Python Lists And List Functions | Python Tutorial...

Show Course Contents ⊕

Overview Q&A Downloads Announcements

Python Lists And List Functions

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. E.g.

```
list1 = ['harry', 'ram', 'Aakash', 'shyam', 5, 4.85]
```

The above list contains strings, an integer, and even an element of type float. A list can contain any kind of data, i.e., it’s not mandatory to form a list of only one data type. The list can contain any kind of data in it.

Do you remember we saw indexing in strings? List elements can also be accessed by using Indices, i.e., the first element of the list has 0 index and the second element has from 0 and goes up to (index-1) or 3.

Have a look at the examples below:

```
                        Lists in Python

[]                                         # list with no member,
[1, 2, 3]                                # list of integers
[1, 2.5, 3.7, 9]                       # list of numbers (integers
['a', 'b', 'c']                          # lisst of characters
['a', 1, 'b', 3.5, 'zero']           # list of mixed value types
['One', 'Two', 'Three']            # list of strings
```

List Methods :

Here is the list of list methods in Python. These methods can be used in any python list to produce the desired output.

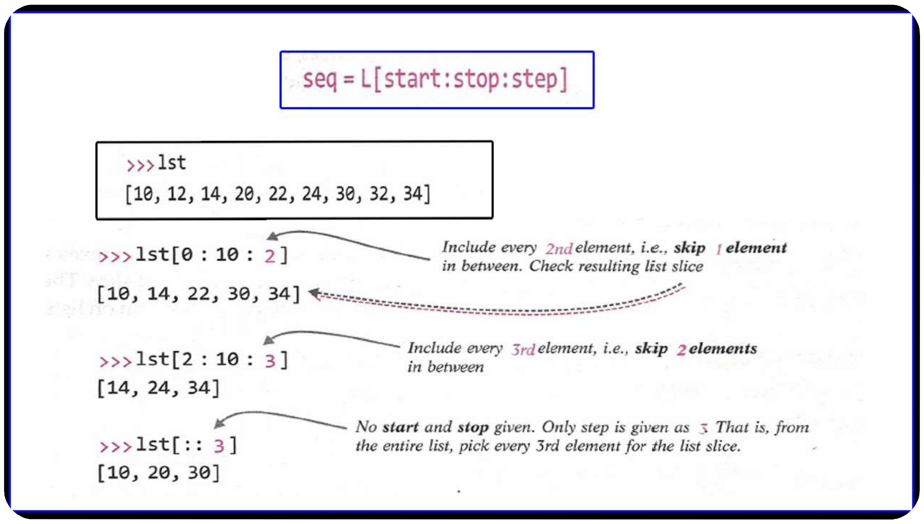
```
 # List Methods :
l1=[1,8,4,3,15,20,25,89,65]       #l1 is a list
print(l1)

l1.sort()
print(l1)      #l1 after sorting
l1.reverse()
print(l1)      #l1 after reversing all elements
```

List Slicing :

List slices, like string slices, return a part of a list extracted out of it. Let me explain; you can use indices to get elements and create list slices as per the following format :

```
seq = list1[start_index:stop_index]
```



```
seq = L[start:stop:step]

>>>lst
[10, 12, 14, 20, 22, 24, 30, 32, 34]

>>>lst[0 : 10 : 2]
[10, 14, 22, 30, 34]

>>>lst[2 : 10 : 3]
[14, 24, 34]

>>>lst[:: 3]
[10, 20, 30]
```

Include every 2nd element, i.e., skip 1 element in between. Check resulting list slice
Include every 3rd element, i.e., skip 2 elements in between
No start and stop given. Only step is given as 3 That is, from the entire list, pick every 3rd element for the list slice.

List Methods:

There are a lot of list methods that make our life easy while using lists in python. Let's have a look at a few of them below:

```
 # List Methods :-
list1=[1,2,3,6,,5,4]     #list1 is a list

list1.append(7)    # This will add 7 in the last of list
list1.insert(3,8)    # This will add 8 at 3 index in list
list1.remove(1)    #This will remove 1 from the list
list1.pop(2)          #This will delete and return index 2 value.
```

Tuples in Python:

A tuple is an immutable data type in Python. A tuple in python is a collection of elements enclosed in () (parentheses). Tuple, once defined, can't be changed, i.e., its elements or values can't be altered or manipulated.

```
 # Tuples in Python :
a=()    # It's an example of empty tuple
x=(1,)   # Tuple with single value i.e. 1
```

the python interpreter will interpret it as a single entity which is why it's important to use a ',' after the element while creating tuples of a single element.

Swapping of two numbers

Python provides a very handy way of swapping two numbers like below:

```
# Swapping of two numbers :
a = 10
b = 15
print(a,b)     #It will give output as: 10 15
a,b = b,a
print(a,b)     #It will give output as: 15 10
```

Code file as described in the video

```
grocery = ["Harpic", "vim bar", "deodrant", "Bhindi",
           "Lollypop", 56]
# print(grocery[5])
numbers = [2, 7, 9, 11, 3]
# numbers.remove(9)
# numbers.pop()
# numbers.sort()
# numbers = []
# numbers.reverse()
# numbers.append(1)
# numbers.append(72)
# numbers.append(5)
# numbers.insert(2, 67)
# print(numbers)
# 3, 11, 9, 7, 2
# tp = (1,)
# print(tp)
a= 1
b = 8
a, b = b,a
# temp = a
# a = b
# b = temp
print(a, b)
```

Previous Next

CodeWithHarry