

## TP PYTHON (2) : LES LISTES

Voici comment on crée une liste vide : **ligne 1** et **ligne 2**

```

1 liste1 = list() # création d'une liste
2
3 liste2 = [] # création d'une liste
4
5 liste3 = [1, 2, 3, 4, 5] # une liste de 5 éléments
6
7 # liste contenant des chaînes
8 listeFruits = ["orange", "citron", "pomme", "raisin"]

```

On peut également créer une liste par initialisation de contenu : **ligne 3** et **ligne 4**

Pour accéder aux différents éléments d'une liste, vous pouvez y accéder de la manière suivante :

```

>>> liste = [1, 2, 3, 4, 5]
>>> liste[0]
1
>>> liste[4]
5

```

Ainsi, vous accéder au premier élément de la liste par son numéro d'emplacement. Le premier élément d'une liste se trouve toujours à l'indice 0.

Le dernier élément d'une liste peut être atteint par l'instruction suivante :

```

>>> listeFruits = ["orange", "pomme", "banane", "citron"]
>>> listeFruits[ len(listeFruits) - 1 ]
'citron'

```

L'emploi de la méthode **len** nous retourne la longueur de la liste. Ici, la valeur **4**, auquel on soustrait **1** pour obtenir le dernier indice de la liste.

Contrairement à la chaîne que l'on ne peut modifier, les éléments d'une liste peuvent être modifié. On dit alors qu'une liste est **mutable**.

```

>>> listeFruits = ["orange", "pomme", "banane", "citron"]
>>> listeFruits[1] = "poire"
>>> listeFruits
['orange', 'poire', 'banane', 'citron']

```

### Slice avec deux indices<sup>1</sup>

Une expression de la forme `liste[2 : 6]` est un slice dans sa syntaxe de base. Cette syntaxe utilise deux indices, ici les indices 2 (indice du début du slice) et 6 (indice de fin du slice). Le deuxième indice est strictement à gauche de l'indice 6, autrement dit jusqu'à l'indice 5 inclus.

```

1 liste = [ 2, 6, 8, 4, 9, 7, 1, 5 ]
2
3 print( liste[2:6] )

```

**[8, 4, 9, 7]**

## Des chaînes en liste avec Split()

Si vous désirez créer une liste à partir d'une chaîne, vous pouvez utiliser la méthode `split()`. Cette méthode va découper notre chaîne initiale en autant d'éléments qui la constitue. Ainsi, la chaîne "Programmation en Python" sera séparé ainsi :

```
>>> chaine = "Programmation en Python"
>>> chaine.split(" ")
['Programmation', 'en', 'Python']
```

La méthode `split()` appliquée sur l'objet chaîne, nous retourne une liste d'éléments. Remarquer que le paramètre fourni à `split` est une espace vide. La chaîne est alors **découpée** à partir des espaces vides entre les éléments qui la constitue.

## Trier une liste avec Sort()

Appliquer la méthode `sort()` sur une liste signifie que l'on modifie le contenu de la liste en ordonnant le contenu par ordre de grandeur ou par ordre alphabétique pour une liste contenant des chaînes.

Pour une liste de chaînes

```
>>> listeFruits = ["orange", "raisin", "banane", "ananas"]
>>> listeFruits.sort()
>>> print(listeFruits)
['ananas', 'banane', 'orange', 'raisin']
```

## Ajouter un item dans une liste

La méthode `append()` permet d'ajouter un élément à une liste.

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 liste.append(6)
3 print(liste)
```

```
[1, 2, 3, 4, 5, 6]
```

## Ajouter un item dans une liste en précisant son emplacement

La méthode `insert()` permet d'ajouter un élément à une liste en précisant son emplacement. Cette fonction reçoit deux paramètres : le premier, le numéro d'index. Le deuxième paramètre correspond au contenu ajouté.

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 liste.insert(2, "python")
3 print(liste)
```

```
[1, 2, 'python', 3, 4, 5]
```

### Retirer un item d'une liste

La méthode `remove()` permet de retirer la première occurrence de l'élément passé en paramètre.

Retirer un item d'une liste en précisant son index

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 liste.remove(2)
3 print(liste)

[1, 3, 4, 5]
```

La méthode `del()` permet de retirer un élément d'une liste en précisant son numéro d'index

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 del liste[3]
3 print(liste)

[1, 2, 3, 5]
```

### Obtenir la valeur maximum d'une liste

La méthode `max()` retourne la valeur la plus élevée présente dans une liste.

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 maximale = max(liste)
3 print(maximale)

5
```

### Obtenir la valeur minimum d'une liste

La méthode `min()` retourne la valeur la moins élevée présente dans une liste.

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 plusPetiteValeur = min(liste)
3 print(plusPetiteValeur)

1
```

### Obtenir la somme d'une liste

La méthode `sum()` retourne le total des valeurs présentes dans une liste.

```
1 liste = [ 1, 2, 3, 4, 5 ]
2 somme = sum(liste)
3 print(somme)

15
```

### Concaténer une liste dans une chaîne

La méthode `join()` permet de concaténer chacun des éléments d'une liste à une chaîne

```
1 chaine = str()
2 liste = [ 'a', 'b', 'c' ]
3 chaine = chaine.join(liste)
4 print(chaine)

abc
```

### Afficher une liste à l'envers

Il est possible d'afficher une liste à l'envers à partir des paramètres de découpage (slicing) :

```
1 liste = [ 1, 2, 3, 4, 5 ]
2
3 print(liste[::-1])

[5, 4, 3, 2, 1]
```

### Inverser une liste avec la fonction reverse

La fonction `reverse()` modifie le contenu d'une liste en inversant son contenu :

```
1 liste = [ 1, 2, 3, 4, 5 ]
2
3 liste.reverse()
4
5 print(liste)

[5, 4, 3, 2, 1]
```