Python Programming

EDUCATION FOR PEACE

(ESTD-1995)

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

LIST DATA TYPE

Guido Van Rossum

Learning Mantra

If you really strong in the basics, then remaining things will become so easy.

Agenda:

1. Important functions of List

i. Manipulating Elements of List

II. Manipulating Elements of List:

3. extend() function:

- To add all items of one list to another list, we use extend() method.

Eg:

l1.extend(l2) ➔ All items present in l2 will be added to l1

Eg:

```
order1=["Chicken","Mutton","Fish"]
order2=["RC","KF","FO"]
order1.extend(order2)
print(order1)       ➔ ['Chicken', 'Mutton', 'Fish', 'RC', 'KF', 'FO']
print(order2)       ➔ ['RC', 'KF', 'FO']
```

II. Manipulating Elements of List:

Eg:

order1=["Chicken","Mutton","Fish"]

order2=["RC","KF","FO"]

order3 = order1 + order2

print(order1) ➔ ['Chicken', 'Mutton', 'Fish']

print(order2) ➔ ['RC', 'KF', 'FO']

print(order3) ➔ ['Chicken', 'Mutton', 'Fish', 'RC', 'KF', 'FO']

II. Manipulating Elements of List:

Eg:

```
l1 = [10,20,30]

l2 = [40,50,60]

l1.extend(l2)

print(l1)                ➔ [10, 20, 30, 40, 50, 60]
```

II. Manipulating Elements of List:

Eg:

```
order=["Chicken","Mutton","Fish"]
order.extend("Mushroom")  # It adds every character as a single element to the list
print(order) ➔['Chicken', 'Mutton', 'Fish', 'M', 'u', 's', 'h', 'r', 'o', 'o', 'm']
```

Explanation:

Here, 'Mushroom' is a string type, in this string 8 elements are there. These elements are added separately.

Eg:

```
order=["Chicken","Mutton","Fish"]
order.append("Mushroom") # It adds this string as a single element to the list
print(order)      ➔ ['Chicken', 'Mutton', 'Fish', 'Mushroom']
```

II. Manipulating Elements of List:

4. remove() function:

- We can use this function to remove specified item from the list.
- If the item present multiple times then only first occurrence will be removed.

Eg:

```
n=[10,20,10,30]
n.remove(10)
print(n)          ➔ [20, 10, 30]
```

Note: If the specified item not present in list then we will get **ValueError.**

II. Manipulating Elements of List:

Eg:

```
n=[10,20,10,30]
n.remove(40)
print(n)            ➔ValueError: list.remove(x): x not in list
```

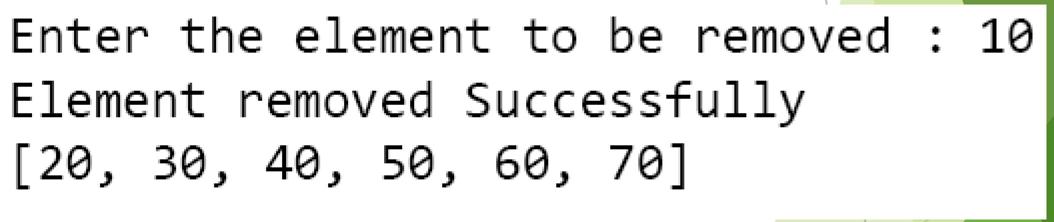
Note:

Hence before using remove() method first we have to check specified element present in the list or not by using in operator.

II. Manipulating Elements of List:

Eg:

```
l1= [10,20,30,40,50,60,70]

x = int(input('Enter the element to be removed : '))

if x in l1:

    l1.remove(x)

    print('Element removed Successfully ')

    print(l1)

else:

    print('Specified element is not available ')
```

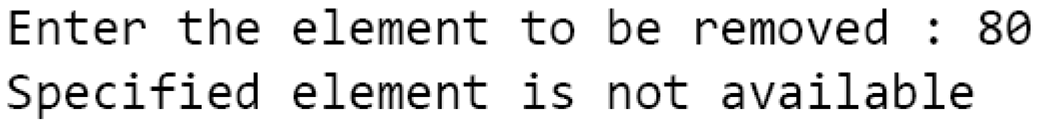


```
Enter the element to be removed : 10
Element removed Successfully
[20, 30, 40, 50, 60, 70]
```

II. Manipulating Elements of List:

Eg:

```
l1= [10,20,30,40,50,60,70]
x = int(input('Enter the element to be removed : '))
if x in l1:
    l1.remove(x)
    print('Element removed Successfully ')
    print(l1)
else:
    print('Specified element is not available ')
```



```
Enter the element to be removed : 80
Specified element is not available
```

II. Manipulating Elements of List:

5. pop() function:

- It removes and returns the last element of the list.
- This is only function which manipulates list and returns some element.

