

## 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  >>>type(MonTab)
4  .... #A compléter

```

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

```

1  >>>MonTab[0]
2  .... #A compléter
3  >>>MonTab[1]
4  .... #A compléter
5  >>>MonTab[2]
6  .... #A compléter
7  >>>MonTab[3]
8  .... #A compléter

```

### 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  >>>len(MonTab)
2  .... #A compléter

```

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

```

1  >>>MonTab.append("bacon")
2  >>>MonTab
3  .... #A compléter
4  >>>MonTab+ ["mushrooms"]
5  >>>MonTab
6  .... #A compléter
7  >>>["ketchup"]+MonTab
8  >>>MonTab
9  .... #A compléterer

```

6. Supprimer un élément dans un tableau

```
1 >>>MonTab.pop()
2 >>>MonTab
3                                     #A compléter
4 >>>MonTab.pop(0)
5 >>>MonTab
6                                     #A compléter
```

7. Je me teste :

- L.pop() permet de .....
- L.pop(0) permet de .....
- L.pop(1) permet de .....

8. Parcourir les éléments d'un tableau

```
1 # Dans la console PYTHON
2 >>>for elt in MonTab:
3     print(elt)
4
5
6
7
8                                     #A compléter
```

## 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<sup>ème</sup> 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**

```
.
```

7. Faire afficher les éléments du tableau les uns en dessous des autres à l'aide d'une boucle.

```

.
.
.
    
```

**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** Amélioration du D.M sur le NIR \_\_\_\_\_

1. Le NIR ("numero de secu" est composé de 13 caractères), par exemple : 2810462041023.  
Le "04" correspond au mois d'avril.

Compléter le code ci-dessous qui à partir d'un NIR (str donné en paramètre) va renvoyer une chaîne de caractère de type : "Vous êtes né(e) en Avril".

```

1 def InfoMois(NIR):
2     mois=NIR[...]+NIR[...] # A completer
3     mois=int(mois) #Convertir la variable mois en entier
4     L=[None,"Janvier","Février","Mars","Avril","Mai",
5        "Juin","Juillet","Aout","Septembre","Octobre","Novembre","Décembre"]
6     return "Vous etes ne(e) en " + ..... # A completer
    
```

**Exercice n° 4** 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):
    L.pop()
print(L3)
    
```

**Exercice n° 5** Pour chaque colonne, écrire le résultat affiché par : \_\_\_\_\_

```

L=[0,0,1,0,1,1,0,0,1]
CopyL=[]
for elt in L:
    if elt!=0:
        CopyL.append(elt)
print(CopyL)
    
```

```

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° 6**

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° 7**

On considère le tableau  $L=["i", "n", "g", "z", "l", "r", "u"]$

1. Ajouter un "o" puis un "f" à la fin de L.
2. Ajouter une "z" au début de L.
3. Écrire un algorithme qui affiche les éléments de L un par un.
4. Écrire un algorithme qui permet de remplacer tous les "z" par des "e".

---

**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. Certains prénoms sont présents plusieurs fois dans le tableau. Écrire un algorithme pour les identifier.