

Fiche cours 6 : manipulation des listes

1 Introduction

Une des classes les plus utilisées en Python lorsque on veut manipuler une série de données est la classe des listes. On présente dans les commandes utiles à leur manipulation.

2 Commandes

2.1 Initialiser une liste

Initialiser dans ce contexte revient à affecter à une variable une liste. Il existe différentes manières suivant la liste que vous voulez initialiser. On donne des exemples, à vous d'adapter suivante ce que vous voulez faire.

1. “saisir à la main” : `L=[3,2,4,1]`.
2. “à l'aide de for” : `L= [f(i) for i in range(1,n)]` où `f` est définie de la manière suivante

```
def f(i) :  
    return i*i+3*i-1
```

Autre exemple : `L=[0 for i in range(n)]` construit une liste contenant n zéros.

Autre variante : `L=[f(i) for i in range(1,n) if i%2==1]`.

3. “à l'aide d'une boucle” :

```
L=[]  
for i in range(1,n) :  
    L.append(f(i))
```

On peut bien entendu remplacer une boucle `for` par une boucle `while`.

2.2 Ajout, concaténation, suppression

Pour ajouter un élément `a` en fin de la liste `L`, la commande est la suivante : `L.append(a)`. Si on a deux listes `L` et `M` on peut les concaténer (les mettre bout à bout) de la manière suivante :

```
L=L+M
```

Pour supprimer l'élément en position i de la liste `L`, on effectue la commande suivante :

```
L.pop(i)
```

De plus, il est possible de récupérer la valeur à supprimer dans une variable :

```
valeur = L.pop(i)
```

`valeur` contient alors l'ancienne valeur en position i de la liste L de la liste.

2.3 Parcourir une liste

Parfois, on a besoin de parcourir une liste. Pour vérifier par exemple une propriété de la liste. Voici deux manières pour effectuer cela :

```
for x in L :  
    ...  
  
for i in range(len(L)) :  
    ...
```

l'avantage de cette méthode est de savoir à quelle niveau on se situe dans la liste.

2.4 Tableau récapitulatif

Actions	syntaxe	Exemples
Initialiser une liste	<code>L=[...]</code>	<code>L=[1,2,3]</code> <code>L=[i for i in range(1,4)]</code>
Supprimer l'élément de L en indice i	<code>L.pop(i)</code>	<code>L.pop(1)</code>
Modifier l'élément en position i de L	<code>L[i]=?</code>	<code>L[1]=3</code>
Echanger <code>L[i]</code> et <code>L[j]</code>	<code>L[i],L[j]=L[j],L[i]</code>	<code>L[1],L[2]=L[2],L[1]</code>
Ajouter en fin de la liste L l'élément x	<code>L.append(x)</code>	<code>L.append(4)</code>
Copie partielle de la liste L de l'indice a à $b-1$	<code>M=L[a :b]</code>	<code>M=L[1 :3]</code>
Longueur de la liste	<code>len(L)</code>	