Eg:

```
n=[10,20,30,40]
print(n.pop())          ➔40
print(n.pop())          ➔30
print(n)                ➔[10,20]
```

II. Manipulating Elements of List:

- If the list is empty then pop() function raises IndexError.

```
n=[]
print(n.pop())        ➔ IndexError: pop from empty list
```

Note:

1. pop() is the only function which manipulates the list and returns some value.

2. In general we can use append() and pop() functions to implement stack data structure by using list, which follows LIFO(Last In First Out) order.

3. In general we can use pop() function to remove last element of the list. But we can use remove() function to remove elements based on index.

II. Manipulating Elements of List:

We can use pop() function in following ways:

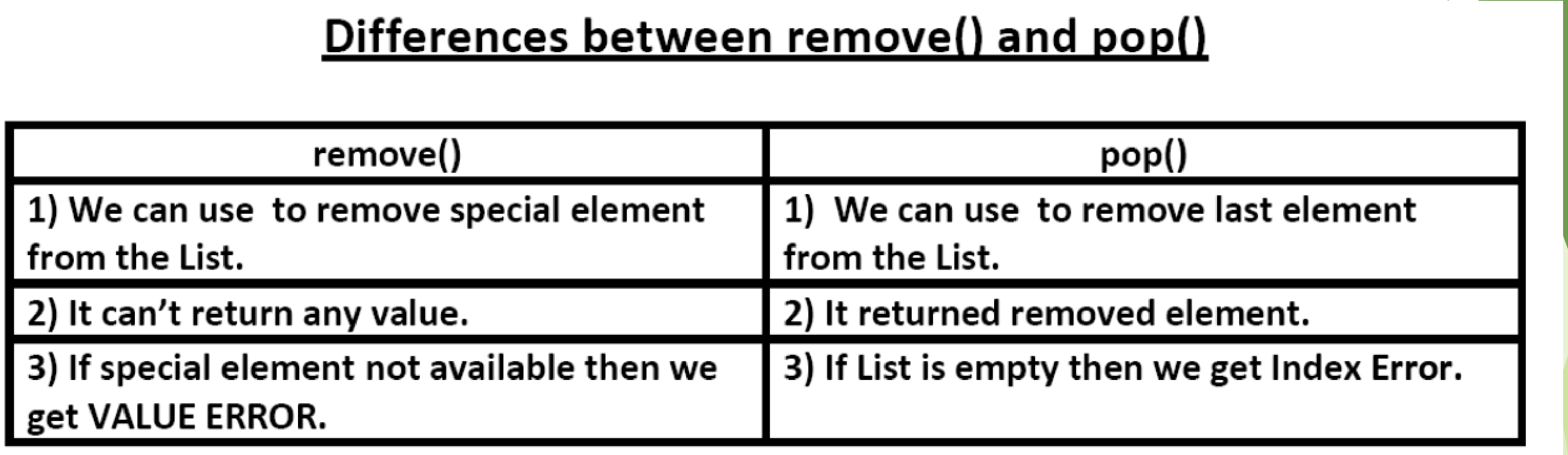
1. n.pop(index) ➔ To remove and return element present at specified index.

2. n.pop() ➔ To remove and return last element of the list.

Eg:

```
n=[10,20,30,40,50,60]
print(n.pop())      ➔ 60
print(n.pop(1))     ➔ 20
print(n.pop(10))    ➔ IndexError: pop index out of range
```

II. Manipulating Elements of List:



Differences between remove() and pop()

remove()	pop()
1) We can use to remove special element from the List.	1) We can use to remove last element from the List.
2) It can't return any value.	2) It returned removed element.
3) If special element not available then we get VALUE ERROR.	3) If List is empty then we get Index Error.

Note:

List objects are dynamic. i.e., based on our requirement we can increase and decrease the size.

- append(),insert() ,extend() ➔ For increasing the size/growable nature
- remove(), pop() ➔ For decreasing the size /shrinking nature

Any question?

If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You