

# LES LISTES EN PYTHON

Réalisée par:

Touayba Baissa  
Meryeme Errajraji

December 22, 2023

# PLAN

## 1 Les Listes

- Définition
- Indéçage
- Opération sur les listes
- Minimum ,maximum et somme d'une liste
- Exercice

## 2 Tuples

- Définition
- Exemple

# Les listes

- **Une liste** est une structure de données qui contient une série de valeurs séparées par des virgules, chaque valeur est automatiquement identifiée par un indice.

Exemple:

```
liste1=[ 1, 2, 3, 4]
```

```
liste2=['paris', 'berlin', 'londres', 'bruxelle']
```

# Indeçage

- un des gros avantages d'une liste est que vous pouvez appeler ses éléments par leur position. ce numéro est appelé **indice**(ou index) de la liste et peut être également positif ou négatif

## Exemple:

```
animaux = ["girafe", "tigre", "singe", "souris"]
```

indice positif	0	1	2	3
----------------	---	---	---	---

indice négatif	-4	-3	-2	-1
----------------	----	----	----	----

```
In [32]: y = 'Meryeme'
          print (y.index('m'))
          x = [0, 3, 9, 4]
          print (x.index(9))
```

5  
2

# Opération sur les listes

- Ajouter deux listes ensemble

L'opération `+` permet d'ajouter deux listes ensemble pour former une troisième liste

## Exemple:

```
In [6]: x = ["casa", "rabat"]
        y = ["zagora", "ouarzazat"]
        print(x + y)
        x = [0, 3, 9, 4]
        y = [12, 3, 4, 6, 10, 88, 6]
        print(x + y)

['casa', 'rabat', 'zagora', 'ouarzazat']
[0, 3, 9, 4, 12, 3, 4, 6, 10, 88, 6]
```

- Créer une liste étendue

L'opérateur `*` permet de créer une liste en répétant le contenu d'une liste autant de fois que spécifié

### Exemple:

```
In [10]: x = ["casa", "rabat"]
          print(x * 2)
          y = [12, 3, 4]
          print(y * 3)

['casa', 'rabat', 'casa', 'rabat']
[12, 3, 4, 12, 3, 4, 12, 3, 4]
```

Figure 3: liste répétant

# Opération sur les listes

## connaître la longueur d'une liste

L'instruction `len()` vous permet de connaître la longueur d'une liste, c'est-à-dire le nombre d'éléments que contient la liste.

### Exemple:

```
In [11]: x = ["casa", "rabat"]  
         Y = [0, 3, 9, 4]  
         taillX=len(x)  
         print(taillX)  
         taillY=len(Y)  
         print(taillY)
```

2

4

## Ajouter une valeur à une liste

vous pouvez utiliser la méthode `append()` lorsque vous souhaitez ajouter **un seul** élément à la fin d'une liste

### Exemple:

```
In [11]: x = [12, 3, 4, 6, 10, 88, 6]
          x.append(0)
          print(x)

          [12, 3, 4, 6, 10, 88, 6, 0]
```

Figure 5: append



## insérer des éléments dans une liste

on utilise la méthode python `insert()` pour insérer un seul élément dans une python liste existante et se construit avec deux paramètres: **l'élément à insérer** et **l'indexe** (l'emplacement) où cet élément doit être inséré.

### Exemple:

```
In [1]: x = [8, 18, 0, 6]
x.insert(4,88)
print(x)
x.insert(1,['ME','AB'])
print(x)

[8, 18, 0, 6, 88]
[8, ['ME', 'AB'], 18, 0, 6, 88]
```

Figure 6: insert

## Supprimer un élément d'une liste

Pour supprimer un élément d'une liste en fonction de l'indice (position) on peut utiliser la fonction **del**:

### Exemple:

```
In [5]: x = [19, 1, 99, 100]
del(x[2])
print(x)

[19, 1, 100]
```

Figure 7: del

## inverser les valeurs d'une liste

pour inverser l'ordre des éléments d'une liste en utilise la méthode **reverse()**

### Exemple:

```
In [18]: x = [0, 3, 9, 4]
         x.reverse()
         x
```

```
Out[18]: [4, 9, 3, 0]
```

Figure 8: reverse

## Trier une liste de nombres par ordre croissant ou décroissant

pour trie tous les éléments de la liste par ordre croissant ou décroissant en utilise la méthode **list.sort()**

### Exemple:

```
In [20]: x = [0,9 , 8, 2]
          x.sort()
          x
```

```
Out[20]: [0, 2, 8, 9]
```

Figure 9: sort

# Minimum, Maximum et Somme d'une liste

## Minimum d'une liste

pour renvoyer le minimum d'une liste passée en argument en utilise la fonction **min()**

### Exemple:

```
In [29]: x = [0, 3, 9, 4]
          print (min(x))
          y = 'bug'
          print (min(y))
```

0

b

Figure 10: min

## Maximum d'une liste

pour renvoyer le maximum d'une liste passée en argument en utilise la fonction **max()**

### Exemple:

```
In [28]: x = [0, 3, 9, 4]
          print (max(x))
          y = 'bug'
          print (max(y))
```

9

u

Figure 11: max

## Somme d'une liste

pour calculer la somme des termes d'une liste de nombres, on va d'abord initialiser une variable à 0, puis on va lui ajouter, un à un, successivement tous les termes de la liste en utilisant la fonction **sum()**

### Exemple:

```
In [23]: x = [0, 3, 9, 4]  
         print (sum(x))
```

16

Figure 12: sum

```
In [24]: x = [0, 3, 'ME' , 4]
print (sum(x))
```

-----  
**TypeError**

Traceback

Cell In[24], line 2  
 1 x = [0, 3, 'ME' , 4]  
----> 2 print (sum(x))

**TypeError:** unsupported operand type(s) for +: 'int'

```
In [27]: x = [0, 3, 9, 4]
print (sum(x[-2:]))
```

13

Figure 13: SUM ERROR



## EXERCICE

Ecrire un programme en python qui demande à l'utilisateur de saisir 5 nombre entiers de son choix et de lui afficher la liste des nombres saisis.

## SOLUTION

```
In [4]: listNombres = []
        for i in range(0,5):
            n = int(input("Tapez la valeur d'un entier : "))
            listNombres.append(n)
        print("Voici la liste des nombres saisis : " , listNombres)
```

Tapez la valeur d'un entier : 4  
Tapez la valeur d'un entier : 0  
Tapez la valeur d'un entier : 3  
Tapez la valeur d'un entier : 6  
Tapez la valeur d'un entier : 2  
Voici la liste des nombres saisis : [4, 0, 3, 6, 2]

Figure 14: solution

# TUPLE

**un tuple** est une liste qui ne plus être modifiée et l'ensemble d'éléments est entouré de parenthèses plutôt que de crochets.

## Exemple:

```
tup = ("première_valeur", "deuxième_valeur", "troisième_val  
print(tup)
```

Figure 15: tuple