

Python 6 : Les listes

En Python les listes sont des listes de variables. Une liste est en fait encore une variable du type list. Elles sont utiles pour contenir les termes d'une suite par exemple, ou encore pour tracer des graphiques (voir cours 8).

1) Créer une liste

On peut définir des listes d'éléments de plusieurs manières :

A la main :

liste1=[1,8,0,-1,1.5,187] (liste1 est donc le nom de ma variable qui est du type list)

liste2=[] veut dire qu'on crée une liste vide (utile pour la remplir après)

Avec range :

liste3=list(range(0,n)) correspond à la liste [0,1,2,...n-1]

Avec une propriété (comme les ensembles) :

liste4=[2*k for k in range(0,4)] correspond à la liste [0,2,4,6]

liste5=[k**2 for k in liste4] correspond donc à [0,4,16,36]

C'est le même principe que pour décrire un ensemble, en effet :
 $\{0,2,4,6\} = \{2k, k \in \llbracket 0, 3 \rrbracket\}$

2) Repérer les éléments d'une liste

Si L est une liste (donc une variable de type list) :

L[0] est le premier élément de la liste, L[1] le second etc.

Par exemple si liste6=[3,7,100], L[0] vaut 3

Par exemple si L=list(range(0,n)), L[n-1] vaut n-1.

3) Opération sur les listes

- On peut concaténer (cela veut dire regrouper, juxtaposer) deux listes avec un + :

L1=[9,18]

L2=[-1,9]

L3=L1+L2 ¹ # L3 est la liste [9,18,-1,9]

- On peut aussi ajouter un élément à une liste avec la commande **append** :

L.append(x) ajoute à la liste L l'élément x, par exemple :

L=[0,4]

L.append(3) # on obtient alors L=[0,4,3]

¹ Cette "addition" n'est pas commutative car si vous faites L2+L1 l'ordre de la nouvelle liste ne sera pas le même. Ne pas confondre avec une addition de deux vecteurs !

- On peut donc construire des listes avec la boucle for, par exemple :

```
L=[] # L est la liste vide pour l'instant
for k in range(0,11):
    L.append(k**2+1)
```

A la fin L=[1,2,5,10,17,...,101]

- On peut supprimer un élément d'une liste avec del :
del L[k] supprime le (k+1)em élément de la liste (la liste est donc réindexée)
- On peut insérer un élément à une position spécifique avec L.insert(n,x) (ajoute l'élément x à la position n dans la liste). On obtient donc une liste avec un élément supplémentaire.
- Si on veut copier une liste on fait :
L2=[x for x in L1] # On crée une nouvelle liste L2 qui est la même que L1

Si on écrit L2=L1 les listes restent solidaires définitivement (c'est-à-dire que toute modification de l'une modifie l'autre, ce qui est embêtant !)

- On peut dupliquer une liste :
L=L*n duplique n fois la liste L # Si L=[6,0] L*3=[6,0,6,0,6,0]

4) **Recherche dans une liste**

- len(L) renvoie la longueur (le nombre d'éléments) de la liste L
Par exemple len([199,0,-19]) renvoie 3
- L.count(x) renvoie le nombre de fois où x apparaît dans L
Par exemple [7,0,9,7,1,-1].count(7) renvoie 2

Par exemple le programme suivant :

```
L=[1,0,7,5,10]
for k in range(0,len(L)):
    L[i]=L[i]**2
```

Va remplacer tous les éléments de la liste L par leur carré.