

A Un carré en ASCII Art

L'« ASCII Art » consiste à utiliser les caractères alphanumériques du code ASCII (Cf. chapitre 9) pour produire des images. Nous allons commencer par dessiner un carré à l'aide de la lettre X.

La fonction `print()` prend un nombre quelconque d'arguments et les affiche en les séparant par un espace et en terminant par un retour-chariot. Il faut donc calculer le contenu d'une ligne entière avant de l'afficher. On peut utiliser des boucles :

```
ligne = ""
for i in range(5):
    ligne = ligne + "-->8--"

print(ligne) # [-->8---->8---->8---->8---->8--
```

On peut également utiliser les opérateurs de concaténation (+) et de répétition (*) des chaînes :

```
print("-->8--" * 5) # [-->8---->8---->8---->8---->8--
```

- Écrire le programme qui affiche le carré ci-dessous (taille 8), en définissant une fonction qui prend en paramètre la longueur du côté.

Remarque : on affiche "X" et " ", plutôt que "X" et "" (un espace supplémentaire). C'est purement esthétique : les caractères affichés ont une taille fixe et sont (à peu près) deux fois plus hauts que larges, notre résultat est ainsi plus proche d'un carré.

```
X X X X X X X X
X       X
X       X
X       X
X       X
X       X
X       X
X X X X X X X X
```

B D'autres formes

- Réutiliser le code du carré pour afficher les formes suivantes, représentées ici avec un `cote` de 9 (mais le code doit permettre de varier la valeur de `cote`).

Croix dans un carré

```
X X X X X X X X X
X       X
X       X
X       X
X X X X X X X X X
X       X
X       X
X       X
X X X X X X X X X
```

Diagonale descendante

```
X X X X X X X X X
X X       X
X X       X
X X       X
X X       X
X X       X
X X       X
X X       X
X X       X
```

"Origami"

```
X X X X X X X X X
X       X X
X       X X
X       X X
X       X X
X       X X
X       X X
X X X X X X X X X
```

Triangle

```
      X
     X X
    X   X
   X     X
  X       X
 X X X X X X X X X
```

Triangle inversé

```
X X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
X X X X X X X X X
```

C Le sapin de Noël

Nous allons maintenant dessiner un sapin de Noël, en quatre étapes illustrées ci-dessous. Comme pour le carré, le programme devra définir une fonction `sapin` qui prend en paramètre la taille de celui-ci.

1^{ère} étape : Afficher un triangle représentant le feuillage puis un rectangle représentant le tronc, en utilisant les barres obliques (/ et \) et verticale (|) ainsi que l'accent circonflexe pour le sommet (^).

2^e étape : Ajouter une étoile au sommet de l'arbre.

3^e étape : Modifier le programme afin d'afficher la texture du feuillage, en alternant les guillemets simples et doubles (' et ").

4^e étape : Ajouter des décorations, représentées par des 'o' disposés aléatoirement sur le sapin.

La fonction `random()` de la bibliothèque `random` retourne un nombre flottant aléatoire compris entre 0 (inclus) et 1 (exclus), suivant une distribution uniforme (tous les nombres entre 0 et 1 ont la même chance d'être tirés au sort). Ce nombre est différent *chaque fois* que l'on appelle la fonction `random()`.

En utilisant `random()`, modifier le code de l'étape 3 pour qu'une décoration "o" apparaisse 20 % du temps à la place d'un guillemet. Le sapin sera différent à chaque appel de `sapin()`.

