

How to Access elements from a list?

There are various ways in which we can access the elements of a list.

List Index

We can use the index operator [] to access an item in a list. Index starts from 0. So, a list having 5 elements will have index from 0 to 4. The index must be an integer. Nested list are accessed using nested indexing.

NOTE:

We can't use float or other types as index, this will result into TypeError.

Example:

```
PyList=["Big Data", "Hadoop", "Spark", "IoT"]
Item           Big Data      Hadoop   Spark    IoT
Index (from left)     0          1          2          3
Index (from right)    -4         -3         -2         -1
```

Example:

```
PyList = ['P', 'Y', 'T', 'H', 'O', 'N']
print(PyList[0])
print(PyList[4])
print(PyList[-1])
print(PyList[-4])
```

Negative indexing

Python allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last item and so on.

Example:

```
PyList=["Big Data", "Hadoop", "Spark", "IoT"]
print(PyList[0])
print(PyList[0],PyList[3])
print(PyList[-1])
print(PyList[4])
```

Example:

```
MyList=["PYTHON",1,2,3,["Machine"]]
print(MyList)
print(MyList[1])
print(MyList[-2])
print(MyList[-1][-1][-1])
print(MyList[-1][-1][-3])
print(MyList[-2])
print(MyList[-5][-1])
```

Example:

```
PyList=[1,2,3,"HELLO",5,6,["PYTHON"]]
print(PyList)
print(PyList[6])
print(PyList[-6])
print(PyList[-2])
print(PyList[3])
print(PyList[6][0])
```

```
print(PyList[-1])
print(PyList[-1][-1])
print(PyList[-1][-1][-1])
print(PyList[-1][-1][-1][0])
print(PyList[-1][-1][-1][-1])
print(PyList[3][0])
print(PyList[3][-3])
```

List Slices

Lists can be sliced like strings and other sequences. We can access a range of items in a list by using the slicing operator (colon).

Syntax :

```
sliced_list = List_Name[StartIndex:EndIndex]
```

Example:

```
PyList=["Big Data", "Hadoop", "Spark", "IoT"]
print(PyList[0:2])
print(PyList[1:2])
print(PyList[1:-2])
print(PyList[:3])
print(PyList[:])
```

How to use start,stop,step:

Example:

```
PyList=["Big Data", "Hadoop", "Spark", "IoT"]
#Start:Stop
print(PyList[1:3])
#Start:Stop:Step(Increment)
print(PyList[1:3:1])
#Start:Stop:Step(Increment)
print(PyList[1:4:2])#Alternate, Hadoop, IoT
#Start:Stop:Step(Increment)
print(PyList[1:4:3])#TwoElements, Hadoop
#Start:Stop:Step(decrement)
print(PyList[3:1:-1])#IoT, SPARK
#Start:Stop:Step(decrement)
print(PyList[3:1:-2])#IoT
```

Example:

```
listx=[1, 5, 7, 3, 2, 4, 6]
print(listx)
sublist=listx[2:7:2]
print(sublist)
sublist=listx[6:2:-1]
print(sublist)
```

Example:

```
listx=[1, 5, 7, 3, 2, 4, 6, 10, 11]
print(listx[7:1:-3])
print(listx[2:7:2])
print(listx[2:7])
print(listx[:4])
print(listx[3:])
```

Example:

```
PyList=[1,3,4,6,7,8,9]
```

```
print(PyList)
print(PyList[1:5])
print(PyList[1:5:1])
print(PyList[1:5:2])
print(PyList[1:5:3])
print(PyList[1:5:4])
print(PyList[5:1:-1])
print(PyList[5:1:-2])
print(PyList[5:1:-3])
print(PyList[6:0:-4])
```

Double Slice Operator:

```
PyList=[1,2,3,4,5,6,7,8,9,10]
print(PyList)
print(PyList[::-1])
print(PyList[::-2])
print(PyList[::-3])
print(PyList[::-4])
print(PyList[::-1])
print(PyList[::-2])
print(PyList[::-3])
print(PyList[::-4])
```

Example:

```
PyList=[1,2,3,4,5,6,7,8,9,10]
print(PyList[1::5])
print(PyList[0::6])
print(PyList[5::9])
print(PyList[0::9])
print(PyList[6::11])
```

Example:

```
PyStr="Malayalam"
if PyStr==PyStr[::-1]:
    print("Yes, This is Palindrome")
else:
    print("NO, This is not Palindrome")
```

Example:

```
PyStr="malayalam"
if PyStr==PyStr[::-1]:
    print("Yes, This is Palindrome")
else:
    print("NO, This is not Palindrome")
```

Python List Methods

In Python Scripting The following list of methods existed in List object.

1 append()	2 extend()
3 insert()	4 remove()
5 pop()	6 clear()
7 index()	8 count()
9 sort()	10 reverse()
11 copy()	

append():Add an element to the end of the list

Syntax:

```
list.append(item)
```

Example:

```
py_list=["Big Data", "Hadoop", "Spark", "IoT"]
print(py_list)
py_list.append("PYTHON")
print(py_list)
```

Example:

```
alist = ['a', 'b', 'c']
alist.append(['d', 'e', 'f'])
print(alist)
```

NOTE: We can Modify an element by using the index of the element

Example:

```
py_list=["Big Data", "Hadoop", "Spark", "IoT"]
print(py_list[0])
py_list[0]="DataScience"
print(py_list)
print(py_list[0])
```

Example:

```
py_list=["Big Data", "Hadoop", "Spark", "IoT"]
print(py_list)
py_list[2]="PYTHON"
print(py_list)
```

Example:

```
PyList=["BigData"]
print(PyList)
PyList.append(("ML", "BC"))
print(PyList)
PyList.append({"AI"})
print(PyList)
PyList.append({1:"PYTHON"})
print(PyList)
```

NOTE: A list can append with any Data Structure..!!

extend(): Add all elements of a list to the another list

Syntax:

```
list.extend(iterable)
```

Example:

```
alist1 = ['a', 'b', 'c']
alist1.extend(['d', 'e', 'f'])
print(alist1)
```

```
# Appending two lists
```

```
Countries = ["India", "Pakistan", "Sri Lanka"]
African_Countries = ["Egypt", "Kenya", "Namibia", "Zimbabwe"]
Countries.extend(African_Countries)
print(Countries)
```