

COURS:

Thème: *Notion de Liste*

- Vous connaissez la notion de suite (u_n) définie sur \mathbb{N} .

ses premiers termes sont:

u_0 qui se lit *u indice 0*

u_1 qui se lit *u indice 1*

..... etc

u_n qui se lit *u indice n*

Imaginez que l'on mette ces $n + 1$ premiers termes entre deux crochets fermés et en les séparant par des virgules.

Vous avez la liste :

$[u_0, u_1, \dots, u_n]$ de longueur $n + 1$

- Une liste est une suite finie d'éléments séparés par une virgule et entourés aux extrémités par des crochets fermés.
- Les éléments ou termes d'une liste peuvent être des réels, des entiers, des chaînes de caractères, même des listes.

Par exemple:

`LOU = ["livre", "table", "crayon", 4.35]`

Elle a 4 termes. Sa longueur est donc 4

On écrit pour le dire:

`len(LOU)=4`

Le terme d'indice 0 est "livre". Il est noté `LOU[0]`

Le terme d'indice 3 est le réel 4.35

En Python 2.5 ou 2.7 on peut mêler dans une liste des termes de natures différentes. On parle du type d'un élément d'une liste.

- `type(LOU[0])` donne le type du terme d'indice 0.

C'est ici `'str'` car "livre" est une chaîne c-à-d `string`

- `type(LOU[3])` donnera `'float'` car 4.35 est un nombre à virgule

Par exemple :

```
LEA=[ " Florian", [1 ,23,41] , "Paul",12.5]
```

est une liste.

LEA[1] est la liste [1 ,23,41] de longueur 3

Pour avoir dans la liste LEA[1] le terme 23 qui est d'indice 1

on considère:

```
(LEA[1])[1]
```

(LEA[1][1] pour certaines versions de Python)

- Ainsi comment désigner un terme d'une liste?

On écrit le nom de la liste suivi, à droite entre crochets

fermés, de l'indice du terme.

Par exemple:

Soit la liste: LOU = [" livre " , "table" , "crayon" , 4.35]

LOU [2] va donner le 3ième élément de la liste, à savoir ' crayon '

c-à-d le terme d'indice 2.

Le premier terme de la liste LOU est LOU[0]

- *Ainsi chaque élément d'une liste de n éléments (n dans \mathbb{N}^*)*

est repéré par un entier de 0 à n-1 .

- *Liste vide.*

```
LAV=[]
```

LAV est une liste sans aucun élément car il n'y a rien entre les crochets.

Cela sert souvent.

Cela permet d'avoir le << récipient >> que l'on peut remplir ensuite.