Lists In Python

Python lists are containers used to store a list of values of any data type. In simple words, we can say that a list is a collection of elements from any data type.

In Python, lists are mutable i.e., Python will not create a new list if we modify an element of the list.

It works as a container that holds other objects in each order. We can perform various operations like insertion and deletion on list.

A list can be composed by storing a sequence of different type of values separated by commas. Python list is enclosed between square [] brackets and elements are stored in the index basis with starting index 0.

How to create a list

```
a = [45, 33, 22, 12]
b = ["Anirudh", 67, 89.99, -100, "Hello"]
c = []
print(a)
print(b)
print(c)
print(type(a))
print(type(b))
print(type(c))
```

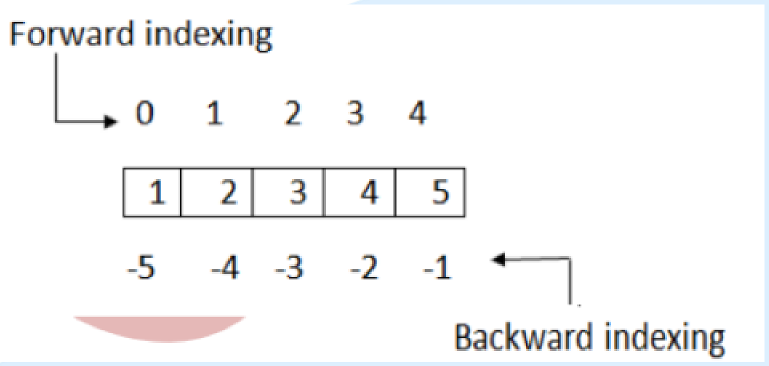
Output

```
[45, 33, 22, 12]
['Anirudh', 67, 89.99, -100, 'Hello']
[]
<class 'list'>
<class 'list'>
<class 'list'>
```

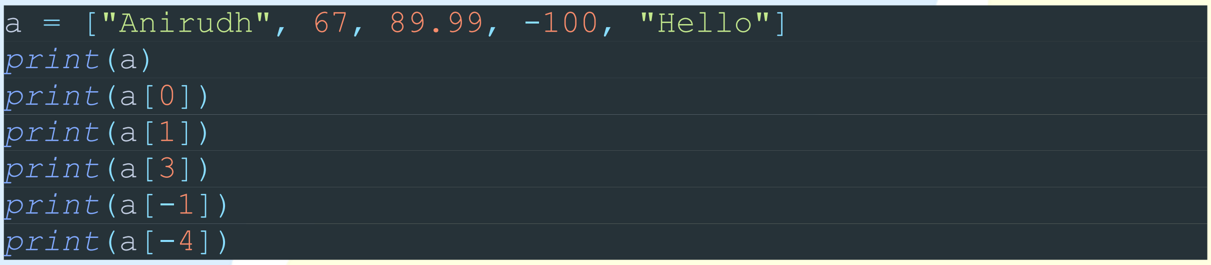
Elements in a List

Following is the pictorial representation of a list. We can see that it allows to access elements from both end (forward and backward).

data = [1, 2, 3, 4, 5]

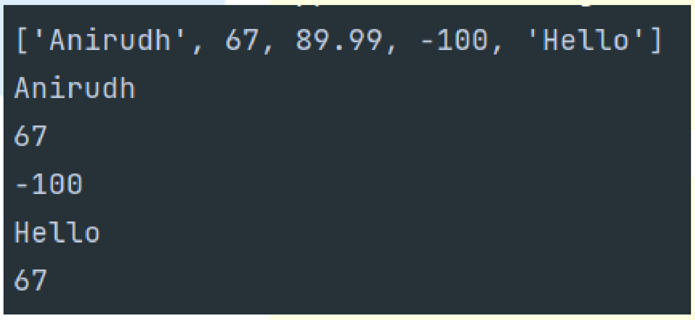


Accessing elements in a List



```python
a = ["Anirudh", 67, 89.99, -100, "Hello"]
print(a)
print(a[0])
print(a[1])
print(a[3])
print(a[-1])
print(a[-4])
```

