

.....

Session d'exercices – Matrices et Listes Imbriquées

Dans cet exercice, nous allons explorer les listes imbriquées à travers l'une de leurs applications les plus courantes : les matrices.

Exemple : Extraction de la Diagonale Principale

Considérons la matrice M définie ci-dessous :

$$M = \begin{bmatrix} -2 & 5 & 3 & 2 \\ 9 & -6 & 5 & 1 \\ 3 & 2 & 7 & 3 \\ -1 & 8 & -4 & 8 \end{bmatrix}$$

Notre objectif est d'extraire la diagonale principale de cette matrice et de l'afficher. La diagonale principale est constituée des éléments où les indices de ligne et de colonne sont les mêmes (c'est-à-dire, les éléments $M[0][0]$, $M[1][1]$, $M[2][2]$, $M[3][3]$). Regardons le code ci-dessous et les explications détaillées qui suivent.

```

1     # Définition d'une matrice 4x4
2     M = [[-2, 5, 3, 2],
3          [9, -6, 5, 1],
4          [3, 2, 7, 3],
5          [-1, 8, -4, 8]]
6
7     # Affichage de la matrice, formatée proprement
8     for row in M:
9         for val in row:
10            print(f"{val:4}", end="")
11            print()
12
13    # Trouver la diagonale principale
14    n = len(M)
15    diagonale_principale = [M[i][i] for i in range(n)]
16    print(f"Diagonale principale = {diagonale_principale}")
17    # Sortie: Diagonale principale = [-2, -6, 7, 8]

```

Définition de la Matrice (Lignes 2-5) : Nous définissons la matrice M comme une liste imbriquée, où chaque élément de la liste externe représente une ligne de la matrice. Chaque ligne est aussi une liste contenant les éléments de cette ligne. Puisqu'il y a des listes à l'intérieur d'une liste, on appelle cela une liste imbriquée.

Affichage de la Matrice (Lignes 8-11) : La matrice est affichée dans un format lisible. La boucle externe à la ligne 8 itère sur les lignes de la matrice, tandis que la boucle interne à la ligne 9 itère sur les éléments de chaque ligne. Le spécificateur de format `f"{val:4}"` garantit que chaque valeur `val` est affichée avec un alignement à droite dans un champ de quatre caractères, rendant l'affichage propre et aligné.

Extraction de la Diagonale Principale (Lignes 14-15) : À la ligne 14, nous calculons la taille de la matrice (puisque'elle est carrée, le nombre de lignes est égal au nombre de colonnes). La ligne 15 utilise une compréhension de liste pour extraire les éléments le long de la diagonale principale. L'indice `i` est utilisé pour sélectionner le `i`-ème élément de la `i`-ème ligne, correspondant aux positions diagonales.

.....

Affichage de la Diagonale (Ligne 16) : La diagonale principale est stockée dans la liste appelée `diagonale_principale`, et nous l'affichons pour montrer le résultat.

Résolvez les problèmes d'exercices suivants. *Avant de coder, élaborer l'algorithme et testez-le sur papier pour vérifier qu'il fonctionne.*

1. [Difficulté : *] **Vérifiez si la Matrice est Carrée.** Ajoutez une vérification pour vous assurer que la matrice est carrée (c'est-à-dire que le nombre de lignes est égal au nombre de colonnes). Si ce n'est pas le cas, affichez un message d'erreur de votre choix.
2. [Difficulté : *] **Extraction de l'Anti-Diagonale.** Écrivez un nouveau bloc de code pour extraire l'anti-diagonale de la matrice, c'est-à-dire les éléments où la somme des indices de ligne et de colonne est égale à $n - 1$. Pour la matrice donnée, l'anti-diagonale contient les éléments `M[0][3]`, `M[1][2]`, etc.

```
M = [[-2, 5, 3, 2], [9, -6, 5, 1], [3, 2, 7, 3], [-1, 8, -4, 8]]
# Sortie: Anti-diagonale = [2, 5, 2, -1]
```

3. [Difficulté : **] **Aplatir une Liste Imbriquée.** Écrivez un bloc de code qui aplatit une liste imbriquée (matrice) en une seule liste.

```
matrix = [[1, 2, 3], [4, 5], [6, 7, 8, 9]]
# Sortie: Liste aplatie = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

4. [Difficulté : **] **Compter les Occurrences d'un Élément Spécifique dans une Liste Imbriquée.** Écrivez un programme qui compte combien de fois un élément spécifique apparaît dans une liste imbriquée.

```
matrix = [[1, 2, 3], [2, 4, 2], [5, 2, 6]]
target = 2
# Sortie: L'element 2 apparait 4 fois.
```

5. [Difficulté : **] **Multiplier les Éléments de Deux Matrices.** Écrivez un programme qui multiplie deux matrices de dimensions égales, élément par élément.

```
matrix1 = [[1, 2, 3], [4, 5, 6]]
matrix2 = [[7, 8, 9], [10, 11, 12]]
# Sortie: Produit élément par élément = [[7, 16, 27], [40, 55, 72]]
```

Pour créer une nouvelle matrice avec les mêmes dimensions que `matrix1` et initialiser tous ses éléments à zéro, vous pouvez utiliser le code ci-dessous. Essayez de comprendre comment fonctionne ce morceau de code.

```
# Créer la matrice de resultat et initialiser ses elements a zero
result = [[0 for j in range(len(matrix1[0]))] # Pour chaque colonne
          for i in range(len(matrix1))] # Pour chaque ligne
```

6. [Difficulté : ***] **Transposer une Matrice.** Étant donné une matrice, écrivez un programme qui transpose la matrice (échange les lignes et les colonnes).

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
# Sortie: Matrice transposée = [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```
