– Python3 –

‘tableaux 1D-2D, fonctions, lecture et écriture de fichiers’

Tableaux 1D :

Un tableau à une dimension est une suite d’emplacements qui peuvent contenir une donnée. Ils sont du type *List* en python. Un tableau peut contenir beaucoup de type de données différents : entiers, flotant, caractères, chaines de caractères, …

L’exemple ci-dessous est donné avec des entiers.

Exemple :

Valeurs	3	4	13	-3	5	6	22	8

Note : La case *Valeurs*, en gris, n’apparait pas dans un tableau écrit en python.

Chaque case du tableau est accessible séparément ou en sous tableau grace aux indices.

Exemple :

Valeurs	3	4	13	-3	5	6	22	8
Indices	**0**	1	2	3	4	5	6	7

Note: Les cases grisées n’apparaissent pas dans un tableau écrit en python.

L’indice minimal est 0.

Syntaxe Python des tableaux 1D en python :

Création du tableau de l’exemple précédent :

```
mon_tab = [3, 4, 13, -3, 5, 6, 22, 8]
```

Accès à la case contenant 3 :

```
mon_tab[0]
```

Accès à la case contenant 5 :

```
mon_tab[4]
```

Accès aux 2 premières valeurs :

```
mon_tab[0:2]
```

mon_tab[:2]

Accès aux 3 dernières valeurs :

mon_tab[5:]

Indice de la valeur 3:

mon_tab.index(3)

Ajout de 3 à la fin du tableau :

mon_tab = mon_tab + [3]

mon_tab.append(3)

Ajout de 4 au début du tableau :

mon_tab.insert(0, 4)

Ajout de 123 entre -3 et 5

mon_tab.insert(5, 123)

Modification l'entier de l'indice 5 (123) par -12 :

mon_tab[5] = -12

Inversion des éléments du tableau :

mon_tab.reverse()

Copier le tableau dans une autre variable *mon_tab2* :

mon_tab2 = mon_tab[:]

mon_tab2 = mon_tab.copy()

```
>>> mon_tab = [3, 4, 13, -3, 5, 6, 22, 8]
>>> mon_tab
[3, 4, 13, -3, 5, 6, 22, 8]
>>> mon_tab[0]
3
>>> mon_tab[4]
5
>>> mon_tab[0:2]
[3, 4]
>>> mon_tab[:2]
[3, 4]
>>> mon_tab[5:]
[6, 22, 8]
>>> mon_tab.index(3)
0
>>> mon_tab = mon_tab + [3]
>>> mon_tab
[3, 4, 13, -3, 5, 6, 22, 8, 3]
>>> mon_tab.append(3)
>>> mon_tab
[3, 4, 13, -3, 5, 6, 22, 8, 3, 3]
>>> mon_tab.insert(0,4)
>>> mon_tab
[4, 3, 4, 13, -3, 5, 6, 22, 8, 3, 3]
>>> mon_tab.insert(5, 123)
>>> mon_tab
[4, 3, 4, 13, -3, 123, 5, 6, 22, 8, 3, 3]
>>> mon_tab.reverse()
>>> mon_tab
[3, 3, 8, 22, 6, 5, 123, -3, 13, 4, 3, 4]
>>>
```

Figure 1: Capture de l'exécution de l'exemple d'un tableau à 1 dimension

Pour plus d'informations sur les opérations possibles avec les tableaux :

https://docs.python.org/fr/3/tutorial/datastructures.html

Ou rechercher sur un navigateur: *python3 list fr*

Pour les exercices suivants, faire un fichier par exercice et ajouter en commentaires le numéro des questions aux bons endroits.

Exercice tableau 1 : (tout doit être réalisé à l'aide de Python)

1. Faire un tableau contenant les valeurs suivantes :

   146, 16, 46, -147, 36, -83, 150, -100, -26 et 122.

2. Trouve la valeur maximale du tableau.
3. Trouver la valeur minimale du tableau.
4. Trouver l'écart maximal entre deux valeurs successives.
5. Stocker dans un second tableau la valeur absolue de toutes les valeurs du tableau de la question 1.
6. [Extra] Convertir en binaire toutes les valeurs du tableau de la question 5.

Comme écrit plus haut, les tableaux peuvent contenir tous types de valeurs. Nous allons maintenant voir des tableaux contenant des chaînes de caractères.

Exercice tableau 2 : (tout doit être réalisé à l'aide de Python)

1. Dans un tableau, stocker tous les prénoms des élèves de la classe.
2. Trouver le prénom le plus long.
3. Dans un second tableau, stocker les prénoms par ordre alphabétique.
4. Dans un troisième tableau, stocker les prénoms, dans l'ordre croissant, par nombre de caractères.
5. Dans un quatrième tableau, inverser l'ordre des lettres de tous les prénoms.
6. [Extra] Existe-t-il des prénoms qui sont aux mêmes indices dans le premier (question 1) et second (question 3) tableau ? Même question pour le second et le troisième tableau (question 4) .

