

## TP 2 : Fonctions et Listes

### Definition : Les listes en python

En langage Python, une liste est (comme son nom l'indique) une liste d'éléments ordonnés. On peut définir plusieurs listes en notant par exemple :

```
L1=[1,2,4,3,7]
L2=[1,'a',True]
L3=[2*i+1 for i in range(5)]
L4=[i for i in range(1,10)]
L4bis=list(range(1,10))
```

Les listes peuvent donc contenir des objets de type différents. On notera que les liste L4 et L4bis sont égales.

On peut manipuler les éléments d'une liste en notant :

```
L1[k]          # renvoie le k+1-ième élément de la liste L1
L1[-k]         # renvoie le k-ième élément de la liste L1 en partant de la fin
L1[i:j]        # renvoie la portion de liste de la position i jusqu'à la position j-1
L1[i:j:k]       # renvoie la liste précédente en prenant un élément sur k
L1+L2          # renvoie la liste contenant les éléments de L1 puis les éléments de L2
2*L1           # équivalent à L1+L1
```

Si on souhaite sauvegarder une liste on doit écrire :

```
L2=list(L1) # fait une copie de L1 qui s'appelle L2 indépendante de L1
```

On peut maintenant modifier L1 sans changer L2

```
L2=L1        # créer un clone de L1 qui s'appelle L2
```

Si on modifie un élément de L1, le même élément sera modifié dans L2.

### Exemple :

```
L=[0,2,4,6,8,10]
L1=[0:2]          >>>>L1=[0,2]
L2=[:2]           >>>>L2=[0,2]
L3=[1:-1]         >>>>L3=[2,4,6,8]
L4=[1:]           >>>>L4=[2,4,6,8,10]
L5=[::2]          >>>>L5=[2,6,10]
```

## Exercice 1 : Modification d'une liste

- Q.1** Écrire une fonction **sans\_quatre(L)** qui renvoie la liste  $L$  en remplaçant tous les 4 par des 0.
- Q.2** Écrire une fonction **rotation(L)** qui renvoie la liste  $L$  en décalant tous les éléments vers la droite (le dernier devient le premier, le premier devient le second, etc).
- Q.3** Écrire une fonction **inverser(L)** qui renvoie la liste  $L$  en inversant l'ordre de tous les éléments (le dernier devient le premier, l'avant dernier devient le second, etc).
- Q.4** Écrire une fonction **Supprimer\_rang(L,i)** qui renvoie la liste  $L$  en supprimant l'élément à la position  $i$ .
- Q.5** Écrire une fonction **Supprimer\_element(L,a)** qui renvoie la liste  $L$  en supprimant tous les éléments qui sont égaux à  $a$ .

## Exercice 2 : Génération par compréhension

On considère la génération d'une liste avec un programme de la forme :

```
def f(u):
    return .....
Objet=[. ....]
L=[f(u) for u in Objet]
```

Compléter les ..... dans le programme ci-dessus pour générer les listes suivantes :

- Q.1**  $L=[2,4,8,10,12,14,16,20]$

**Q.2** `L=['mon chat','ton chat','son chat','leur chat','le chat','ce chat']`

**Q.3** `L=[0,1,0,1,0,1,0,1]`

### Exercice 3 : Recherche de maximum

**Q.1** Écrire une fonction **indices\_max(L)** qui reçoit comme paramètre une liste d'entiers *L*, et qui retourne la plus grande valeur de cette liste et une liste contenant les indices de tous les éléments égaux au maximum.

**Q.2** Écrire une fonction **second\_max(L)** qui reçoit comme paramètre une liste d'entiers *L*, et qui retourne la seconde plus grande valeur de cette liste.

### Exercice 4.

Rédiger un programme `compteurVoyelle.py` constitué des deux fonctions suivantes :

1. Une fonction `testVoyelle`, dont le paramètre d'entrée est un caractère, qui renvoie `True` si le caractère est une voyelle et `False` sinon.

2. Une fonction `comptVoyelle`, dont le paramètre d'entrée est une chaîne de caractères, qui renvoie le nombre de caractères qui sont des voyelles. Indication : une variable *k* parcourt la chaîne de caractère reçue en argument; un compteur s'incrémente à chaque fois que *k* tombe sur une voyelle : le test est réalisé à l'aide de la fonction précédente.

Votre programme demande à l'utilisateur de saisir une chaîne de caractères et lui répond le nombre de voyelles contenues dans la chaîne.