

A retenir du chapitre 3 :

Types séquentiels 1 : listes et tuples



I Types

- **list** : liste,
- **tuple** : tuple,

II Opérations sur les listes et les tuples

- **len(...)** : longueur,
- **s[k]** : extraction du k ième élément de la liste ou du tuple s (la numérotation commence à 0),
- **s[-1]** : extraction du dernier élément de la liste ou du tuple s,
- **s[i :j]** : extraction des éléments de la liste ou du tuple s de i inclus à j exclus,
- **s[i :j :p]** : extraction des éléments de la liste ou du tuple s de i inclus à j exclus avec un pas p,
- **...in ...** : test d'appartenance,

- **...not in ...** : négation du test d'appartenance,
- **[...for ...in ...]** : définition d'une liste (ou d'un tuple) en compréhension,
- **+** : concaténation de deux listes ou de deux tuples,
- **n*s** : n copies de la liste ou du tuple s ,

III Opérations sur les listes

- **l.append(x)** : rajoute l'élément x à la fin de la liste l,
- **l.pop()** : renvoie et supprime le dernier élément de la liste l,

IV Commandes du module copy

- **copy** : copie superficielle,
- **deepcopy** : copie profonde.



- La numérotation des éléments de listes et de tuples commence à 0.
- Les symboles + et * ne représentent pas l'addition et la multiplication lorsqu'ils sont appliqués à des séquences.
- Les listes sont mutables, les tuples sont immuables.
- Il faut utiliser la méthode `.append` pour ajouter un élément à une liste plutôt que la concaténation.



- Les listes et les tuples sont itérables : on peut les parcourir dans une boucle (`for x in l :`).
- Comme les listes sont mutables, elles sont plus adaptées à l'écriture de fonctions dans lesquelles elles doivent être modifiées.
- Recherche d'un entier x dans une liste d'entiers l :

```
1 def recherche(l,x) :  
2     n=len(l)  
3     v=False  
4     for i in range(n) :  
5         if l[i]==x :  
6             v=True  
7     return v
```

- Recherche du maximum d'une liste de nombres l :

```
1 def max(l) :  
2     n=len(l)  
3     m=l[0]  
4     for i in range(n) :  
5         if m<l[i] :  
6             m=l[i]  
7     return m
```