



## Les listes

```
Fibo = [1,1,2,3,5,8,13]
```

Une liste est un ensemble d'éléments indexés. La liste Fibo ci-dessus contient les sept premiers termes de la suite de Fibonacci.

Le langage Python permet de nombreuses opérations sur les listes.

### Exemple :

```
g=9.81                # Unités SI
h=320
altitudes=[]          # création de la liste vide altitudes

def z(t):              # définition de la loi horaire
    return -0.5*g*t**2+h

for t in range(9):     # t parcourt les entiers de 0 à 8
    altitudes.append(z(t)) # à chaque passage, on ajoute l'altitude à la liste

altitudes              # affichage de la liste altitudes
```

Ce programme affiche la liste des altitudes pour  $t$  allant de 0 à 8.

### Applications :

1. Écrire un programme qui affiche la liste de tous les multiples de 3 ou 5 strictement inférieurs à 100.
2. Écrire une fonction qui renvoie une liste de 20 nombres entiers pseudo-aléatoires compris entre 1 et 10.
3. Modifier cette fonction pour qu'elle renvoie la liste dans l'ordre croissant.

### Boîte à outils

On considère la liste  $L = [14, 11, 5, 20, 11, 11]$

$L[2]$  renvoie l'élément de rang 2 de la liste,  $L[2]=5$ ,  $L[0]=14$ .

$L.append(33)$  ajoute 33 en fin de liste :  $[14, 11, 5, 20, 11, 11, 33]$ .

$L.remove(11)$  supprime **la première** occurrence de 11 dans la liste.

$L.insert(2, 47)$  insère la valeur 47 au rang 2 :  $[14, 5, 47, 20, 11, 11, 33]$

$L.count(11)$  renvoie 2, c'est le nombre d'occurrences de l'élément 11.

$len(L)$  renvoie 7, c'est la longueur de la liste.

$L.sort()$  renvoie la liste dans l'ordre croissant :  $[5, 11, 11, 14, 20, 33, 47]$ .

$del L[3]$  supprime la valeur de rang 3 :  $[5, 11, 11, 20, 33, 47]$ .

$[4, 9] + [7, 6, 0]$  renvoie  $[4, 9, 7, 6, 0]$ .