

1 Découverte

Un tableau indexé est une structure de données linéaires qui peut contenir des éléments (ou items) de types quelconques (variables, chaînes de caractères, tableaux indexés,...) placés les uns derrière les autres (c'est à dire de façon séquentielle) et repérés par un indice, en commençant par l'indice 0. En Python, l'implémentation de ces tableaux est de **type list**, il arrive souvent qu'on les appelle ainsi (liste) alors qu'il faudrait les appeler tableaux indexés.

1. Définir un tableau :

```
1 # Dans la console PYTHON
2 >>>MonTab = ["beans", "spams", "eggs"]
3
4 >>>type(MonTab)
5
```

2. On peut accéder aux éléments du tableau via l'index (ou indice) des éléments (d'où le nom tableau indexé) :

```
1
2 >>>MonTab[0]
3
4 >>>MonTab[1]
5
6 >>>MonTab[2]
7
8 >>>MonTab[3]
9
```

3. Je me teste :

- Quel est l'indice du dernier élément d'un tableau de taille 3?
- Quel est l'indice du dernier élément d'un tableau de taille 10?
- Quel est l'indice du dernier élément d'un tableau de taille n?

4. l'instruction **len** renvoie le nombre d'éléments (ou items) du tableau qu'on appelle sa taille :

```
1
2 >>>len(MonTab)
3
```

5. Voici plusieurs façons d'ajouter un élément dans un tableau

```
1 >>>MonTab.append("bacon")
2
3 >>>MonTab
4
5
```

6. Supprimer un élément dans un tableau

```
1 >>>MonTab.pop()
2
3 >>>MonTab
4
5 >>>MonTab.pop(0)
6
7 >>>MonTab
8
```

7. — L.pop() permet de

— L.pop(0) permet de

— L.pop(1) permet de

8. Parcourir les éléments d'un tableau

```
1 for elt in MonTab:
2     print(elt)
3
4 for i in range(len(MonTab)-1):
5     print(MonTab[i])
```

2 Exercices

Exercice n° 1 Compléter les encadrés python ci-dessous : _____

1. Créer un tableau nommé **planetes** contenant les mots "Mercure", "Venus", "Terre", "Mars", "Jupiter", "Saturne", "Uranus".

```
planetes=[
```

2. Afficher le 2^{ème} élément de **planetes**

```
print(
```

3. Afficher le dernier élément de **planetes**

```
print(
```

4. Afficher les 3 derniers éléments de **planetes**.

```
print(
```

5. Ajouter "Neptune" et "Pluton" à **planetes**

```
planetes
```

6. Enlever "Pluton" du tableau **planetes**

```
.
```

Exercice n° 2 Pour chaque colonne, écrire le résultat affiché par : _____

```
L=[12,7,3,59]
L.pop()
L.pop()
L.append(10)
L.append(1)
L.pop(0)
L.pop(0)
L.pop(1)
print(L)
```

```
Tab=[1,2,3]
Tab[0]=1
Tab[1]=0
print(Tab)
```

```
TabBis=[0,0,0,4,5,6]
TabBis[4]=0
TabBis[0]=TabBis[0]+1
TabBis.pop()
print(TabBis)
```

Exercice n° 3 Pour chaque colonne, écrire le résultat affiché par : _____

```
L2=[]
for i in range(4):
    L2.append(3*i)
print(L2)
```

```
L1=[]
for i in range(10):
    if i%2 ==0:
        L1.append(i)
print(L1)
```

```
L3=[1,2,3,4,5,6,7,8,9]
for i in range(5):
    L3.pop()
print(L3)
```

Exercice n° 4 Pour chaque colonne, écrire le résultat affiché par : _____

```
L=[0,0,1,0,1,1,0,0,1]
compteur=0
for elt in L:
    if elt!=0:
        compteur=compteur+1
print(compteur)
```

```
L=[1,2,3,4,4,5]
for i in range(len(L)-1):
    if L[i]==4:
        L[i]=5
print(L)
```

```
L=[1,2,3,4,4,5]
for i in range(len(L)-1):
    L[i]=L[i]+1
print(L)
```

Exercice n° 5 _____

On considère le tableau $L=[0,1,1,1,1,0,1,1,1,1,0,0,0,0,1,1,0,1,0,1,0,0,1,1,0,0,0,0,0,0,1,1,1]$

1. Écrire un algorithme en Python qui renvoie le nombre de 0 dans le tableau.

Exercice n° 6 Pour chaque colonne, écrire le résultat affiché par : _____

```
Tab=[[1,2,3],[4,5,6]]
print(Tab[0])
print(Tab[0][1])
print(Tab[1][0])
```

```
TabBis=[[0,0,0],[4,5,6]]
TabBis[0]=[1,2,3]
TabBis[1][2]=0
TabBis[1][0]=0
print(TabBis)
```

Exercice n°7

On considère un tableau de notes contenant les notes de NSI d'un élève.

1. Quelle commande permet de connaître le nombre de note dans le tableau ?
2. Quelles commandes permet d'ajouter un 20 puis un 12 à la liste NSI ?
3. Le professeur s'est trompé sur la deuxième note du tableau. Quelle commande permet d'ajouter un 1 point.
4. Quelle commande permet d'afficher la première note obtenue.
5. Quelle commande permet d'afficher la dernière note obtenue.
6. Quelle commande permet de supprimer la dernière note du tableau ?
7. Ecrire une fonction qui permet de renvoyer la note minimal de l'élève.
8. Ecrire une fonction qui permet de renvoyer la note maximale de l'élève.
9. Ecrire une fonction moyenne qui permet de calculer la moyenne des notes.
10. Reprendre l'ensemble des questions avec le tableau $NSI=[15,8,17]$.

Exercice n°8

A l'aide du fichier : prenom.py

1. Combien y a t-il de prénoms dans le tableau ?
2. A quel indice ce trouve le prénom "Charlie" ?
3. Écrire une fonction **estPresent** qui teste si un prénom est présent dans la liste. (La réponse doit être un booléen)
4. Tester votre fonction avec votre prénom.
5. Certains prénoms sont présents plusieurs fois dans le tableau. Écrire un algorithme pour les identifier.

Exercice n°9

On considère un tableau de notes contenant les notes de NSI d'un élève avec le coefficient. Par exemple, un 13 coefficient 2 puis un 15 coefficient 3 est modélisé par le tableau :

$$NSI = [[13, 2], [15, 3]]$$

1. Quelle commande permet de connaître le nombre de notes dans le tableau ?
2. Quelles commandes permet d'ajouter un 20 coefficient 2 puis un 12 coefficient 5 à la liste NSI ?
3. Le professeur s'est trompé sur la deuxième note du tableau. Quelle commande permet d'ajouter un 1 point à la note et d'ajouter 2 au coefficient ?
4. Quelle commande permet d'afficher la première note obtenue.
5. Quelle commande permet d'afficher la dernière note obtenue.
6. Quelle commande permet de supprimer la dernière note du tableau ?
7. Ecrire une fonction qui permet de renvoyer la note minimal de l'élève.
8. Ecrire une fonction qui permet de renvoyer la note maximale de l'élève.
9. Ecrire une fonction moyenne qui permet de calculer la moyenne des notes.
10. Le professeur s'est trompé sur les coefficient, il faut tous les diminuer de 1. Écrire un algorithme qui permet de corriger cela.
11. Recalculer la moyenne de l'élève.