Tableau 2D :

Les tableaux à deux dimensions sont des tableaux à une dimension dans lesquels chaques cases contient un tableau. En effet, un tableau peut contenir une variable de type List.

Exemple :

Indices	0		1		2		3		4		5		6	
Valeurs	Val.	Ind.	Val.	Ind.	Val.	Ind.	Val.	Ind.	Val.	Ind.	Val.	Ind.	Val.	Ind.
	14	0	20	0	-4	0	0	0	9	0	-55	0	-9	0
	13	1	21	1	-3	1	1	1	99	1	18	1	66	1
	12	2	22	2	-2	2	2	2	999	2	32	2	77	2
	11	3	23	3	-1	3	3	3	998	3	8	3	-88	3

Note : Les cases grisées n'apparaissent pas dans un tableau écrit en python.

C’est un tableau de dimension 7 par 4.

Les sous tableaux ne sont pas obligatoirement de même tailles.

Syntaxe Python des tableaux 2D en python :

Création du tableau de l’exemple précédent :

ma_mat = [[14, 13, 12, 11], [20, 21, 22, 23], [-4, -3, -2, -1], [0, 1, 2, 3], [9, 99, 999, 998], [-55, 18, 32, 8], [-9, 66, 77, -88]]

Accès à la case contenant 13 :

ma_mat[0][1]

Note : le premier chiffre entre crochets désigne l’indice du tableau principal, le second celui du sous tableau contenu dans la case du tableau principal.

Accès à la case contenant -88 :

ma_mat[6][3]

Accès *aux* valeurs de la première case :

ma_mat[0]

ma_mat[0][:]

Ajout de 4 à la fin du premier sous tableau :

ma_mat[0].append(4)

Ajout de -54 au début du tableau :

ma_mat[0].insert(0, -54)

Modification l’entier 21 (indice [0][1]) par -12 :

ma_mat[1][1] = -12

```
>>> ma_mat = [[14, 13, 12, 11], [20, 21, 22, 23], [-4, -3, -2, -1], [0,
 1, 2, 3], [9, 99, 999, 998], [-55, 18, 32, 8], [-9, 66, 77, -88]]
>>> ma_mat
[[14, 13, 12, 11], [20, 21, 22, 23], [-4, -3, -2, -1], [0, 1, 2, 3], [9
, 99, 999, 998], [-55, 18, 32, 8], [-9, 66, 77, -88]]
>>> ma_mat[0][1]
13
>>> ma_mat[6][3]
-88
>>> ma_mat[0]
[14, 13, 12, 11]
>>> ma_mat[0][:]
[14, 13, 12, 11]
>>> ma_mat[0].append(4)
>>> ma_mat
[[14, 13, 12, 11, 4], [20, 21, 22, 23], [-4, -3, -2, -1], [0, 1, 2, 3],
 [9, 99, 999, 998], [-55, 18, 32, 8], [-9, 66, 77, -88]]
>>> ma_mat[0].insert(0, -54)
>>> ma_mat
[[-54, 14, 13, 12, 11, 4], [20, 21, 22, 23], [-4, -3, -2, -1], [0, 1, 2
, 3], [9, 99, 999, 998], [-55, 18, 32, 8], [-9, 66, 77, -88]]
>>> ma_mat[1][1] = -12
>>> ma_mat
[[-54, 14, 13, 12, 11, 4], [20, -12, 22, 23], [-4, -3, -2, -1], [0, 1,
2, 3], [9, 99, 999, 998], [-55, 18, 32, 8], [-9, 66, 77, -88]]
>>>
```

Figure 2: Capture de l'exécution de l'exemple d’un tableau à 2 dimensions

Exercice tableau 3 : (tout doit être fait en Python)

1. Faire un tableau de 10 cases par 5 contenant les résultats 5 premiers résultats des tables de multiplication de 1 à 11.

1	2	3	…	10
2	4	6	…	20
3	6	9	…	30

4	8	12	…	40
5	10	15	…	50

2. Afficher les résultats « proprement » (saut de ligne toutes les lignes, une tabulation entre chaque colonnes.
3. En utilisant la fonction randint(valeur_min,valeur_max), de la librairie random, remplissez un second tableau de 20 par 18 de valeurs comprises entre -22 et 55.
4. Trouver les valeurs minimale et maximale.
5. Remplissez un second tableau à une dimension contenant tous les nombres pairs et un troisième tableau contenant celle divisible par 4.

Pour les deux parties suivantes, il faut

Fonctions :

https://www.codingame.com/playgrounds/17176/recueil-dexercices-pour-apprendre-python-au-lycee/cours---introduction-aux-fonctions

Ou chercher sur un navigateur : codingame cours introduction au fonctions

Exercice fonction :

Refaire à l'aide de fonctions :

1. Les questions 2, 3, 4, et 5 de l'exercice tableau 1.
2. Les questions 2 à 6 de l'exercice tableau 2.
3. Toutes les questions de l'exercice tableau 3

Note : Vous pouvez faire du copier-coller ; pensez à réutiliser les fonctions si possible.

Lecture et écriture de fichier texte :

https://python.sdv.univ-paris-diderot.fr/07_fichiers/