

# Les listes en python

## Consigne générale :

*Les tests peuvent être réalisés dans le terminal ou dans idle de python.*

*Les exercices sont à faire dans un script (fichier nommé avec une extension **.py**).*

## 1. Les listes ou tableaux

### 1.1 accéder aux éléments d'une liste

Les listes sont des objets capables de contenir d'autres objets de python comme des chaînes de caractères, des nombres, des fonctions, d'autres listes, des tuples, des dico et bien d'autres choses encore.

```
>>> liste_test = [1, 'un', 15.5, False]
```

Elements	1	'un'	15.5	False
Indice	0	1	2	3

Les éléments du tableau sont accessibles en utilisant les indices

```
>>> print (liste_test[0])
1
>>> print (liste_test[3])
False
```

Il est important de bien comprendre que la première valeur de la liste est obtenue avec l'indice **0**. C'est une particularité à connaître en programmation, on démarre toujours par l'indice **0**.

```
>>> len(liste_test)
4
```

L'objet **len()**, renvoie le nombre d'éléments contenus dans la *liste\_test*.

### 1.2 les fonctions utiles

La communauté Python a développé énormément d'outils, d'objets, fonctions et librairies dans le but de réaliser un maximum de choses en programmation.

Le lien suivant permet de voir les différentes possibilités d'utilisation concernant les listes :

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

### Exercice 1 : 1.indice\_list.py

Déclaration des variables :

```
tab = ["pain", "orange", "eau", "fruit", "legume"]
```

Tâches à réaliser :

A l'aide d'une boucle et de l'indice du tableau, faire un affichage sur une ligne d'un élément du tableau.

Affichage final :

```
indice : 0 -> pain
indice : 1 -> orange
indice : 2 -> eau
indice : 3 -> fruit
indice : 4 -> legume
```

Il est également possible d'accéder directement aux itérations du tableau sans utiliser l'indice.

La syntaxe est la suivante :

```
for element in tab :
```

*element* prendra alors à chaque tour de la boucle la valeur suivante dans le tableau.

### Exercice 2 : 4.append\_list.py

Déclaration des variables :

```
tab_a = [15,20,5,14,1,0,19,26,4,6,5]
tab_b = [3,26,18,3,7,6,9,20,8,1,23]
```

Tâches à réaliser :

Comparer chaque élément du tableau. L'élément le plus grand des deux doit être placé dans une nouvelle liste.

Affichage final :

```
les plus grands sont : [15, 26, 18, 14, 7, 6, 19, 26, 8, 6, 23]
```

### Exercice 3 : 5.compter\_les doublons.py

Déclaration des variables :

```
serie_a = [1,5,1,6,8,4,5,9,5,6,1,1,4,7,6,2,5,6,0,0,7,8,5,6,1,2]
```

Tâches à réaliser :

Afficher la liste de départ.

Afficher la liste de départ trié

Compter le nombre d'apparition des valeurs de 0 à 9 dans la liste

Afficher le résultat comme ci-dessus.

Affichage final :

```
[1, 5, 1, 6, 8, 4, 5, 9, 5, 6, 1, 1, 4, 7, 6, 2, 5, 6, 0, 0, 7, 8, 5, 6, 1, 2]  
[0, 0, 1, 1, 1, 1, 1, 2, 2, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7, 8, 8, 9]  
0 apparait : 2  
1 apparait : 5  
...
```

