

# List functions

January 30, 2025

## 0.1 List

```
[1]: l = [10,20,20,30,51,65,85]
```

```
[2]: l
```

```
[2]: [10, 20, 20, 30, 51, 65, 85]
```

## 0.2 Append function

```
[3]: l
```

```
[3]: [10, 20, 20, 30, 51, 65, 85]
```

```
[4]: l.append(25) #this append function add any value in the end of a list
```

```
[5]: l
```

```
[5]: [10, 20, 20, 30, 51, 65, 85, 25]
```

```
[6]: l.append([1,2,3]) #we can also add another list into list
```

```
[7]: l
```

```
[7]: [10, 20, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

## 0.3 copy function

```
[8]: l1 = l.copy() #this function copies the list to another variable
```

```
[9]: l1
```

```
[9]: [10, 20, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

```
[10]: l1 == l
```

```
[10]: True
```

## 0.4 Remove function

```
[11]: l1.remove(20)    # this function removes the value that is passed and only first  
      ↪ occurrence of the value is removed
```

```
[12]: 11
```

```
[12]: [10, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

## 1 List Indexing

```
[13]: s = ['apple', 'banana', 3.0, 45]
```

```
[14]: s[0] #we can fetch the value by index numbers
```

```
[14]: 'apple'
```

```
[15]: s[3]
```

```
[15]: 45
```

### 1.1 list SLicing

```
[16]: s1 = s[0:2] # we can set the range to slice and store in another variable
```

```
[17]: s1
```

```
[17]: ['apple', 'banana']
```

```
[18]: s[:] # to access all elements of list we can write like this
```

```
[18]: ['apple', 'banana', 3.0, 45]
```

```
[19]: s[2:] #to access all element from a particular we can write like this to fetch
```

```
[19]: [3.0, 45]
```

## 2 Step indexing

```
[20]: s[::2] # to fetch elements by steps we use two colons last value represent steps
```

```
[20]: ['apple', 3.0]
```

```
[21]: s[::3]
```

```
[21]: ['apple', 45]
```

```
[22]: s[::-1] #another way to reverse the list
```

```
[22]: [45, 3.0, 'banana', 'apple']
```

### 3 Reverse function

```
[24]: s.reverse() #this functions reverse the elements of the list
```

```
[25]: s
```

```
[25]: [45, 3.0, 'banana', 'apple']
```

#### 3.1 Clear function

```
[27]: s.clear() # this functions clears the elements in the list
```

```
[28]: s
```

```
[28]: []
```

```
[29]: ## Count function
```

```
[30]: l
```

```
[30]: [10, 20, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

```
[31]: l.count(20) #this function counts the occurence of value that is passed
```

```
[31]: 2
```

```
[32]: l.count(51)
```

```
[32]: 1
```

#### 3.2 Index Function

```
[33]: l.index(51) #this function returns the index of first occurence the given vaalue
```

```
[33]: 4
```

```
[34]: l.index(25)
```

```
[34]: 7
```

```
[35]: l.index(20)
```

```
[35]: 1
```

### 3.3 Insert Function

```
[36]: 1
```

```
[36]: [10, 20, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

```
[37]: 1.insert(0,55)  #this function inser value on a particular index (index,value)
```

```
[38]: 1
```

```
[38]: [55, 10, 20, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

```
[39]: 1.insert(1,44)
```

```
[40]: 1
```

```
[40]: [55, 44, 10, 20, 20, 30, 51, 65, 85, 25, [1, 2, 3]]
```

### 3.4 Pop Function

```
[41]: 1.pop() #this function pop the last index value if no arguements are passed or_
      ↪pop value of the passed as index
```

```
[41]: [1, 2, 3]
```

```
[42]: 1
```

```
[42]: [55, 44, 10, 20, 20, 30, 51, 65, 85, 25]
```

```
[44]: 1.pop(2)
```

```
[44]: 10
```

### 3.5 Remove Function

```
[48]: 1.remove(20) # Remove function removes the first occurence of value that is_
      ↪passed as arguements
```

```
[49]: 1
```

```
[49]: [55, 44, 20, 30, 51, 65, 85, 25]
```

```
[50]: 1.remove(51)
```

```
[51]: 1
```

```
[51]: [55, 44, 20, 30, 65, 85, 25]
```

### 3.6 Sort Function

```
[53]: l.sort() # This function sort the list in ascending order
```

```
[54]: 1
```

```
[54]: [20, 25, 30, 44, 55, 65, 85]
```

```
[58]: l.sort(reverse =True) # if reverse is true then this will sort in descending  
↪order
```

```
[59]: 1
```

```
[59]: [85, 65, 55, 44, 30, 25, 20]
```

```
[ ]:
```