

# L5 - Listes et tableaux

*2019-2020*

# 1 - Listes

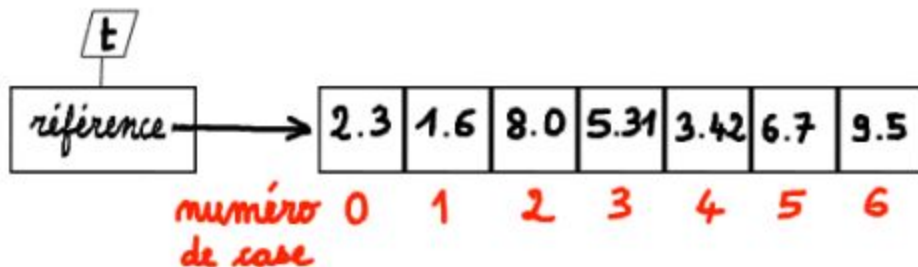
## Définition :

Une liste est une **collection d'éléments** séparés par des **virgules**, l'ensemble étant entre crochets [...].

Le premier élément d'une liste possède l'**indice 0**. Dans une liste on peut avoir des éléments de plusieurs types

**Exemple :**  $t = [2.3, 1.6, 8.0, 5.31, 3.42, 6.7, 9.5]$

Le premier élément est 2.3, et il est noté  $t[0]$ .



# Créer une liste

- Par simple affectation de valeurs

```
ma_liste = [2014, 'Python', 3.14]
```

- Par simple affectation d'une liste vide qui sera remplie plus tard

```
ma_liste = []
```

- Par une définition sous forme de boucle

```
ma_liste = [2*k for k in range(10)]  
ma_liste = [0 for k in range(10)]
```

- Par la répétition d'un élément de base

```
ma_liste = [0]*10
```

# Accès aux éléments d'une liste

Comme pour les chaînes de caractères, les éléments sont numérotés à partir de 0.

```
>>> ma_liste = [2014, 'Python', 3.14]
>>> ma_liste[1]
'Python'
```

# Modification d'une liste

- Modifier un élément préexistant

```
>>> ma_liste = [2014, 'Python', 3.14]
>>> ma_liste[1] = 'Anaconda'
>>> ma_liste
[2014, 'Anaconda', 3.14]
```

- Ajout d'un élément en fin de liste

```
ma_liste.append('Vacances')
ou
ma_liste = ma_liste + ['Vacances']
```

# Parcours d'une liste

Pour afficher tous les éléments un à un deux possibilités :

```
for k in range(len(ma_liste)) :  
    print(ma_liste[k])
```

ou

```
for element in ma_liste :  
    print(element)
```

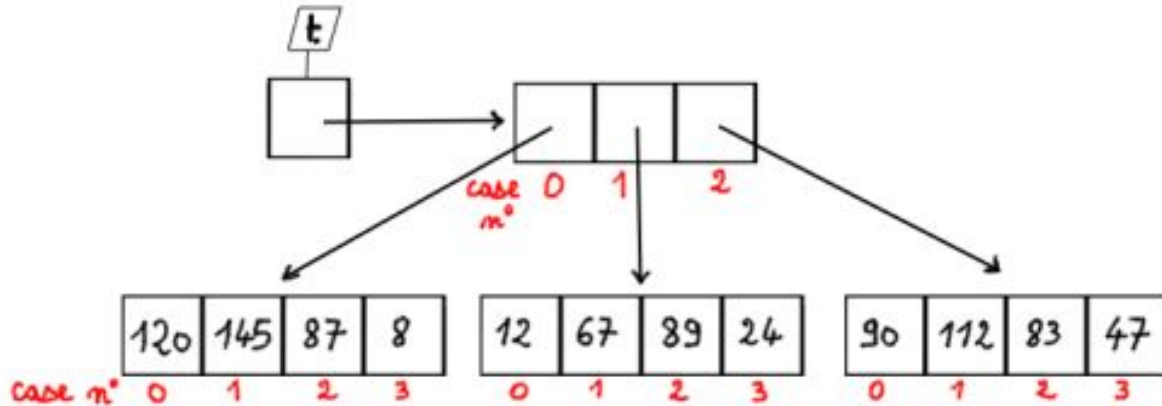
# 2 - Les tableaux

Exemple de tableau à deux dimensions :

120	145	87	8
12	67	89	24
90	112	83	47

# Cas particulier des tableaux à 2 dimensions :

On remarque que dans ce tableau à deux dimensions, chaque élément du tableau  $t$  est lui même un tableau.



Par exemple  $t[1][2]$  correspond à l'élément contenant la valeur 89.