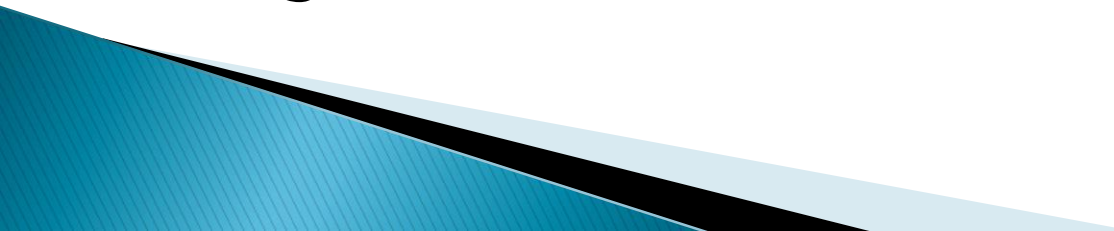


Les listes

Définition

- ▶ Une **liste** est un type de données qui contient une série de valeurs (on parle plus de donnée simple).
 - ▶ Python autorise la construction de liste contenant des valeurs de types différents (par exemple entier et chaîne de caractères), ce qui leur confère une grande flexibilité.
 - ▶ Les valeurs d'une liste sont séparés par virgule (,) et sont mise entre crochets []
- 

Créer une liste

- ▶ Voici des exemples :

```
animaux = ["girafe", "tigre", "singe", "souris"]
tailles = [5, 2.5, 1.75, 0.15]
mixte = ["girafe", 5, "souris", 0.15]
listeV = [] #liste vide

print(animaux)
print(tailles)
print(mixte)
print(listeV)
```

Indices et éléments d'une liste

- ▶ Un des gros avantages d'une liste est que vous pouvez appeler ses éléments par leur position. Ce numéro est appelé **indice** (**index**) de la liste.

```
maliste : ["girafe", "tigre", "singé", "souris"]  
indice :   0         1         2         3
```

- ▶ Les indices d'une liste de n éléments commence à 0 et se termine à $n-1$.

```
animaux = ["girafe", "tigre", "singé", "souris"]  
print(animaux[0])  
print(animaux[1])  
print(animaux[3])
```

Modifier la valeur d'un élément de la liste

- Pour modifier la valeur d'un élément de la liste, on fait une simple affectation:

```
animaux[0]="lapin"
```

- Résultat:

```
print(animaux)
```

Donne:

```
['lapin', 'tigre', 'singe', 'souris']
```

Opérations sur les listes

- ▶ Tout comme les chaînes de caractères, les listes supportent l'opérateur + de concaténation, ainsi que l'opérateur * pour la duplication :

```
ani1 = ["girafe", "tigre"]  
ani2 = ["singe", "souris"]  
print(ani1 + ani2)  
print(ani1 * 3)
```

- ▶ Résultat:

```
['girafe', 'tigre', 'singe', 'souris']  
['girafe', 'tigre', 'girafe', 'tigre', 'girafe', 'tigre']
```

Indexage négatif

- La liste peut également être indexée avec des nombres négatifs selon le modèle suivant :

```
maliste      : ["girafe", "tigre", "singé", "souris"]  
indice positif :    0      1      2      3  
indice négatif :   -4     -3     -2     -1
```

```
animaux = ["girafe", "tigre", "singé", "souris"]  
print(animaux[-1])  
print(animaux[-2])
```

- Résultat:

```
souris  
singé
```

Les tranches

- ▶ Avec les listes on peut sélectionner une partie d'une liste en utilisant un indexage construit sur le modèle **[m:n+1]** pour récupérer tous les éléments, du m ème au $n+1$ ème (de l'élément m inclus à l'élément $n+1$ exclu). On dit alors qu'on récupère une **tranche** de la liste.

Les tranches (exemples)

► Exemples:

```
animaux = ["girafe", "tigre", "singe", "souris"]  
print(animaux[0:2])  
print(animaux[0:3])  
print(animaux[0:])  
print(animaux[:])
```

► Résultat:

```
['girafe', 'tigre']  
['girafe', 'tigre', 'singe']  
['girafe', 'tigre', 'singe', 'souris']  
['girafe', 'tigre', 'singe', 'souris']
```

La fonction **len()**

- ▶ La fonction **len()** vous permet de connaître la longueur d'une liste, c'est-à-dire le nombre d'éléments que contient la liste.
- ▶ Exemple:

```
animaux = ["girafe", "tigre", "singe", "souris"]  
L=len(animaux)  
print(L)  
X=len([1, 2, 3, 4, 5, 6, 7, 8])  
print(X)
```

- ▶ Résultat:

```
4  
8
```

Parcourir une liste

- ▶ Pour parcourir une liste on peut se servir de la boucle for:
- ▶ Exemple:

```
T=[12,5,76,-12]
```

```
for x in T :  
    print(x)
```

- ▶ Résultat:

```
12  
5  
76  
-12
```

Parcourir une liste

- ▶ Parcourir une liste à l'aide de la boucle **for** et la fonction **range()**

```
MaListe=[12,8,-4,19,34]  
for i in range(0,5):  
    print(MaListe[i])
```

- ▶ En utilisant la fonction **len()**:

```
Liste=[12,8,-4,19,34]  
  
for i in range(0,len(MaListe)):  
    print(MaListe[i])
```

Parcourir une liste

- ▶ Parcourir une liste à l'aide de la boucle **while**
- ▶ Exemple:

```
t=[12,5,76,-12]
i=0
while i in range(len(t)):
    print(t[i])
    i+=1
```

Ajouter un élément à une liste

- ▶ Pour ajouter un élément à une liste on se sert de la fonction **append()**
- ▶ Exemple:

```
animaux = ["girafe", "tigre", "singé", "souris"]  
animaux.append("Lion")  
print(animaux)
```

- ▶ Résultat:

```
['girafe', 'tigre', 'singé', 'souris', 'Lion']
```

Ajouter des valeurs dans une liste

- ▶ Exemple: dans cet exemple, l'utilisateur doit saisir les valeurs de la liste

```
t=[]          #t est une liste vide

n=int(input("saisir le nombre d'éléments dans le tableau"))

for i in range(n):
    valeur=int(input(F"nombre numéro {i+1} ))
    t.append(valeur)

for x in t:
    print(x)
```

Insérer un élément dans une liste

- ▶ On a la possibilité d'insérer un élément dans une liste à une position bien déterminée à l'aide de la fonction **insert()**.

- ▶ Exemple:

```
a = [1, 2, 3]  
a.insert(2, -15)  
print(a)
```

- ▶ Résultat:

```
[1, 2, -15, 3]
```


Supprimer un élément d'une liste

- ▶ L'instruction **del** supprime un élément d'une liste à un indice déterminé :

- ▶ Exemple:

```
a = [1, 2, 3]  
  
del a[1]  
  
print(a)
```

- ▶ Résultat:

```
[1, 3]
```

Supprimer un élément de la liste

- ▶ La méthode **remove()** supprime un élément d'une liste à partir de sa valeur.

- ▶ Exemple:

```
a = [1, 2, 3]  
a.remove(3)  
print(a)
```

- ▶ Résultat:

```
[1, 2]
```

Vider une liste de son contenu

- ▶ Si on veut vider une liste, on peut écrire:

```
del MaListe[:]
```

Trier les valeurs d'une liste

- ▶ Pour trier les éléments d'une liste, on se sert de la méthode `sort()`
- ▶ Exemple:

```
t=[12,6,-12,90,11]  
t.sort()  
  
for x in t:  
    print(x)
```