

TP 3 : Listes et chaînes de caractères

I Listes

I.1 Exemples de listes de nombres

Exercice 1

Essayez :

```
>>> [1, 2] + 3          >>> 4 * [1, 2, 3]
>>> [1, 2] + [2, 3]    >>> [1, 2] * 4
>>> [1, 2] - [2, 1]    >>> [2] * [3]
>>> [1, 2, 3] + [1, 2] >>> [2, 3] * [2, 4]
>>> [1, 2, 3] * 4
```

Exercice 2

Créer la liste de manière rapide (sans recopier toutes les valeurs) : `[1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]`.

Exercice 3

Essayez

```
>>> L = [1, 2, 3, 4, 5]   >>> G = [2, 4, 6, 8, 10]   >>> H = [-3, 3, 0]
>>> L[0]                 >>> G[0]                 >>> H[0]
>>> L[1]                 >>> G[1]                 >>> H[1]
>>> L[4]                 >>> G[4]                 >>> H[2]
>>> len(L)               >>> G[len(G)]           >>> H[len(H) - 1]
```

Exercice 4

Pour placer la valeur de `a` dans la cellule `L[0]`, on écrit `L[0]=a`. Transformer la liste `G` ci-dessus, en la liste `[2, -4, 6, -8, 10]`.

I.2 Définition et opérations élémentaires

Exercice 5 1. Créer une liste `L` contenant les chaînes "Cerisier", "Pommier", "Poirier", "Prunier".

2. Tester les commandes et écrire le résultat de chacune :

```
>>> type(L)              >>> L[0 : 2]
>>> type(L[0])          >>> L[0 : 3]
>>> len(L)              >>> L[-1]
>>> len(L[0])          >>> L.append("Abricotier")
>>> L[1]                >>> len(L)
>>> L[0]                >>> L[len(L)-1]
```

3. Créer une liste `N` contenant tous les entiers de 1 à 10 et tester :

```
>>> N[len(N)-1]         >>> N[0 : 6 : 2]
>>> N[len(N)-2]         >>> N[0 : 6 : 3]
>>> N[:3]
```

I.3 Modifier des listes

Exercice 6 1. Créer une liste `L` composée de 5 zéros.

2. Créer une liste `M` composée des nombres entiers de 1 à 5.

3. Tester :

```

>>> L.extend(M)
>>> print(L)
>>> L.count(0)
>>> L.count(1)
>>> L.remove(1)

>>> print(L)
>>> del L[0]
>>> print(L)
>>> L.remove(0)
>>> print(L)

```

Exercice 7

Ecrire une fonction qui agit comme `.count` : à partir d'une liste donnée L compte le nombre d'occurrences d'une valeur a donnée.

I.4 Copier une liste

Exercice 8

```

>>> L = [1, 2, 3]
>>> M = L
>>> del M[len(M)-1]
>>> L

>>> L = [1, 2, 3]
>>> M = L
>>> M[0] = 4
>>> L

```

Pour copier une liste indépendante, on utilise la commande : $M = L[:]$ ou $M = L.copy()$

Essayez Les commandes ci-dessus en copiant L de manière indépendante.

I.5 Afficher graphiquement une liste

Exercice 9

Pour afficher des graphiques, on a besoin de la bibliothèque `matplotlib.pyplot`.

```
import matplotlib.pyplot as plt
```

```

L = [1, 3, 2, 5]
plt.plot(L) # dessine la liste
plt.show() # affiche la fenêtre graphique

```

On peut faire un peu mieux. Commentez les lignes suivantes :

```

L = [1, 3, 2, 5]
x = [0, 2, 4, 6] # la liste des abscisses
plt.plot(x, L, 'r--', label = 'liste')
plt.legend()
plt.show()

```

I.6 Boucles **for** sur le **range**

Pour générer une liste de nombres réguliers `range([start], stop, [step])` donne une liste d'entiers de $[start]$ (ou 0) inclus à $stop$ (exclu) de pas $[step]$ (ou 1)

Remarque : les nombres entre crochets `[]` sont facultatifs. Seul le `stop` est obligatoire.

```
for i in range(4, 10):  
    print(i)
```

```
for k in range(4, 10, 2):  
    print(k)
```

```
for d in range(0, -10, -3):  
    print(d)
```

Exercice 10

Ecrire un algorithme qui prend en argument une liste de nombre L et qui renvoie la somme de tous les éléments de L .

Exercice 11

Ecrire un algorithme qui à partir de deux listes L et M de même taille et contenant des entiers, renvoie une liste K qui contient la somme des nombres de L et de M .

Exercice 12

Ecrire un algorithme qui à partir d'une liste L contenant des nombres, affiche la courbe des nombres au carré.

II Exercices

II.1 Moyenne

Exercice 13 *Moyenne*

Ecrire un algorithme qui, à partir d'une liste L de nombres, affiche la moyenne de la liste L .

Ecrire un algorithme qui, à partir d'une liste $coeff$ contenant des coefficients, affiche la moyenne de la liste L pondérée par les coefficients de $coeff$.