

Nom :	Prénom :	Classe :
-------	----------	----------

NSI 1re — Matrice

Objectifs :

- Savoir identifier une matrice implémentée en Python.
- Connaître quelques usages des matrices.
- Savoir construire une matrice en Python. Savoir accéder ou modifier les éléments de cette matrice.

Introduction

Les matrices¹ sont des **tableaux** de nombres organisés en lignes et en colonnes ; en Python, on les représente souvent par une **liste de listes**, et en NSI au lycée elles servent à manipuler des données de façon structurée, un peu comme un tableau de notes, un planning ou une grille de jeu.

L'analogie la plus simple, c'est un tableau à double entrée, comme un emploi du temps ou une grille de résultats, où chaque case a une place précise repérée par sa **ligne** et sa **colonne**. Les matrices servent beaucoup à représenter des images, des réseaux, des calculs scientifiques, et surtout à résoudre des systèmes d'équations ou à automatiser des traitements de données en informatique.

Exemples de matrices

Des matrices peuvent être utilisées pour représenter toutes sortes de concepts :

Exemple de matrice : la liste d'élèves d'un lycée

Nom	Prénom	Classe	Proposition d'implémentation en Python
Martin	Emma	2°12	<pre>li = [['Martin', 'Emma', '2°12'], ['Petit', 'Lucas', '1°5'], ['Garcia', 'Jade', '1°8'], ['Leroy', 'Louise', '2°12'], ['Girard', 'Ahmed', 'T°2'],]</pre>
Petit	Lucas	1°5	
Garcia	Jade	1°8	
Leroy	Louise	2°12	
Girard	Ahmed	T°2	

¹ Pour l'histoire, il n'y a pas une seule autrice ou un seul auteur à l'origine des matrices : l'idée s'est construite progressivement avec plusieurs mathématiciens, notamment autour des systèmes d'équations, avec des noms comme Gauss, Cramer, Leibniz ou Laplace.

Exemple de matrice : un planning

Date	Évènement	Proposition d'implémentation en Python
01/05/2030	Teknival	<pre>tab = [["01/05/2030", "Teknival"], ["20/06/2030", "Festival Hellfest"], ["30/06/2030", "Repas avec tante Chloé"], ["04/08/2030", "Camping Aveyron"],]</pre>
20/06/2030	Festival Hellfest	
30/06/2030	Repas avec tante Chloé	
04/08/2030	Camping Aveyron	

Exemple de matrice : un petit jeu

	X	X
	O	
		O

```
board = [  
    ['', 'x', 'x'],  
    ['', 'o', ''],  
    ['', '', 'o'],  
]
```

Exemple de matrice : la carte d'un jeu vidéo

Proposition d'implémentation en Python

```
m = [
    ['house', tree, '', tree, '', '', ''],
    ['', tree, '', '', '', '', house],
    ['', tree, wolf, '', tree, '', ''],
    ['', '', tree, crown, tree, '', ''],
    [tree, tree, '', tree, frog, tree, ''],
    [axe, tree, '', tree, wolf, '', ''],
    [frog, '', tree, wolf, '', '', '']
]
```

Création d'une matrice en Python

La création d'une matrice en Python s'effectue en créant une `list` (tableau indexé), dont les éléments sont eux-mêmes des `list`.

Prenons un exemple :

```
1 a = [  
2     [29, 18, 39],  
3     [3, 5, 57],  
4 ]
```

Dans l'exemple ci-dessus, la variable `a` contient une donnée de type `list` dont chaque élément est une `list` d'entiers.



Python accepte également de mettre tout sur une même ligne ...

Mais ça n'est pas très lisible pour un être humain ! Exemple : `a = [[29, 18, 39], [3, 5, 57]]`

► Exercice 1 — Implémenter des matrices en Python

Implémenter les matrices suivantes en Python :

Matrice		
3	2	1
6	5	4
9	8	7

Implémentation en Python (à compléter)

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

Matrice		
i	h	g
f	e	d
c	b	a

Implémentation en Python (à compléter)

```
a = [  
    ['i', 'h', 'g'],  
    ['f', 'e', 'd'],  
    ['c', 'b', 'a'],  
]
```

Matrice		
True	False	False
False	True	False
False	False	True

Implémentation en Python (à compléter)

```
u = [  
    [True, False, False],  
    [False, True, False],  
    [False, False, True],  
]
```

Accès à un élément d'une matrice, en Python

L'accès à un élément d'une matrice se fait de la même manière que pour un tableau indexé (list) : grâce à des indices.

Prenons nouvel exemple de matrice :

Matrice			Implémentation en Python
k	s	p	<pre>m = [['k', 's', 'p'], ['h', 'e', 't'], ['x', 'c', 'd']]</pre>
h	e	t	
x	c	d	



Dans cette list de “sous-list” stockée dans `m`, comment accéder par exemple à l'élément « k » ? Nous voyons bien que « k » est le premier élément de la première “sous-list” ... Il y aurait 2 indices ?

1 matrice = 2 indices

Pour accéder à un élément d'une matrice nous avons besoin de deux indices :

- Un **1er indice** — généralement noté `i` — servant à déterminer **quelle “sous-list”** utiliser.
- Un **2ème indice** — généralement noté `j` — servant à déterminer **quel élément de cette “sous-list”** utiliser.

Ce qui donnerait, pour notre exemple de matrice :

	j = 0	j = 1	j = 2
i = 0	k	s	p
i = 1	h	e	t
i = 2	x	c	d

Implémentation en Python
<pre>m = [['k', 's', 'p'], ['h', 'e', 't'], ['x', 'c', 'd']]</pre>

Ainsi, pour accéder à l'élément « k », `i` vaut `0` et `j` vaut `0`.

Et dans l'implémentation Python de la matrice `m`, on accéderait à l'élément « k » grâce à `m[0][0]`.

Autre exemple, pour accéder à l'élément « x », `i` vaut `2` et `j` vaut `0`.

Et dans l'implémentation Python de la matrice `m`, on accéderait à l'élément « x » grâce à `m[2][0]`.

► Exercice 2 — Identifier les indices nécessaires

Compléter les phrases suivantes :

- Pour accéder à l'élément « d », `i` vaut et `j` vaut
- Pour accéder à l'élément « s », `i` vaut et `j` vaut

► Exercice 3 — Savoir accéder à une “list” de “list” en Python

Analyser les programmes Python ci-dessous, puis déterminer comment accéder à l'élément « a » (str).

N°	Déclaration d'une matrice	Code Python permettant d'accéder à l'élément « a »
0	<pre>m = [['f', 'd', 'b'], ['h', 'i', 'a'], ['t', 't', 'c']]</pre>	<code>m[1][2]</code>
1	<pre>m = [[0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 'a', 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0]]</pre>	<pre>m[3][7] # ou m[3][-2]</pre>
2	<pre>x = [[False, True, False, True], [False, True, False, True], [False, True, False, True], [False, 'a', False, True], [False, True, False, True],]</pre>	<pre>x[3][1] # ou x[-2][-3]</pre>
3	<pre>u = [['b', 'h', 'g', 't'], ['c', 'j', 'f', 'z'], ['d', 'k', 'e', 'a']]</pre>	<pre>u[2][3] # ou u[-1][-1]</pre>
4	<pre>k = [[0], [0], ['a'], [0]]</pre>	<code>k[2][0]</code>
5	<code>m = [[0, 0], [0, 'a'], [0, 0]]</code>	<code>m[1][1]</code>

Modifier un élément d'une matrice, en Python

La modification d'un élément d'une matrice nécessite de connaître les indices permettant d'accéder à cet élément.

Reprenons la matrice suivante :

	j = 0	j = 1	j = 2
i = 0	k	s	p
i = 1	h	e	t
i = 2	x	c	d

Implémentation en Python

```
m = [  
    ['k', 's', 'p'],  
    ['h', 'e', 't'],  
    ['x', 'c', 'd']  
]
```

Si nous voulons par exemple remplacer l'élément de valeur « s » par la valeur « a », il nous faut d'abord connaître les indices *i* et *j* de cet élément « s »...

Dans notre matrice, l'élément « s » a pour indice *i* `0` et pour indice *j* `1`.

Dans l'implémentation Python proposée, il est donc possible d'y accéder grâce à `m[0][1]` :

```
1 >>> print( m[0][1] )    # Affiche la valeur de l'élément d'indices 0 1  
2 s
```

Et pour **modifier la valeur** de cet élément en `m[0][1]`, il suffit d'utiliser le symbole `=`.

Par exemple, pour donner la valeur « a » (str) à l'élément en `m[0][1]`, il faut faire :

```
1 m[0][1] = 'a'
```

Notre matrice a été modifiée !

Essayons de l'afficher pour voir à quoi elle ressemble :

```
1 >>> print( m )  
2 [['k', 'a', 'p'], ['h', 'e', 't'], ['x', 'c', 'd']]
```

Cette matrice correspond à l'illustration suivante :

k	a	p
h	e	t
x	c	d

Voilà comment nous avons réussi à modifier la valeur d'un élément d'une matrice.

► Exercice 4 — Modifier les éléments d'une matrice

Dans les propositions qui suivent, analyser la matrice de gauche puis la matrice de droite, et rédiger le code Python permettant de passer de l'une à l'autre.

N°	Matrice Python “avant”	Matrice Python “après”	Code Python permettant cette modification
0	<pre>m = [[0, 3, 8], [0, 3, 8], [0, 3, 8]]</pre>	<pre>m = [[0, 3, 8], [0, 6, 8], [0, 3, 8]]</pre>	<pre>m[1][1] = 6</pre>
1	<pre>a = [[0, 1, 7, 4], [9, 8, 6, 4], [4, 7, 5, 4]]</pre>	<pre>a = [[0, 1, 7, 4], [9, 8, 6, 4], [4, 7, 5, 2]]</pre>	<pre>a[2][3] = 2</pre>
2	<pre>b = [[0, 3, 8], [0, 3, 8], [0, 3, 8]]</pre>	<pre>b = [[1, 3, 8], [0, 3, 8], [0, 3, 7]]</pre>	<pre>b[0][0] = 1 b[2][2] = 7</pre>
3	<pre>c = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]</pre>	<pre>c = [[0, 0, 1], [0, 1, 0], [1, 0, 0]]</pre>	<pre>c[0][2] = 1 c[1][1] = 1 c[2][0] = 1</pre>
4	<pre>d = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]</pre>	<pre>d = [[0, 0, 1], [0, 0, 0], [0, 0, 1]]</pre>	<pre>d[0][2] = 1 d[2][2] = 1</pre>
5	<pre>e = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]</pre>	<pre>e = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]</pre>	<pre>e[0][0] = 1 e[0][1] = 1 e[0][2] = 1 e[1][0] = 1 # ... for i in range(len(e)): for j in range(len(e[i])): e[i][j] = 1</pre>

► Exercice 5 — Concevoir et modifier une matrice

Dans le jeu qui suit, la joueuse est représentée par l'emoji « 😊 », les obstacles sont représentés par « 🌳 » et le but à atteindre est représenté par « 🍎 ».

Voici la carte du jeu :

😊	🌳	🍎
	🌳	

a) Proposer une implémentation Python de cette matrice notée `m`, en prenant en considération les transcriptions suivantes :

- Une case vide a comme valeur `0`.
- Un obstacle a comme valeur `1`.
- La joueuse a comme valeur `8`.
- Le but à atteindre a comme valeur `9`.

```
m = [  
    [8, 1, 9],  
    [0, 1, 0],  
    [0, 0, 0]  
]
```

b) Écrire les modifications successives de `m`, de manière à déplacer la joueuse de case en case jusqu'au but à atteindre, en évitant les obstacles.

```
m[1][0] = 8  
m[2][0] = 8  
m[2][1] = 8  
m[2][2] = 8  
m[1][2] = 8  
m[0][2] = 8
```