Output



```
['Anirudh', 67, 89.99, -100, 'Hello']
Anirudh
67
-100
Hello
67
```

Python List Operations

Apart from creating and accessing elements from the list, Python allows us to perform various other operations on the list. Some common operations are given below: -

1) Adding Python Lists

In Python, lists can be added by using the concatenation operator (+) to join two lists.

```
list1 = [100, 200, 1]
list2 = ["plane", 44, "python"]
list3 = list1 + list2
print(list3)
```

Output

```
[100, 200, 1, 'plane', 44, 'python']
```

2) Replicating Python Lists

Replicating means repeating, It can be performed by using '*' operator by a specific number of time.

```
list1 = ["plane", 44, "python"]
list2 = list1 * 4
print(list2)
```

```
['plane', 44, 'python', 'plane', 44, 'python', 'plane', 44, 'python', 'plane', 44, 'python']
```

3) Slicing Python Lists

A subpart of a list can be retrieved based on index. This subpart is known as list slice. This feature allows us to get sub-list of specified start and end index.

```
list1 = [54, 14, 47, 89, 63, 21, 44, 28]
print(list1)
print(list1[4])
print(list1[0:5])
print(list1[2:6])
print(list1[-1:-3:-1])
print(list1[0:6:2])
```

```
[54, 14, 47, 89, 63, 21, 44, 28]
63
[54, 14, 47, 89, 63]
[47, 89, 63, 21]
[28, 44]
[54, 47, 63]
```

4) Updating List

To update or change the value of index of a list, assign the value to that index of the List.

```
data1 = [56, "india", "python", 45, 12, 22]
print(data1)
data1[1] = "india is great"
print(data1)
data1[3] = 99
data1[-1] = 100
print(data1)
```

Output

```
[56, 'india', 'python', 45, 12, 22]
[56, 'india is great', 'python', 45, 12, 22]
[56, 'india is great', 'python', 99, 12, 100]
```

5) Appending elements to Python Lists

Python provides, append() method which is used to append i.e., add an element at the end of the existing elements.

```
data1 = [56, "india", "python"]
print(data1)
data1.append("hello")
print(data1)
data1.append(55)
print(data1)
```

Output

```
[56, 'india', 'python']
[56, 'india', 'python', 'hello']
[56, 'india', 'python', 'hello', 55]
```

6) **Delete elements**

In Python, del statement can be used to delete an element from the list. It can also be used to delete all items from **startIndex** to **endIndex**.

```python
data1 = ["india", "python", 54, 12, 78, "australia"]
print(data1)
del data1[0]
print(data1)
del data1[1:4]
print(data1)
```

Output

```
['india', 'python', 54, 12, 78, 'australia']
['python', 54, 12, 78, 'australia']
['python', 'australia']
```

Lists more built-in methods

Python provides various Built-in functions and methods for Lists that we can apply on the list.

Following are the common list functions.

Function	Description
min(list)	Returns the minimum value in a list.
max(list)	Returns the maximum value in a list.
len(list)	Returns the length of a list.
sum(list)	Returns the total sum of all values in the list

1) **Python list min() method**

This method is used to get min value from the list.

```python
list1 = [56, 43, 212, 17, 891, 90, 94, 34]
print("Minimum value in List 1 = %d" % min(list1))

# Error because we can't compare STRING and INT
list2 = [45, 67, "Python", "India", 1]
print("Minimum value in List 2 = %d" % min(list2))
```

```
Minimum value in List 1 = 17
Traceback (most recent call last):
  File "C:\Users\aniru\Desktop\Python Programming\hello.py", line 6, in <module>
    print("Minimum value in List 2 = %d" % (min(list2)))
TypeError: '<' not supported between instances of 'str' and 'int'
```

