

Les listes à deux dimensions

(liste de listes ou matrice)

Liste à une dimension

- Les listes que nous avons utilisé jusqu'à présent sont des listes de cette forme:

0	1	2	3	4	5
12	5	-8	77	-6	10

Cette liste est composé d'une seule ligne et plusieurs colonnes

Définition

- Une **liste de liste** s'apparente à un tableau à deux dimensions (en plusieurs lignes et plusieurs colonnes) appelé **matrice**.
- Exemple:

12	7	-1
9	16	40
5	1	3

Matrice : 3 X 3

- Pour déclarer cette liste de listes qui contient 3 lignes et 3 colonnes et contenant ces valeurs, on écrit:

Maliste= [[12,7,-1], [9,16,40], [5,1,3]]

Accès aux lignes d'une matrice

- On peut accéder à une ligne d'une matrice, à l'aide de son **indice** associée.
- Exemple:

	0	1	2
0	12	7	-1
1	9	16	40
2	5	1	3

Maliste[1] = [9, 16, 40]

Accès aux lignes d'une matrice

- Exemples:

```
Maliste=[ [12,7,-1] , [9,16,40] , [5,1,3] ]
```

```
print(Maliste[0])          # [12, 7, -1]
```

```
print(Maliste[1])          # [9, 16, 40]
```

```
print(Maliste[2])          # [5, 1, 3]
```

Accéder à une valeur dans une matrice

- Pour accéder ou référencer une valeur dans une matrice, on se sert des indices.
- Syntaxe:

MaListe[indiceLigne][indiceColonne]

- Exemple: **MaListe**[2][1] fait référence à la valeur qui se trouve dans la ligne d'indice 2 et la colonne d'indice 1

	0	1	2
0	12	7	-1
1	9	16	40
2	5	1	3

Diagram illustrating matrix access:

- MaListe**[0][1] points to the value 7 (Row 0, Column 1).
- MaListe**[2][1] points to the value 1 (Row 2, Column 1).

Accéder à une valeur de la matrice

- Exemples:

```
Maliste=[ [12,7,-1] , [9,16,40] , [5,1,3] ]

print(Maliste[0][2])           # -1

print(Maliste[1][1])           # 16

print(Maliste[2][0])           # 5

#modifier la valeur d'un élément d'une liste de liste
Maliste[2][0]=80

print(Maliste)   #Maliste=[ [12,7,-1] , [9,16,40] , [80,1,3] ]
```

Parcourir une matrice

- On utilise pour cela un premier indice **i** par exemple qui va parcourir les lignes et un deuxième indice **j** par exemple qui va parcourir les colonnes.

```
Maliste=[ [12,7,-1] , [9,16,40] , [5,1,3] ]
```

```
for i in range(3):           #parcourir les lignes : 3 lignes ici
    for j in range(3):       #parcourir les colonnes: 3 colonnes
        print(Maliste[i][j],end=" ")
```

```
print()
```

```
#le résultat:
```

```
12  7  -1
 9  16 40
 5   1  3
```


Parcourir une matrice

- Si on a pas besoin des indices on peut parcourir la matrice de cette manière :

```
Maliste=[ [12,7,-1] , [9,16,40] , [5,1,3] ]
```

```
for x in M:  
    for y in range(3):  
        print(y,end=" ")
```

```
print()
```

```
#le résultat:
```

```
12  7  -1  
9   16 40  
5   1   3
```

Remplir les valeurs de la matrice

- **Exercice** : Ecrire un programme qui permet de remplir une liste de liste composée de 4 lignes et 2 colonnes par des **zéros**.
- Notre matrice doit être comme suit:

M

o	o
o	o
o	o
o	o

Remplir les valeurs de la matrice

- On va créer une liste qui va être remplie par les valeurs de la première ligne puis on va l'ajouter à la matrice:

```
M=[]  
Ligne=[]  
  
for j in range(2):  
    Ligne.append(o)  
  
M.append(Ligne)
```

- On va répéter cette action pour toutes les lignes, c'est-à-dire 4 fois pour obtenir la solution complète suivante:

Remplir les valeurs de la matrice

- Solution de l'exercice:

```
M=[]  
  
for i in range(4):  
    Ligne=[]  
    for j in range(2):  
        Ligne.append(0)  
  
M.append(Ligne)
```

Utilisation de la fonction len()

- L'utilisation de la fonction **len()** pour une matrice est comme suit:

```
M=[ [7,-1] , [16,4] , [5,1] , [24,65] ]
```

```
X=len(M)      #x=4   c'est le nombre de lignes
```

```
A=len(M[1])    #A = 2
```

```
B=len(M[1])    #B=2
```

```
C=len(M[2])    #C=2
```

```
D=len(M[3])    #d=2
```