INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Leçon sur les listes

Université de Toulouse

Et les listes alors ?

- **Les chaines de caractères c'est bien mais des tableaux de valeurs (entières, réelles…) c'est mieux !**
- **De prime abord, cela se ressemble :**

```
Tablint =[1, 2, 44, 3, -5]
```

- **Mais dans le fond, cela n'a pas grand-chose à voir….**
  - Le type est un type **list** (et plus **str**)
  - Les fonctions et les méthodes qui les manipulent sont différentes même si certaines sont identiques (ex : **len(Tabint)** retournera aussi la longueur du tableau, ici 5)
  - **La duplication et la modification d'une liste sont très différentes.**

Commençons par dupliquer…

- **Exemple troublant :**

```
Tablint =[1, 2, 44, 3, -5]
Tabl2  =  Tablint
Tablint[0] = 100
print( "Le tableau dupliqué vaut : ",Tabl2)
```

Le tableau dupliqué vaut : [100, 2, 44, 3, -5]

- **Afficher un tableau de valeurs = Afficher toutes ses valeurs**
- **Je peux modifier une valeur d'une liste (ce qui n'était pas possible avec une chaine)**
- **Si je duplique un tableau et que je modifie l'original, le doublon est modifié aussi (et réciproquement) !!!**

Continuons à dupliquer…

- Il « suffit » de créer une nouvelle liste en recopiant l'ensemble de ses valeurs :

```
TablDup  =  Tablint[:]
Tablint[0] = -5
print( 'Le tableau originale : ',Tablint)
print( 'Le tableau dupliqué : ',TablDup)
```

```
Le tableau originale :  [-5, 2, 44, 3, -5]
Le tableau dupliqué :  [100, 2, 44, 3, -5]
```

- Dans ce cas les deux tableaux sont bien des entités différentes.

Création d'une liste

- **Syntaxe simple :**
  - Crochet ouvrant au début [
  - Crochet fermant à la fin]
  - Eléments séparés par des virgules

```
CreaTab = [34 ,45.3, -6 ]
```

- **Une liste est une collection de « trucs » pas homogènes :**

```
Collec = [3 ,'L', 0.24,'Texto',-6 ]
```

  - A priori nous n'utiliserons que des listes d'éléments homogènes
  - Chaque élément de la liste compte pour 1 élément

```
print('Nb d\'élts de Collec : ',len(Collec))
```

```
Nb d'élts de Collec :  5
```

  - Ici « Texto » ne compte que pour 1 élément

Création d'une liste (2)

- **On peut créer un liste vide avec la fonction liste :**

```
Tabvide = list()
```

  - Utilité : initialiser la structure avant dans rentrer dans une boucle

- **On peut créer une liste de listes :**

```
ListedeListe = [CreaTab, Collec,[-5,0 3] ]
print(ListedeListe)
```

```
[[34, 45.3, -6], [3, 'L', 0.24, 'Texto', -6], [-5, 0, 3]]
```

  - Utilité : créer des tableaux à plusieurs dimensions
  - Nous n'utiliserons pas cette notion…
  - On peut créer des listes de listes de listes…..etc !!!

Application directe

- **Créer une liste contenant les 5 nombres entiers de votre numéro de téléphone**

Accès aux éléments d'une liste

- **Tous les éléments sont accessibles en lecture et en écriture**
- **L'accès se fait directement en utilisant l'indice (la position) de l'élément dans la liste**
  - Le premier élément est d'indice 0
  - L'indice est obligatoirement un nombre entier positif
  - Le dernier élément est d'indice (longueur-1) de la liste
  - L'indice se donne avec des crochets

```
print(Collec[0])
i=3
print(Collec[i])
print(Collec[len(Collec)-1])
```

```
3
Texto
-6
```

Accès aux éléments d'une liste (2)

- **L'accès avec une boucle for**
  - avec la directive in

```
for val in Collect
        print(val)
```

```
3
L
0.24
Texto
-6
```

  - ou avec l'indice

```
for ind in range(0,len(Collec)):
    print(ind, "...", Collec[ind])
```

```
0 ... 3
1 ... L
2 ... 0.24
3 ... Texto
4 ... -6
```

  - mieux encore…avec la directive enumerate

```
for ind,val in enumerate(Collec):
    print(ind, "--->", val)
```

```
0 ---> 3
1 ---> L
2 ---> 0.24
3 ---> Texto
4 ---> -6
```

Application directe

- **Avec la liste contenant votre numéro de téléphone, rechercher combien de nombres sont pairs, et à quelle position se trouve le nombre paire le plus grand.**

Suppression d'un élément

- On est parfois amener à ajouter ou à supprimer des éléments
- Pour la suppression c'est assez simple : utilisation de la fonction del(…)

```
print(Collec)
del(Collec[3])
print(Collec)
```

```
[3, 'L', 0.24, 'Texto', -6]

[3, 'L', 0.24, -6]
```

- La technique utilisée pour les chaines fonctionne aussi

```
Texto = Texto[0:13]+ Texto[14:20]    supprime le caractère d'indice 13
Collec = Collec[0:2]+ Collec[3:5]    supprime l'élement d'indice 2
```

Ajout d'un élément

- Il n'existe pas de fonctions spécifiques
- Le plus simple : fusionner 2 listes avec l'opérateur +

```
print(Collec)
Collec = Collec + [12]
print(Collec)
Collec = Collec + [-4, 'Bob']
print(Collec)
H = math.sqrt(2)
Collec = Collec + [H]
print(Collec)
```

```
[3, 'L', 0.24, -6]
[3, 'L', 0.24, -6, 12]
[3, 'L', 0.24, -6, 12, -4, 'Bob']
[3, 'L', 0.24, -6, 12, -4, 'Bob', 1.41421356]
```

- D'autres opérateurs (la multiplication par exemple) peuvent fonctionner aussi…à essayer !

Application directe

- **Toujours avec la liste contenant votre numéro de téléphone, mettre le plus grand nombre en dernière position**

Récapitulatif sur les listes

- Liste = collection d'éléments pas forcément homogènes
- Chaque élément est repéré par un indice qui commence à 0
- L'élément d'indice i est accessible par la syntaxe [i]
- On peut modifier directement (accès en écriture par indice) un élément de la liste
- On peut parcourir une liste avec la boucle for par indice, par valeur ou les 2 (enumerate)
- On peut supprimer un élément avec la fonction del
- On peut ajouter un ou plusieurs éléments par fusion (+)