

Lists

In Python the lists can contain a combination of data types such as strings, tuples, lists , dictionaries, functions, file objects, and any type of number. ->Lists are mutable ->lists can be accessed by indexing . ->Lists are represented in [] ->Lists are slower than tuples.

Extend

```
In [31]: # extend = extends the new elements in the new list
l = [1,2,3]
l.extend([4,5])
print(l)
```

```
[1, 2, 3, 4, 5]
```

Insert

```
In [7]: # inserting the elements in a specific position
x = [1,2]
x.insert(2,"y")
print(x)
```

```
[1, 2, 'y']
```

Delete

```
In [12]: # delete specific element from specific position
x = [1,2,3]
del x [0]
print(x)
```

```
[2, 3]
```

Remove

```
In [15]: # removes the first match element from a list
y = [1,2,3,'a',4,5,'a']
y.remove('a')
print(y)
```

```
[1, 2, 3, 4, 5, 'a']
```

Reverse

```
In [16]: # reverse the order of elements in the list this places the final elements at the first and initial elements in the first stage
x = [1,2,3,4,5]
x.reverse()
print(x)
```

```
[5, 4, 3, 2, 1]
```

Sort

```
In [17]: # sort elements of the list from smallest to largest this behaviour can be modified using parameters reverse = True
x = [1,2,3,4,5,6]
x.sort()
print(x)
```

```
[1, 2, 3, 4, 5, 6]
```

Sort using parameter "Reverse "

```
In [18]: #sorting parameters = True
x = [1,2,3,4,5,6]
x.sort(reverse = True)
print(x)
```

```
[6, 5, 4, 3, 2, 1]
```

Sorting

```
In [24]: def sorting_my_length(str):
          return len(str)

x = ['Python', 'is', 'the', 'best']

x.sort(key = sorting_my_length)

print(x)
```

```
['is', 'the', 'best', 'Python']
```

Concatenate or Join

```
In [26]: # concatenate two lists join two lists using + operator
list1 = [1,2,3]
list2 = [4,5,6]
concat = (list1,list2) # or list3 = (list1 + list2)
print(concat)          # print(list3)
```

```
([1, 2, 3], [4, 5, 6])
```

* Repeating a list

```
In [29]: # * this operation will replace a list to the specified number of times
x = [1,2,3]
print(x * 3)
```

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

Minimum

```
In [30]: # min this method will return the smallest element in the list
x = [1,2,3,4,5,6]
print(min(x))
```

```
1
```

Maximum

```
In [32]: # max this returns the largest elements in the list
x = [1,10,2,4,65,2]
print(max(x))
```

```
65
```

Incrementation of List

```
In [42]: list = []

# Using List comprehension
[print(i) for i in range(0,9)]
```

```
0
1
2
3
4
5
6
7
8
```

```
Out[42]: [None, None, None, None, None, None, None, None, None]
```

Finding out the maximum element in the given nested list

```
In [9]: list2 = []
def get_max(list1):
    for i in list1:
        if type(i) == list:
            get_max(i)
        else:
            list2.append(i)
    return max(list2)

list1 = [1,2,3,[45,95],[100,20,30,30,40,50]]
print(get_max(list1))
```

100

Finding out the minimum element in the given nested list

```
In [10]: #
list2 = []
def get_max(list1):
    for i in list1:
        if type(i) == list:
            get_max(i)
        else:
            list2.append(i)
    return min(list2)

list1 = [1,2,3,[45,95],[100,20,30,30,40,50]]
print(get_max(list1))
```

1

Appending List

```
In [13]: l = ["Shankar", "Vayusena", "Ramaraj", "Vamshi"]
v = ['Naidu']
v.append(l)
print(v)
```

['Naidu', ['Shankar', 'Vayusena', 'Ramaraj', 'Vamshi']]

```
In [15]: l = ["Shankar", "Vayusena", "Ramaraj", "Vamshi"]
l.append('Naidu')
#v.append(l)
print(l)
```

['Shankar', 'Vayusena', 'Ramaraj', 'Vamshi', 'Naidu']

In []:

