

INFO-H-100 - Informatique

Séance d'exercices 5
Introduction à Python
Listes

Université libre de Bruxelles
École polytechnique de Bruxelles

2013-2014

Listes

En Python, une liste est une **séquence** d'éléments qui peuvent être de types différents.

Une liste de taille n est **indiquée** de 0 à $n - 1$ et de -1 à $-n$ et on peut accéder à chaque élément à l'aide des **crochets**.

On utilise également les crochets pour construire une liste.

```
>>> li1 = []           #empty list []
>>> type(li1)
<type 'list'>
>>> li1
[]
>>> li2 = [1,2,3,4]
>>> li2
[1, 2, 3, 4]
>>> li2[2]
3
>>> li2[-1]
4
>>> li3 = ["SPAM", True, ('eggs', 42)] #different types of elements
>>> print(li3[2][0][3])
s
```

Opérations sur les listes

Les listes et les chaînes étant des séquences, certaines opérations sont similaires.

```
>>> li1 = [1,2,3,4]
>>> li2 = [5,6,7,8]
>>> li1 + li2
[1, 2, 3, 4, 5, 6, 7, 8]
>>> li1[1:] + li2[:-1]
[2, 3, 4, 5, 6, 7]
>>> len(li1)
4
>>> 5 in li1
False
>>> li2 = li2 * 3
>>> li2
[5, 6, 7, 8, 5, 6, 7, 8, 5, 6, 7, 8]
>>> li2.index(7)
2
>>> li2.count(8)
3
```

Listes et for

```
>>> ls = [1, 2, 3]
>>> for x in ls:
    print(x)
```

```
1
2
3
```

```
>>> ls = [1, 2, 3]
>>> for i in range(len(ls)):
    print("pos:", i, ", val:", ls[i])
```

```
pos: 0 -> val: 1
pos: 1 -> val: 2
pos: 2 -> val: 3
```

```
def sum_list(li):
    total = 0
    for item in li:
        total += item
    return total
```

```
ls = [ 1, 2, 3 ]
```

```
print(sum_list(ls)) # -> 6
```

Consultez la documentation pour trouver d'autres opérations.

Une liste est une séquence mutable.

Contrairement aux chaînes, une liste est **mutable**, c'est-à-dire que l'on peut la changer.

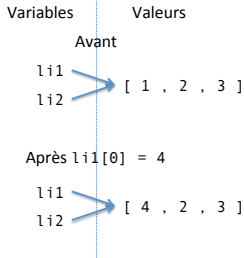
```
>>> my_list = [1,7]
>>> my_list[1] = 2
>>> my_list
[1, 2]
```

```
>>> message = "bienvenue"
>>> message[0] = 'B'
TypeError: 'str' object does not support item assignment
```

Une liste est une séquence mutable.

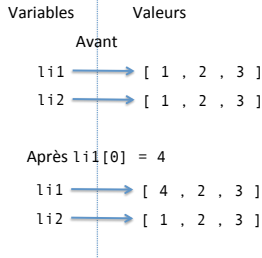
```
>>> li1 = [1,2,3]
>>> li2 = li1
>>> li1[0] = 4
>>> print(li1)
[4, 2, 3]
>>> print(li2)
[4, 2, 3]
>>> li1 == li2
True
```

li2 = li1



```
>>> li1 = [1,2,3]
>>> li2 = li1[:]
>>> li1[0] = 4
>>> print(li1)
[4, 2, 3]
>>> print(li2)
[1, 2, 3]
>>> li1 == li2
False
```

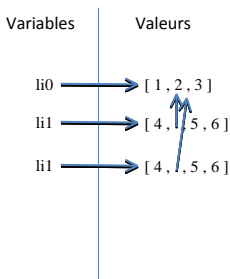
li2 = li1[:]



Une liste est une séquence mutable.

Les copies sont superficielles

```
>>> li0 = [1, 2, 3]
>>> li1 = [4, 5, li0, 6, 7]
>>> li2 = li1[:]
>>> li1[2][0] = 8
>>> print(li2)
[4, 5, [8, 2, 3], 6, 7]
```



Nouvelles opérations sur les listes

De par leur mutabilité, les listes possèdent d'autres opérations :

```
>>> li1 = [5,2,6,7,1]
>>> li1.append(9)           #add an item to the end
>>> li1
[5, 2, 6, 7, 1, 9]
>>> li1.sort()             #sort the items by ascending order
>>> li1
[1, 2, 5, 6, 7, 9]
>>> li1.insert(2,'eggs')   #insert an item at a given position
>>> print(li1)
[1, 2, 'eggs', 5, 6, 7, 9]
>>> del li1[4]             #remove an item
>>> li1
[1, 2, 'eggs', 5, 7, 9]
>>> list("SPAM")           #convert to list
['S', 'P', 'A', 'M']
>>> li1.extend([3, 4, 5])  #extend with another list
>>> li1
[1, 2, 'eggs', 5, 7, 9, 3, 4, 5]
>>> li1.append([6, 7, 8])  # append another list
>>> li1
[1, 2, 'eggs', 5, 7, 9, 3, 4, 5, [6, 7, 8]]
```

Consultez la documentation pour trouver d'autres opérations.

Listes de listes et for imbriqués

```
>>> mat = [ [ 1 , 2 , 3 , 4 ],  
             [ 5 , 6 , 7 , 8 ],  
             [ 9 , 10 , 11 , 12 ] ]
```

```
>>> len(mat)
```

```
3
```

```
>>> mat[1]
```

```
[5, 6, 7, 8]
```

```
>>> mat[1][0]
```

```
5
```

```
>>> for line in mat:  
        print(sum_list(line))
```

```
10
```

```
26
```

```
42
```

```
def sum_list_of_lists(li):  
    total = 0  
    for line in li:  
        for item in line:      # or total += sum_list(line)  
            total += item      #  
    return total
```

```
print(sum_list_of_lists(mat)) # -> 78
```

Exercices