

TP_Art_ASCII

January 11, 2021

1 Art-ASCII

L'Art-ASCII consiste à réaliser des images uniquement à l'aide des lettres et caractères spéciaux contenus dans le code ASCII. Vous pouvez en admirer quelques exemples sur la [page wikipedia](#)



Dans ce TP, on se propose de réaliser quelques **figures géométriques** en Art-ASCII à l'aide de **boucles for**.

Attention : La plupart du temps, il faudra faire des **boucles imbriquées**. Cela peut vite devenir complexe, il est donc important de procéder par une succession de “essais/erreurs” pour écrire du code qui fonctionne.

1.1 La fonction print

La fonction **print** va naturellement à la ligne après avoir fait son affichage. Ceci est dû à la valeur **par défaut** du paramètre **end** (Vous pouvez consulter la documentation de la fonction **print** grâce à la cellule ci-dessous)

```
[ ]: help(print)
```

Ce comportement par défaut est problématique pour faire de l'Art-ASCII. Heureusement on peut empêcher ce retour à la ligne en donnant une chaîne de caractère vide "" au paramètre `end`. Voir Exemples ci-dessous

```
[1]: # Comportement par défaut de la fonction print
print("O")
print("X")
print("O")
```

```
O
X
O
```

```
[2]: # Comportement sans retour à la ligne
print("O", end="")
print("X", end="")
print("O", end="")
```

```
OXO
```

Enfin, pour faire de l'Art-ASCII, il est utile de faire des retours à la ligne "sans rien afficher". Ceci peut se faire simplement en appelant la fonction `print` sans lui donner de paramètre. Voir exemples ci-dessous :

```
[3]: print("O", end="")
print("X", end="")
print("O", end="")
```

```
OXO
```

```
[4]: print("O", end="")
print("X", end="")
print() # On ajoute un retour à la ligne
print("O", end="")
```

```
OX
O
```

1.2 Travail demandé

- Pour chaque fonction à écrire, deux exemples d'appels sont systématiquement donnés pour comprendre ce qui est demandé
- Les fonctions sont de plus en plus complexes :
- La question 1 est de niveau facile
- Les questions 2 à 4 sont de niveau intermédiaire
- Les question 5 et 6 sont de niveau difficile

- La question 7 est de niveau expert !
- **Attention : Dans tout le TP, l'usage de l'opérateur * sur des chaînes de caractères est INTERDIT !!**

1. Ecrire une fonction `afficher_ligne(n)` qui affiche une ligne de `n` caractères '0'

```
[6]: afficher_ligne(5)
```

```
00000
```

```
[7]: afficher_ligne(12)
```

```
000000000000
```

2. Ecrire une fonction `afficher_carre(n)` qui affiche un carré de côté `n` rempli de caractères '0'

Attention : il NE FAUT PAS utiliser la fonction `afficher_ligne` pour écrire le code de la fonction `afficher_carre` !!

```
[11]: afficher_carre(5)
```

```
00000
00000
00000
00000
00000
```

```
[12]: afficher_carre(12)
```

```
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
```

3. Ecrire une fonction `afficher_rectangle(hauteur, largeur)` qui affiche un rectangle rempli de caractères '0'

Attention : il NE FAUT PAS utiliser la fonction `afficher_ligne` pour écrire le code de la fonction `afficher_rectangle` !!

```
[14]: afficher_rectangle(8, 3)
```

```
000
000
000
000
000
000
000
000
000
```

```
[15]: afficher_rectangle(5, 20)
```

```
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
```

4. Ecrire une fonction `afficher_triangle_rectangle(n)` qui affiche un triangle rectangle de hauteur `n` rempli de caractères '0'

Attention : il NE FAUT PAS utiliser la fonction `afficher_ligne` pour écrire le code de la fonction `afficher_triangle_rectangle` !!

```
[22]: afficher_triangle_rectangle(5)
```

```
0
00
000
0000
00000
```

```
[21]: afficher_triangle_rectangle(10)
```

```
0
00
000
0000
00000
000000
0000000
00000000
000000000
0000000000
```

5. Ecrire une fonction `afficher_carre_diagonale(n)` qui affiche un carré de hauteur `n` rempli de caractères '0' avec une diagonale de caractères 'X'

Attention : il NE FAUT PAS utiliser la fonction `afficher_ligne` pour écrire le code de la fonction `afficher_triangle_rectangle` !!

[24]: `afficher_carre_diagonale(5)`

```
X0000
0X000
00X00
000X0
0000X
```

[25]: `afficher_carre_diagonale(8)`

```
X00000000
0X0000000
00X000000
000X00000
0000X0000
00000X000
000000X00
0000000X0
00000000X
```

6. Ecrire une fonction `afficher_triangle(n)` qui affiche un triangle “creux” de hauteur **n** composé de caractères '0'

Attention : il NE FAUT PAS utiliser la fonction `afficher_ligne` pour écrire le code de la fonction `afficher_triangle` !!

[27]: `afficher_triangle(10)`

```
      0
     0 0
    0  0
   0   0
  0    0
 0     0
0      0
0     0
0    0
0   0
0  0
0 0
00000000000000000000
```

[28]: `afficher_triangle(4)`

```
      0
     0 0
    0  0
0000000
```

7. Ecrire une fonction `afficher_losange(n)` qui affiche un losange “de vide” de côté **n** dans un rectangle rempli de caractères '\$'

Attention : il NE FAUT PAS utiliser la fonction `afficher_ligne` pour écrire le code de la fonction `afficher_losange` !!

```
afficher_losange(5)
```

\$\$\$\$ \$\$\$\$
 \$\$\$ \$ \$\$\$
 \$\$ \$\$\$ \$
 \$ \$\$\$\$\$ \$
 \$\$\$\$\$\$
 \$ \$\$\$\$\$ \$
 \$\$ \$\$\$ \$
 \$\$\$\$ \$\$\$\$

```
afficher_losange(15)
```

[illegible]