Qu’est-ce qu’une liste ?

Une liste est une série d’éléments, de valeurs qui peut être de tous types (string, int, float, etc.).

Un exemple pour illustrer ce propos :

animaux = ["girafe", "tigre", "singe", "souris"]

tailles = [5, 2.5, 1.75, 0.15]

liste = ["girafe", 5, 0.5]

Lister les fonctions utiles pour manipuler les listes

* split()
* list[index]
* index()
* indexlist()
* range()
* count()
* les boucles
* max\_consecutive()
* reverse()
* append()
* remove()
* pop()
* extend()
* sort()
* sotred()
* len()
* del()

Comment parcourir une liste ?

On peut utiliser l’index de la liste en question pour parcourir la liste. Par exemple, dans cette liste appelée suit\_of\_numbers : [1, 2, 3, 4, 5].

suit\_of\_numbers[0] est le premier élément de la liste, soit 1.

Explorer les listes de compréhension

Une list comprehension est un code optimisé par le développeur. Le but est de provoquer le même résultat mais de manière plus simplifiée, plus rapide et donc moins lourde pour la mémoire de l’ordinateur. L’exemple le plus connu est la création d’une boucle :

Pour :

a = [1,4,2,7,1,9,0,3,4,6,6,6,8,3]

Soit :

b = []

for x in a:

if x > 5:

b.append(x)

Donne le même résultat que :

[x for x in a if x > 5]

List vs Tupples

Une liste[] est une suite ordonnée d’éléments, tandis qu’un tuple() est une suite non ordonnée d’éléments. Cela signifie qu’à l’inverse d’un tuple(), nous pouvons parcourir une liste étant donné que les éléments qui la composent ont un indice mesurable.