Python Programming

Presented by

Mariena A. A.

Asst. Professor and Head, Department of Computer Science

Little Flower College, Guruvayoor.

List

- List in python is implemented to store the sequence of various type of data. However, python contains six data types that are capable to store the sequences but the most common and reliable type is list.
- A list can be defined as a collection of values or items of different types. The items in the list are separated with the comma (,) and enclosed with the [ ].

- Ex:-

L1 = ["John", 102, "USA"]
L2 = [1, 2, 3, 4, 5, 6]
L3 = [1, "Ryan’]

Example

```
emp = ["John", 102, "USA"]
print("printing employee data...");
print("Name : %s, ID: %d, Country: %s"%(emp[0],emp[1],emp[2]))
```

list

The indexing are processed in the same way as it happens with the strings. The elements of the list can be accessed by using the slice operator [].

The index starts from 0 and goes to length - 1. The first element of the list is stored at the 0th index, the second element of the list is stored at the 1st index, and so on.

List = [0, 1, 2, 3, 4, 5]

0	1	2	3	4	5

List[0] = 0

List[1] = 1

List[2] = 2

List[3] = 3

List[4] = 4

List[5] = 5

List[0:] = [0,1,2,3,4,5]

List[:] = [0,1,2,3,4,5]

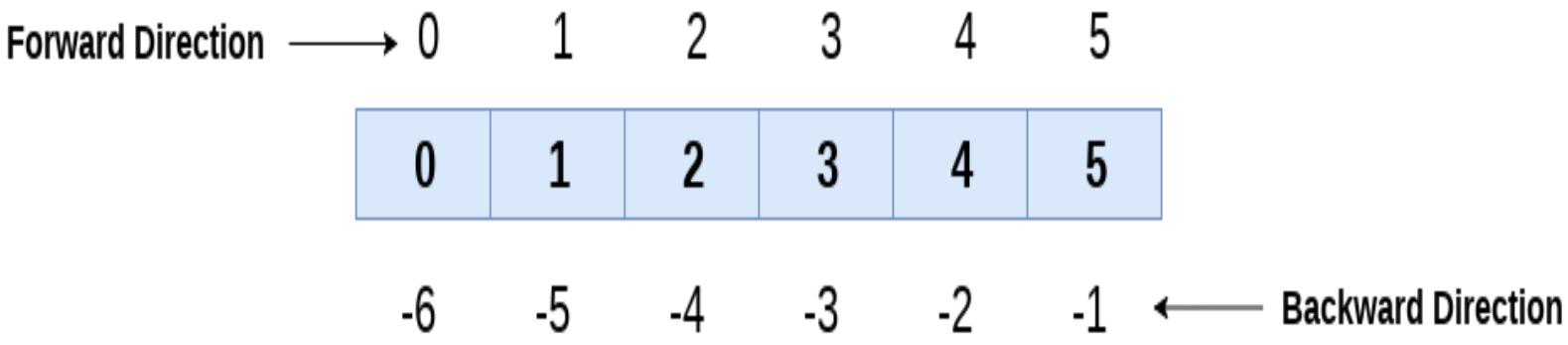
List[2:4] = [2, 3]

List[1:3] = [1, 2]

List[:4] = [0, 1, 2, 3]

list

List = [0, 1, 2, 3, 4, 5]



list

Lists are the most versatile data structures in python since they are immutable and their values can be updated by using the slice and assignment operator.
Python also provide us the append() method which can be used to add values to the string.

Ex:-

```
List = [1, 2, 3, 4, 5, 6]
print(List)
List[2] = 10;
print(List)
List[1:3] = [89, 78]
print(List)
```

The list elements can also be deleted by using the **del** keyword. Python also provides us the remove() method if we do not know which element is to be deleted from the list.

List operations

1.Consider a List l1 = [1, 2, 3, 4], **and** l2 = [5, 6, 7, 8]

Operator	Description	Example
Repetition	The repetition operator enables the list elements to be repeated multiple times.	L1*2 = [1, 2, 3, 4, 1, 2, 3, 4]
Concatenation	It concatenates the list mentioned on either side of the operator.	l1+l2 = [1, 2, 3, 4, 5, 6, 7, 8]
Membership	It returns true if a particular item exists in a particular list otherwise false.	print(2 in l1) prints True.
Iteration	The for loop is used to iterate over the list elements.	for i in l1: print(i)**Output**1 2 3 4
Length	It is used to get the length of the list	len(l1) = 4

List operations

A list can be iterated by using a for - in loop. A simple list containing four strings can be iterated as follows.

```
List = ["John", "David", "James", "Jonathan"]
for i in List: #i will iterate over the elements of the List and contain
s each element in each iteration.
    print(i);
```

Python provides append() function by using which we can add an element to the list. However, the append() method can only add the value to the end of the list.

```
l =[];
n = int(input("Enter the number of elements in the list")); #Number
of elements will be entered by the user
for i in range(0,n): # for loop to take the input
    l.append(input("Enter the item?")); # The input is taken from the
user and added to the list as the item
print("printing the list items....");
for i in l: # traversal loop to print the list items
    print(i, end = "  ");
```

list

```
List = [0,1,2,3,4]
print("printing original list: ");
for i in List:
    print( i, end=" ")
List. remove(0)
print("\n printing the list after the removal of first element...")
for i in List:
    print(i, end=" ")
```

Built in function

No:	Function	Description
1	cmp(list1, list2)	It compares the elements of both the lists.
2	len(list)	It is used to calculate the length of the list.
3	max(list)	It returns the maximum element of the list.
4	min(list)	It returns the minimum element of the list.
5	list(seq)	It converts any sequence to the list.

Built in functions

1	list.append(obj)	The element represented by the object obj is added to the list.
2	list.clear()	It removes all the elements from the list.
3	List.copy()	It returns a shallow copy of the list.
4	list.count(obj)	It returns the number of occurrences of the specified object in the list.
5	list.extend(seq)	The sequence represented by the object seq is extended to the list.
6	list.index(obj)	It returns the lowest index in the list that object appears.

7	list.insert(index, obj)	The object is inserted into the list at the specified index.
8	list.pop(obj=list[-1])	It removes and returns the last object of the list.
9	list.remove(obj)	It removes the specified object from the list.
10	list.reverse()	It reverses the list.
11	list.sort([func])	It sorts the list by using the specified compare function if given.

Built in functions in list

Built in functions that return value but doesn't change the list

seq=[2,3,5,7,4,8,4]

seq. count(4) # it returns 2

Index with one argument and 2 argument

seq. index(5) # it returns index of first occurrence of the given value 2

seq.index (value, start) # from which index value we have to start search

seq.index(4,5)

seq.index(4,3)

Doesn't return value but change the list

append(value)

seq.extend([7,8,9]) # more than one elements are added to the list

seq.insert(index, value) # we can add at given index value

seq. remove(4) # first occurrence value will be removed

seq. reverse() # it reverse the list

seq. sort() # sort the list in ascending order

Seq. pop() # remove the last element


Thank you!