

## Boucles et Ascii-art

### *Ascii-art*

En combinant des boucles et des textes, on peut faire des “dessins” à base de lettres et de symboles. On appelle cela de “l’Ascii art”, l’Ascii étant une des techniques d’encodage des symboles dans un ordinateur.

```
>>> for i in range(5):  
    print("@"*i)
```

```
@  
@@  
@@@  
@@@@
```

On remarque que la première ligne est vide. Si on veut éviter cela, on peut faire :

```
>>> for i in range(5):  
    print("@"*(i+1))
```

```
@  
@@  
@@@  
@@@@  
@@@@@
```

Ou :

```
>>> for i in range(1, 6):  
    print("@"*i)
```

```
@  
@@  
@@@  
@@@@  
@@@@@
```

Créez un nouveau fichier `feuille-boucles-ascii-NOM-PRENOM.py` et recopiez les lignes ci-dessous :

```
def carre(n):  
    for i in range(n):  
        print("@"*n)  
  
def triangle1(n):  
    for i in range(n):  
        print("@"*(i+1))
```

Ces fonctions permettent d'obtenir les figures suivantes :

```
>>> carre(2)
@@
@@
>>> carre(3)
@@@
@@@
@@@
```

```
>>> triangle1(2)
@
@@
>>> triangle1(3)
@
@@
@@@
```

Rajoutez les lignes suivantes :

```
def diagonale1(n):
    for i in range(n):
        print(" "*i+"\\"")
```

On remarque que pour avoir le symbole “\” il faut en mettre 2. C’est parce qu’il y a certains caractères spéciaux, comme par exemple \' ou \" qui permettent d’avoir des apostrophes et des guillemets dans du texte.

```
>>> diagonale1(3)
\
 \
  \
```

### ? EXERCICE 1 :

Rajoutez la fonction triangle2(n) qui permet d'obtenir la figure suivante :

```
>>> triangle2(3)
@@@
 @@
  @
```

### ? EXERCICE 2 :

Rajoutez la fonction triangle3(n) qui permet d'obtenir la figure suivante :

```
>>> triangle3(3)
@@@
@@
@
```

### ? EXERCICE 3 :

Rajoutez la fonction `triangle4(n)` qui permet d'obtenir la figure suivante :

```
>>> triangle4(3)
@
@@
@@@
```

### ? EXERCICE 4 :

Rajoutez la fonction `diagonale2(n)` qui permet d'obtenir la figure suivante :

```
>>> diagonale(3)
/
/
/
```

Rajoutez la fonction suivante :

```
def motif1(n):
    for i in range(n):
        print("|"*i+"/"+"-"*(n-i-1))
```

```
>>> motif1(3)
/--
|/-
||/
```

### ? EXERCICE 5 :

Rajoutez la fonction `motif2(n)` qui permet d'obtenir la figure suivante :

```
>>> motif2(3)
||\
|\-
\--
```

### ? EXERCICE 6 :

Rajoutez la fonction `motif3(n)` qui permet d'obtenir la figure suivante :

```
>>> motif3(3)
/----\
|/--\|
||/\||
||\|||
|\--/|
\----/
```

Vous aurez peut-être besoin d'utiliser deux boucles l'une après l'autre.

### ? EXERCICE 7 :

Rajoutez la fonction `pyramide(n)` qui permet d'obtenir les figures suivantes :

```
>>> pyramide(3)
```

```
  @
```

```
  @@@
```

```
 @@@@@
```

```
>>> pyramide(4)
```

```
  @
```

```
  @@@
```

```
 @@@@@
```

```
 @@@@@@@
```