

FORMATION PYTHON

LES LISTES

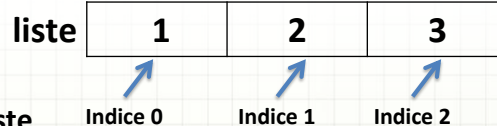


LES LISTES

Une liste est un ensemble d'éléments, on le déclare entre crochets.

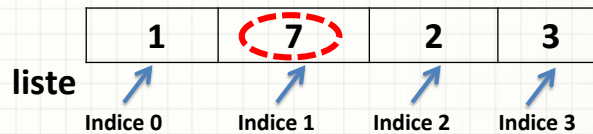
- Pour créer une liste

```
>>> liste = [1,2,3]
```



- Pour voir le contenu de la liste

```
>>> liste
[1, 2, 3]
```



- Insérer un élément dans une liste

```
>>> liste.insert(1, 7 )
```

- Pour ajouter un élément après la création avec la méthode append

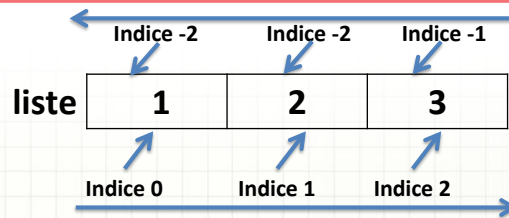
```
>>> liste.append(4)
>>> liste.append("ok")
>>> liste
[1, 2, 3, 4, 'ok']
```

LES LISTES

Affichage et Modification

Afficher un item d'une liste

```
>>> liste[0]
1
>>> liste[2]
3
```



Il est d'ailleurs possible de modifier une valeur avec son index

```
>>> liste[2] = 5
>>> liste
[1, 2, 5]
```

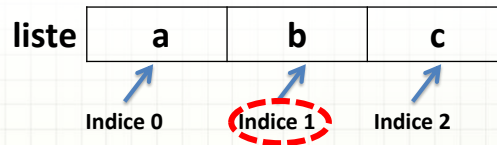
3

LES LISTES

Suppression

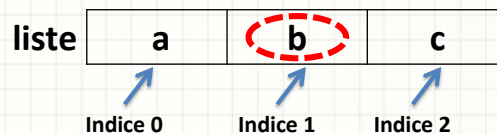
Supprimer une entrée avec un index

```
>>> liste = ["a", "b", "c"]
>>> del liste[1]
>>> liste
['a', 'c']
```



Supprimer une entrée avec sa valeur

```
>>> liste = ["a", "b", "c"]
>>> liste.remove("a")
>>> liste
['b', 'c']
```



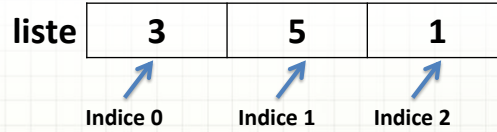
4

LES LISTES

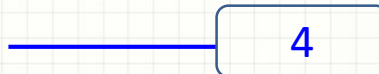
Manipulation

Inverser les valeurs d'une liste

```
>>> liste = [3, 5, 1]
```



```
>>> print len(liste)
```



```
>>> print(min(liste))
```



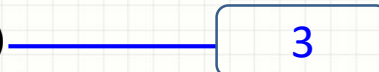
```
>>> print(max(liste))
```



```
>>> print sum(liste)
```



```
>>> print sum(liste)/len(liste)
```



5

LES LISTES

Manipulation

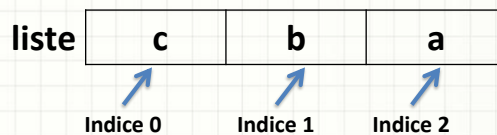
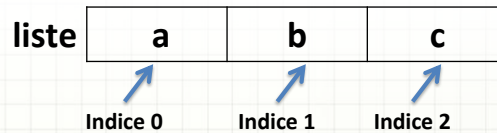
Inverser les valeurs d'une liste

```
>>> liste = ["a", "b", "c"]
```

```
>>> liste.reverse()
```

```
>>> liste
```

```
['c', 'b', 'a']
```



6



LES LISTES

Manipulation

Compter le nombre d'occurrences d'une valeur

```
>>> liste = ["a", "a", "a", "b", "c", "c"]
>>> liste.count("a")
3
>>> liste.count("c")
2
```

Trouver l'index d'une valeur

```
>>> liste.index("b")
3
```

Trouver un item dans une liste

Pour savoir si un élément est dans une liste

```
>>> liste = [1,2,3,5,10]
>>> 3 in liste
True
>>> 11 in liste
False
```

7



LES LISTES

Manipulation

Voici quelques manipulation des listes:

```
>>> liste = [1, 10, 100, 250, 500]
>>> liste[0]
1

>>> liste[-1] # Cherche la dernière occurrence
500

>>> liste[-4:] # Affiche les 4 dernières occurrences
[500, 250, 100, 10]

>>> liste[:] # Affiche toutes les occurrences
[1, 10, 100, 250, 500]
```

8

LES LISTES

Manipulation

Pour afficher les valeurs d'une liste, on peut utiliser une boucle:

```
>>> liste = ["a", "d", "m"]
>>> for lettre in liste:
...     print (lettre)
a
d
m
```

Si vous voulez en plus récupérer l'index, vous pouvez utiliser la fonction enumerate .

```
>>> for lettre in enumerate(liste):
...     print (lettre)
(0, 'a')
(1, 'd')
(2, 'm')
```

9

LES LISTES

Manipulation

Copier une liste

```
>>> x = [1,2,3]
>>> y = x
```



si vous changez une valeur de la liste y , la liste x sera elle aussi affectée par cette modification:

```
>>> x = [1,2,3]
>>> y = x
>>> y[0] = 4
>>> x
[4, 2, 3]
```

10

LES LISTES

Manipulation

La fonction range

La fonction range génère une liste composée d'une simple suite arithmétique.

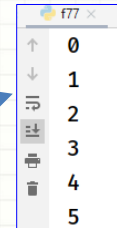
Syntaxe: range (start, stop[, step])

Exemple:

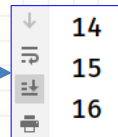
```
x = range(6)
for n in x:
    print(n)
```

```
x = range(14, 17)
for n in x:
    print(n)
```

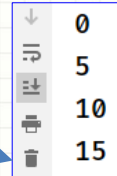
```
x = range(0, 20, 5)
for n in x:
    print(n)
```



```
0
1
2
3
4
5
```



```
14
15
16
```



```
0
5
10
15
```

11

LES LISTES

Manipulation

La fonction range

Exemple

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Agrandir une liste par une liste avec la méthode extend

```
>>> x = [1, 2, 3, 4]
>>> y = [4, 5, 1, 0]
>>> x.extend(y)
>>> print x
[1, 2, 3, 4, 4, 5, 1, 0]
```

12

LES LISTES

Manipulation

Vous pouvez additionner deux listes pour les combiner ensemble en utilisant l'opérateur + :

```
>>> x = [1, 2, 3]
>>> y = [4, 5, 6]
>>> x + y
[1, 2, 3, 4, 5, 6]
```

Vous pouvez même multiplier une liste:

```
>>> x = [1, 2]
>>> x*5
[1, 2, 1, 2, 1, 2, 1, 2, 1, 2]
```

Ce qui peut être utile pour initialiser une liste:

```
>>> [0] * 5
[0, 0, 0, 0, 0]
```

13

LES LISTES

Opérations sur les listes

```
import statistics
liste=[17,9,12,10]
print(liste)
print(statistics.mean(liste))
```

```
from statistics import mean
liste=[17,9,12,10]
print(liste)
print(mean(liste))
```

```
[17, 9, 12, 10]
12
```

14

LES LISTES

Opérations sur les listes

```
from statistics import mean
import random
liste=[17,9,12,10]
print(liste)
print(mean(liste))
random.shuffle(liste)
print(liste)
```

```
[17, 9, 12, 10]
12
[9, 10, 12, 17]
```

15

LES LISTES

Listes imbriquées

```
a = [[1, 2, 3], [4, 5, 6]]
print(a[0])
print(a[1])
b = a[0]
print(b)
print(a[0][2])
a[0][1] = 7
print(a)
print(b)
b[2] = 9
print(a[0])
print(b)
```

16



LES LISTES

Listes imbriquées

Pour traiter un tableau à deux dimensions, vous utilisez généralement des boucles imbriquées.

```
a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]  
for i in range(len(a)):  
    for j in range(len(a[i])):  
        print(a[i][j], end=' '  
    print()
```

```
1 2 3 4  
5 6  
7 8 9
```

17