

Algorithmique et programmation, cours, classe de première, spécialité Mathématiques

1 Généralités sur les listes

Définitions :

- Une liste est une collection ordonnée d'objets. La liste contenant les éléments a_0, a_1, \dots, a_n où n est un entier naturel, est notée $[a_0, a_1, \dots, a_n]$.
- Chaque élément de la liste est repéré par son indice, sachant que l'indice du premier élément est 0. L'élément d'indice i est noté $L[i]$.
- La longueur d'une liste est le nombre d'éléments qu'elle contient. Elle est notée $\text{len}(L)$.

Exemples :

$L=['\text{lundi}', 10, '\text{mardi}', 11]$ est une liste qui contient des valeurs entières et des chaînes de caractères.

$L[1]$ a pour valeur 10, c'est l'élément d'indice 1.

$T=[]$ définit une liste vide.

2 Mode de génération d'une liste

2.1 En extension

Définition :

On dit que l'on définit une liste L *en extension* lorsque l'on initialise la liste puis on ajoute des éléments à la liste.

Exemple [Programmation d'une liste en extension] :

```
L = []  
L.append("a")
```

ou

```
L = []  
L = L + ["a"]
```

permettent de définir une liste L et de lui ajouter l'élément "a".

2.2 En compréhension

Définition :

On dit que l'on définit une liste L *en compréhension* lorsque l'on définit la liste par une propriété caractéristique.

Exemple [Programmation d'une liste en compréhension] :

```
L=[n**2 for n in range(0,21)]
```

permet de définir la liste des 20 premiers carrés d'entiers naturels.

3 Parcours d'une liste

Syntaxe :

On peut parcourir une liste L de deux manières avec une boucle bornée :

- en parcourant l'ensemble de tous ses éléments :

```
for m in L:
```

- en parcourant ses indices :

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

Exemple [Parcourir une liste] :

Le programme suivant crée, à partir d'une première liste L , une liste T contenant les carrés des éléments de L . On pourra le tester en tapant par exemple dans la console `carresListe([3,5,9,13,16])`.

```
def carresListe(L):  
    T=[]  
    for i in range(len(L)):  
        T.append(L[i]*L[i])  
    return T
```

ou

```
def carresListe(L):  
    T=[]  
    for n in L:  
        T.append(n*n)
```