

Python: Listes à 2 Dimensions (Tableaux)

En programmation, les tableaux ou listes à deux dimensions (souvent appelés 'matrices') sont des structures de données qui permettent de stocker des informations sous forme de grille, de tableau ou de matrice. Cela est particulièrement utile lorsqu'on travaille avec des données qui ont une organisation en lignes et en colonnes, comme un tableau de chiffres, un échiquier, ou une image pixelisée.

Exploration

Dans cet article, nous allons explorer en profondeur les listes à deux dimensions en Python, comprendre leur utilité, et illustrer leur fonctionnement avec des exemples concrets. Nous aborderons également des méthodes pour manipuler ces tableaux efficacement, en passant par des boucles imbriquées, des compréhensions de liste et d'autres techniques.

Qu'est-ce qu'une liste à 2 dimensions ?

Une **liste à deux dimensions** est simplement **une liste qui contient d'autres listes** en tant qu'éléments. Cela permet de représenter une structure bidimensionnelle, où chaque sous-liste correspond à une "ligne" de la matrice.

Imaginons que nous souhaitons représenter une grille de 3x3 en Python, où chaque cellule contient un chiffre. Voici à quoi cela pourrait ressembler :

```
1 | matrice = [  
2 |     [1, 2, 3],  
3 |     [4, 5, 6],  
4 |     [7, 8, 9]  
5 | ]
```

Dans cette matrice, la première sous-liste `[1, 2, 3]` représente la première ligne, la deuxième sous-liste `[4, 5, 6]` représente la deuxième ligne, et ainsi de suite. Chaque sous-liste est donc une ligne, et chaque élément à l'intérieur de ces sous-listes représente une colonne.

Visualisation de la matrice

On peut représenter cette matrice de manière plus visuelle comme ceci :

```
1 2 3  
4 5 6  
7 8 9
```

Chaque cellule contient un nombre et a une position définie par deux indices : l'indice de la ligne et l'indice de la colonne.

Accéder aux éléments d'une liste à 2 dimensions

En Python, pour accéder à un élément spécifique dans une liste à deux dimensions, on utilise une double indexation. La première correspond à l'indice de la ligne, et la seconde à l'indice de la colonne.

Accéder à un élément

Supposons que nous voulons accéder à l'élément qui se trouve à la deuxième ligne et à la troisième colonne (c'est-à-dire le 6 dans notre matrice) :

```
1 | element = matrice[1][2] # Rappel : les indices commencent à 0
2 | print(element) # Affichera 6
```

Dans cet exemple :

- `matrice[1]` accède à la deuxième ligne : `[4, 5, 6]`
- `matrice[1][2]` accède ensuite à l'élément à l'indice 2 dans cette sous-liste, soit 6.

Quelques exercices

Créer une matrice identité

Une **matrice identité** est une matrice carrée où les éléments de la diagonale principale sont égaux à 1 et tous les autres éléments sont 0. Nous pouvons créer une matrice identité en utilisant la compréhension de liste :

Résultat :

```
[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]
```

Échiquier

Un autre exemple classique est l'affichage d'un échiquier sous forme de matrice avec des X et des 0 alternés.

Résultat :

```
X O X O X O X O
O X O X O X O X
X O X O X O X O
O X O X O X O X
X O X O X O X O
```

```
0 X 0 X 0 X 0 X
X 0 X 0 X 0 X 0
0 X 0 X 0 X 0 X
```

Pour l'exemple, un échiquier de 5x5 sera suffisant 😊.

Labyrinthe

Un labyrinthe est une structure en 2 dimensions. Il peut donc être représenté en Python sous forme d'un tableau à 2 dimensions (matrice).

Crée un petit labyrinthe de 5x5 où les murs sont représentés par des 1 et le sol par des 0.

Résultat:

```
11111
10001
10101
10001
11111
```

Manipuler une liste à 2 dimensions

Souvent, lorsqu'on travaille avec des listes à deux dimensions, on souhaite effectuer des opérations sur l'ensemble des éléments de la matrice. Pour cela, les **boucles imbriquées** sont très utiles.

Afficher tous les éléments d'une matrice

Voici un exemple simple où nous parcourons et affichons tous les éléments d'une matrice 3x3 :

```
1 | for ligne in matrice:
2 |     for element in ligne:
3 |         print(element, end=' ')
4 |     print() # Pour changer de ligne après chaque sous-liste
```

Résultat :

```
1 2 3
4 5 6
7 8 9
```

Explication :

- La première boucle parcourt chaque sous-liste (chaque ligne).
- La deuxième boucle parcourt chaque élément de la sous-liste.
- La fonction `print(element, end=' ')` permet d'afficher les éléments sur la même ligne sans retour à la ligne automatique.
- `print()` après la deuxième boucle sert à passer à la ligne suivante une fois que tous les éléments de la sous-liste ont été affichés.

Calculer la somme de tous les éléments de la matrice

Si nous voulions calculer la somme de tous les éléments d'une matrice, nous pourrions utiliser une approche similaire avec des boucles imbriquées :

```
1 | somme = 0
2 | for ligne in matrice:
3 |     for element in ligne:
4 |         somme += element
5 |
6 | print("Somme de tous les éléments :", somme)
```

Résultat :

Somme de tous les éléments : 45

Conclusion

Les listes à deux dimensions sont des structures de données fondamentales pour organiser et manipuler des données en Python, en particulier lorsqu'elles sont disposées en lignes et colonnes. Que vous souhaitiez représenter un tableau de multiplication, un échiquier, ou des données complexes, les listes