

TP Informatique 7 : 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
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.

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 : Calcul d'une suite

On considère la suite u définie par $u_0 = 0$, et pour tout entier n : $u_{n+1} = 2 * u_n + 1$

- Q.1** Écrire une fonction **suite(n)** qui calcule tous les éléments de la suite jusqu'à u_n et les stocke dans une liste L tel que $L[k] = u_k$.
- Q.2** Écrire une fonction **Somme_suite(N)** qui calcule la somme de tous les éléments de tous les éléments de la suite jusqu'à u_n :

$$\sum_{n=0}^N u_n$$

Exercice 5 : Charge d'un condensateur On considère la charge d'un condensateur dans un circuit RC série tel que :

$$u_C(t) = E \left(1 - e^{-\frac{t}{\tau}} \right)$$

- Q.1** Écrire une fonction **Temps(Tf,N)** qui crée une liste T composée de N éléments tels que :
- $L[0] = 0$ et $L[N - 1] = Tf$;
 - $L[k] - L[k - 1] = C^{te} \forall k < N$, constante à déterminer.
- Q.2** Écrire une fonction **Charge(E,tau,Tf,N)** qui crée une liste U composée de N éléments tels que : $U[k] = u_C(Tf[k])$

Exercice 6 : Primalité (BONUS)

Pour cet exercice on pourra reprendre la fonction **TestP(n)** du TP5.

- Q.1** Écrire une fonction **facteurs_premiers(n)** qui renvoie la liste de tous les facteurs premiers d'un entier $n \geq 2$.
- Q.2** Écrire une fonction **liste_premiers(n)** qui renvoie la liste L de tous les nombres premiers strictements inférieurs à n .