2) Python list max() method

This method is used to get max value from the list.

```
list1 = [56, 43, 212, 17, 891, 90, 94, 34]
print("Maximum value in List 1 = %d" % max(list1))
```

Output

```
Maximum value in List 1 = 891
```

3) Python list len() method

This method is used to get total length of list.

```
list1 = [56, 43, 212, 17, 891, 90, 94, 34]
print("Length of the List 1 = %d" % len(list1))
```

Output

```
Length of the List 1 = 8
```

4) Python list sum() method

Returns the sum of all the elements in the list.

```
list1 = [56, 43, 212, 17, 891, 90, 94, 34]
a = sum(list1)
print("Sum of the List 1 = %d" % a)

list2 = [55, 66, 77, 88, 99]
print("Sum of the List 2 = %d" % sum(list2))
```

Output

```
Sum of the List 1 = 1437
Sum of the List 2 = 385
```

More Python list built in methods

Function	Description
index(object)	It returns the index value of the object.
count(object)	It returns the number of times an object is repeated in list.
pop()/pop(index)	It returns the last object or the specified index object. It removes the popped object.
insert(index,object)	It inserts an object at the index position.
extend(sequence)	It adds the sequence to an existing list.
remove(object)	It removes the object from the given list.
reverse()	Reverses the position of all elements of a list.
sort()	Sort the elements of the list.

1) Python list index() method

```
list1 = ["python", 56, 123, "india", "hello", 56]
print("Index of 56 = %d" % (list1.index(56)))
print("Index of hello = %d" % (list1.index("hello")))

# Will give us error because 4545 is not in List
print("Index of 4545 = %d" % (list1.index(4545)))
```

Output

```
Index of 56 = 1
Index of hello = 4
```

2) Python list count() method

```
list1 = ["python", 56, 123, "india", "hello", 56, 56]
print("56 has occurred %d times." % (list1.count(56)))
print("python has occurred %d times." %
(list1.count("python")))
print("4545 has occurred %d times." % (list1.count(4545)))
```

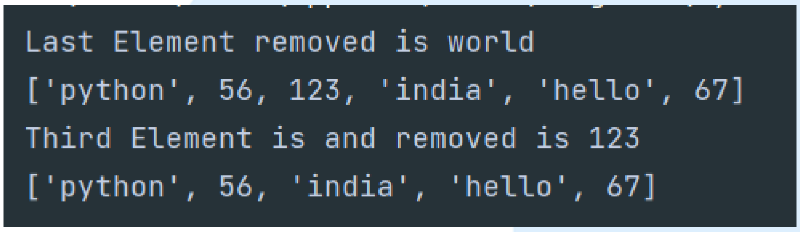
Output

```
56 has occurred 3 times.
python has occurred 1 times.
4545 has occurred 0 times.
```

3) Python list pop()/pop(index) method

```
list1 = ["python", 56, 123, "india", "hello", 67, "world"]
print("Last Element removed is", list1.pop())
print(list1)
print("Third Element is and removed is", list1.pop(2))
print(list1)
```

Output



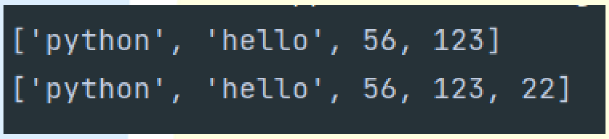
```
Last Element removed is world
['python', 56, 123, 'india', 'hello', 67]
Third Element is and removed is 123
['python', 56, 'india', 'hello', 67]
```

4) Python list insert(index,object) method

```
list1 = ["python", 56, 123]
var = "hello"
list1.insert(1, var)
print(list1)

# Adds element to last position if Position does not exists
list1.insert(10, 22)
print(list1)
```

