)
    print("P=",P)

```

Nombre entier positif:10

S= 55

P= 3628800

## 6 Exercice

1. Saisir dans l'interpréteur la fonction "bin(9)". Quel est le résultat affiché ?
2. L'objectif est de créer une boucle WHILE qui donne l'écriture binaire d'un nombre entier positif en utilisant les divisions successives.
  - a) On utilisera les variables n pour le nombre entier positif, b pour l'écriture binaire obtenue qui sera de type "string" et r pour les restes des divisions entières.
  - b) On calculera les divisions en mémorisant le reste dans une variable r et le nouveau quotient dans n.
  - c) Le nombre n est divisé par 2 jusqu'à devenir nul. On fait les calculs tant que n est différent de 0.
  - d) On construit le nombre binaire b comme chaîne de caractères constituée des restes.
3. On pourra faire une vérification avec la fonction int(chaine,base).

```

In [35]: # On affiche l'écriture binaire du nombre 9
        bin(9)

```

```

Out[35]: '0b1001'

```

```

In [50]: n=int(input("Nombre entier positif:"))
        b=''
        while n!=0:
            r=n%2
            b=str(r)+b
            n=n//2
        print(b,"est l'écriture binaire de",int(b,2))

```

Nombre entier positif:999

1111100111 est l'écriture binaire de 999