

Python: afficher un labyrinthe



Lorsqu'on programme, il arrive souvent qu'on doive traiter des données organisées en plusieurs dimensions. Par exemple, imaginez que vous deviez parcourir un tableau en deux dimensions (comme une grille ou un échiquier), générer une table de multiplication ou dessiner des motifs complexes comme une pyramide d'étoiles. Dans ces cas-là, une simple boucle ne suffit pas, car il faut répéter des actions sur plusieurs niveaux.

5TTR



6TTR

 Exploration

Exercice Python : Affichage d'un labyrinthe en ASCII

Objectif : Créer un programme Python qui affiche un labyrinthe dans la console en utilisant les caractères ASCII  (176) pour le sol et  (178) pour les murs. Le labyrinthe doit être représenté sous la forme d'une matrice (tableau à deux dimensions).

Consignes :

1. Le labyrinthe sera représenté par une **matrice** (ou tableau à deux dimensions) où chaque élément est soit un mur (**1**), soit un sol (**0**).
2. Le programme doit parcourir cette matrice et afficher chaque élément à l'aide des caractères ASCII suivants :
 -  pour représenter le sol (correspondant à **0** dans la matrice).
 -  pour représenter un mur (correspondant à **1** dans la matrice).


Utilisez 2 caractères ASCII pour obtenir un affiche plus "carré".

3. Afficher le labyrinthe dans la console sous la forme d'un tableau en parcourant la matrice.

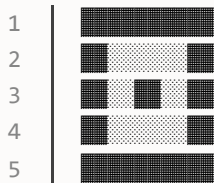
Exemple d'un labyrinthe

Vous allez d'abord définir une matrice 5x5 de base qui ressemble à ceci :

```
1 | 1 1 1 1 1
2 | 1 0 0 0 1
3 | 1 0 1 0 1
4 | 1 0 0 0 1
5 | 1 1 1 1 1
```

Le programme doit alors afficher le labyrinthe dans la console en remplaçant les **0** par  et les **1** par .

Résultat attendu :



Étapes à suivre :

1. Définir une matrice contenant des `1` pour les murs et des `0` pour les espaces.
2. Créer une boucle qui parcourt chaque élément de la matrice.
3. Remplacer les `1` par `█` et les `0` par `░` lors de l'affichage.
4. Afficher le résultat à la console.

Indications supplémentaires :

- Vous pouvez définir votre matrice comme suit en Python :

```
1 | labyrinth = [  
2 |     [1, 1, 1, 1, 1],  
3 |     [1, 0, 0, 0, 1],  
4 |     [1, 0, 1, 0, 1],  
5 |     [1, 0, 0, 0, 1],  
6 |     [1, 1, 1, 1, 1]  
7 | ]
```

- Utilisez une boucle `for` pour parcourir les lignes de la matrice et une autre boucle `for` pour parcourir chaque élément de la ligne.

Étapes suivantes :

- Modifiez le labyrinthe pour qu'il soit de dimensions différentes.
- Utilisez Rich pour afficher le labyrinthe en couleurs
- Ajoutez un nouveau type de tuile qui représente des flaques d'eau (`☪` ascii-177)

Bonne chance !