Output

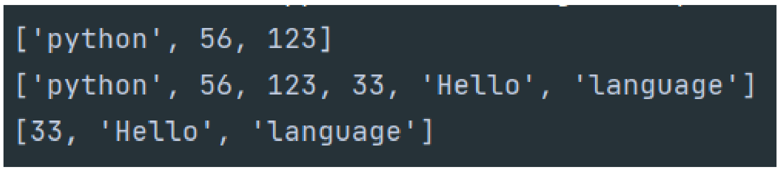


```
['python', 'hello', 56, 123]
['python', 'hello', 56, 123, 22]
```

5) Python list extend(sequence) method

```
list1 = ["python", 56, 123]
list2 = [33, "Hello", "language"]
print(list1)
list1.extend(list2)
print(list1)
print(list2)
```

Output



```
['python', 56, 123]
['python', 56, 123, 33, 'Hello', 'language']
[33, 'Hello', 'language']
```

6) Python list remove(object) method

```
data1 = ['abc', 123, 10.5, 'a', 'xyz']
print(data1)
data1.remove(123)
print(data1)

# This will give us error because "python" is not in list
data1.remove(100)
```

Output

```
['abc', 123, 10.5, 'a', 'xyz']
['abc', 10.5, 'a', 'xyz']
Traceback (most recent call last):
  File "C:\Users\aniru\Desktop\Python Programming\hello.py", line 7, in <module>
    data1.remove(100)
ValueError: list.remove(x): x not in list
```

7) Python list reverse() method

```
data1 = ['abc', 123, 10.5, 'a', 'xyz']
print(data1)

data1.reverse()
print(data1)
```

Output

```
['abc', 123, 10.5, 'a', 'xyz']
['xyz', 'a', 10.5, 123, 'abc']
```

8) Python list sort() method

```
list1 = [56, 123, -56, 2111, 3]
list1.sort()
print(list1)

list2 = ["python", "world", "elonmusk", "language"]
list2.sort()
print(list2)
```

Output

```
[-56, 3, 56, 123, 2111]
['elonmusk', 'language', 'python', 'world']
```

Iteration through a list

Iterating lists mean going through each value present in that list.

There are multiple ways to iterate over a list in Python. Let us see all the different ways to iterate over a list in Python:

Using FOR-LOOP

```
list1 = [56, 123, -56, 2111, 3]

for i in list1:
    print(i)
```

Output

```
56
123
-56
2111
3
```

Using FOR-LOOP and RANGE()

In case we want to use the traditional for loop which iterates from number x to number y.

```
list1 = [56, 123, -56, 2111, 3]

length = len(list1)
for i in range(length):
    print(list1[i])
```

Output

```
56
123
-56
2111
3
```

Using IN LIST

```
list1 = [56, 123, -56, 2111, 3]

for i in list1:
    print(i)
```

Output

```
56
123
-56
2111
3
```

Using WHILE-LOOP

```
list1 = [56, 123, -56, 2111, 3]

length = len(list1)
i = 0

while i < length:
    print(list1[i])
    i = i + 1
```

Output

```
56
123
-56
2111
3
```

Practice Examples (LIST)

1. Write a Python program to sum all the items in a list.
2. Write a Python program to multiply all the items in a list.
3. Take 10 integer inputs from user and store them in a list and print them on screen.
4. Take 20 integer inputs from user and print the following:
    a. number of positive numbers
    b. number of negative numbers
    c. number of odd numbers
    d. number of even numbers
    e. number of 0s.
5. Take 10 integer inputs from user and store them in a list. Again, ask user to give a number. Now, tell user whether that number is present in list or not.
6. Make a list by taking 10 inputs from user. Now delete all repeated elements of the list.
   E.g.-
   INPUT: [1,2,3,2,1,3,12,12,32]
   OUTPUT: [1,2,3,12,32]
7. Ask user to give integer inputs to make a list. Store only even values given and print the list.

For solutions, check the end of the book.