

# INFO-H-100 - Informatique

Séance d'exercices 6  
Introduction à Python  
Listes

Université Libre de Bruxelles  
Faculté des Sciences Appliquées

2011-2012

# 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 = [] #liste vide []
>>> 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)] #types differents
```

# 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
```

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 qu'on peut la changer.

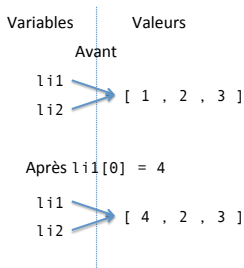
```
>>> liste = [1,7]
>>> liste[1] = 2
>>> liste
[1, 2]
```

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

# Une liste est une séquence mutable.

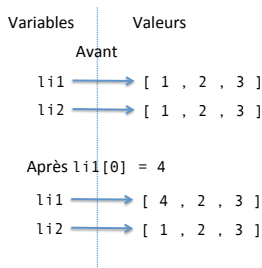
```
>>> li1 = [1,2,3]
>>> li2 = li1
>>> li1[0] = 4
>>> print li1
[4, 2, 3]
>>> print li2
[4, 2, 3]
>>> li1 == li2
True
```

**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[:]**



# Nouvelles opérations sur les listes

Du fait de leur mutabilité, les listes possèdent d'autres opérations :

```
>>> li1 = [5,2,6,7,1]
>>> li1.append(9)           #ajout d'un element a la fin
>>> li1
[5, 2, 6, 7, 1, 9]
>>> li1.sort()            #tri
>>> li1
[1, 2, 5, 6, 7, 9]
>>> li1.insert(2, 'eggs')  #ajout a une position
>>> print li1
[1, 2, 'eggs', 5, 6, 7, 9]
>>> del li1[4]            #suppression d'un element
>>> li1
[1, 2, 'eggs', 5, 7, 9]
>>> list("SPAM")         #conversion vers liste
['S', 'P', 'A', 'M']
>>> str(li1)             #conversion vers chaine
"[1, 2, 'eggs', 5, 7, 9]"
```

Consultez la documentation pour trouver d'autres opérations.

# Listes et for

```
def sommeListe(li):  
    somme = 0  
    for element in li:  
        somme += element  
    return somme  
  
ls = [ 1, 2, 3 ]  
  
print sommeListe(ls) # -> 6
```

## 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 ligne in mat:
```

```
    print sommeListe(ligne)
```

```
10
```

```
26
```

```
42
```

```
def sommeListeDeListes(li):
```

```
    somme = 0
```

```
    for ligne in li:
```

```
        for element in ligne: # ou somme += sommeListe(ligne)
```

```
            somme += element #
```

```
    return somme
```

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



# Exercices