Informatics Practices

Lists in python

Rajesh Verma MRA DAV Public School,Solan

Introduction

- The python lists are containers that are used to store a list of values of any type.
- It is a standard data type of Python that can store a sequence of values belonging to any type.
- Python lists are mutable.
- List differs from string and tuples as lists are mutable but strings and tuples are immutable.
- The lists are depicted through square brackets [ ].

Creating Lists

- [ ]
- [1,2,3]
- [‘a’, ‘b’, ‘c’]
- [‘one’, ‘two’, ‘three’]

- empty list
- list of integers
- list of characters
- list of strings

To create a list, put a no. of expressions in square brackets. Use square brackets [ ] to indicate the ***start*** and ***end*** of the list, and separate the items by commas(,).

Empty List

The empty list is [ ]. It is the list equivalent of 0 or ‘ ’ and like them it also has truth value as ***false.***

You can create an empty list by-

L = [ ] or

L = list()

It will generate an empty list and name that list as L.

Nested Lists

A list can have an element in it, which itself is a list. Such a list is called a nested list, e.g.,

L = [3,4,[5,6],7]

L is a nested list with four elements : 3, 4, [5,6]and 7. L[2] element is a list [5,6]. Length of L is 4 as it counts [5,6] as one element.

Similarity with Strings

- Length- Function len(L) returns the no. of items in the list L.
- Indexing - L[i] returns the item at index i(the first item has index 0), and
- Slicing - L[a:b] returns a new list, containing the objects at indexes between a and b(excluding b)
- Concatenation and replication operators + and *-

  The + operator adds one list to the end of another. The * operator repeats a list.

Accessing Individual Elements of Lists

The individual elements of a list are accessed through their indexes. List elements are indexed, i.e., forward indexing as 0,1,2,3,…. And backward indexing as -1,-2,-3,….

vowels = [‘a’, ‘e’, ‘i’, ‘o’, ‘u’]

vowels[0] **#a**

vowels[4] **#u**

vowels[-5] **#a**

If you give index outside the legal indices(0 to length-1 or –length,-length+1,…,-1), Python will raise ***Index Error***

vowels[5] #Index Error: list index out of range

Lists are Mutable

vowels = [‘a’, ‘e’, ‘i’, ‘o’, ‘u’]

vowels[0] = ‘A’

vowels = [‘A’, ‘e’, ‘i’, ‘o’, ‘u’]

vowels[-4] = ‘E’

vowels = [‘A’, ‘E’, ‘i’, ‘o’, ‘u’]

It changes the element in place in the same list as lists are ***mutable***

Comparing Lists

- For comparing operators >, <, >=, <=, the corresponding elements of two lists must be of comparable types, otherwise python would give an error.
- Python gives the result of non equality comparisons as soon as it gets a result in terms of true/false from corresponding elements comparison. It jumps to next element when the first element of both lists is same and so on.

```
[ 1, 2, 4 ] > [ 1, 2 ]          #True
[ 1, 2, 8, 9 ] < [ 1, 2, 9, 1 ]  #True
[ 1, 2, 3 ] < [ 1, [1,2] ,3 ]    # Type error
```

Joining, Replicating and Slicing of lists

- The concatenation operator +, when used with lists, join two lists. Two lists can be joined.

  L1,L2= [1, 2],[3, 4, 5] L1 + L2 = [1, 2, 3, 4, 5]

  L1 + 2 #Type Error

- The * operator replicate a list specified no. of times.

  L1*3 = [1, 2, 1, 2, 1, 2,]

- List slices are the sub part of a list extracted out. The list slice is itself a list.

  L1 =[10, 12, 13, 20, 25, 29, 20, 48]

  seq= L1[3:7] seq=[20, 25, 29, 20]

List Functions and Methods

1. Index method – The function returns the index of first matched item from the list.

   L=[13, 18, 11,16,18, 14]

   L.index(18) #1

   Returns 1st index of value 18 even if there is another 18 at index 4

2. Append method – The method adds an item at the end of the list.

   L = [‘red’, ‘blue’]

   L.append(‘yellow’)

   L = [‘red’, ‘blue’, ‘yellow’]

3. Extend method – It takes a list as an argument and appends all of the elements of the argument list to the list on which extend() is applied.

L1,L2 = [1, 2, 3], [4, 6]

L1.extend(L2)

L1= [1, 2, 3, 4, 6]

4. Insert method- This function inserts an item at a given position.

L1.insert(2, '4')

L1= [1, 2, 4, 3]

5. Pop method – It removes an element from the given position in the list, and return it. If no index is specified, pop() removes and returns the last element.

L = [1, 2, 4, 1, 6]

ele= L.pop(0) #1

L = [2, 4, 1, 6]

6. Remove method – It removes the first occurrence of given item from the list.

L.remove(1)

L= [2, 4, 1, 6]

7. Clear method – It removes all the elements in the list and makes it empty and return nothing.

L= [1, 3, 2, 5, 7]

L.clear()

L = [ ]

8. Count method – It returns the count of the item that you passed as argument.

```
L = [ 13, 18, 29, 34, 18 ]
L.count(18)                    #2
L.count(32)                    #0
```

9. Reverse method – It reverses the items of the list.

```
L1 = L.reverse()
L1 = [ 18,34, 29, 18, 13 ]
```

10. Sort method - It sorts the items of the list in increasing order.

```
L = [ 'a', 'f', 'd', 'e' ]
L.sort()
L= [ 'a', 'd', 'e', 'f' ]
```

For decreasing order using sort, write

```
List.sort(reverse=True)
```

Some Programs on Lists

- Program to find minimum element from a list of elements along with its index in the list

```
a=eval(input("Enter list:"))
length=len(a)
min_ele=a[0]
min_index=0
For i in range(1,length-1):
    if a[i]<min_ele:
        min_ele=a[i]
        min_index=I
print("Given list is :",a)
print("The minimum element of the given list is :")
print(min_ele,"at index",min_index)
```

Output

```
Enter list: [2,3,4,-2,6,-7,8,11,-9,11]
Given list is : [2,3,4,-2,6,-7,8,11,-9.11]
The minimum element of the given list is :
-9 at index 8
```

- Program to calculate mean of a given list of numbers

```
a=eval(input("Enter list:"))
length=len(a)
mean=sum=0
for i in range(0,length-1):
    sum+=a[i]
mean=sum/length
print("Given list is :",a)
print("The mean of the given list is :",mean)
```

Output

```
Enter list: [7,23,-11,55,13,5,20.05,-5.5]
Given list is : [7,23,-11,55,13,5,20.05,-5.5]
The mean of the given list is : 15.364285714285714
```

Any Questions Please