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. **Traversing the elements of List**
2. **Important functions of List**

4. Traversing the elements of List:

- The sequential access of each element in the list is called traversal.

1. By using while loop:

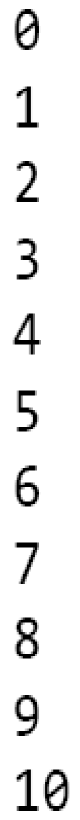
```
n=[0,1,2,3,4,5,6,7,8,9,10]

i=0

while i<len(n):

    print(n[i])

    i=i+1
```



```
0
1
2
3
4
5
6
7
8
9
10
```

2. By using for loop:

```
n=[0,1,2,3,4,5,6,7,8,9,10]
for n1 in n:
    print(n1)
```

Output:

```
0
1
2
3
4
5
6
7
8
9
10
```

3. Python Program to display only even numbers in the given list.

```
n=[0,1,2,3,4,5,6,7,8,9,10]
for n1 in n:
    if n1%2==0:
    print(n1)
```

Output:

```
0
2
4
6
8
10
```

4. Python program to display elements by index wise.

```
l=["A","B","C"]
x=len(l)
for i in range(x):
    print(l[i],"is available at positive index: ",i,"and at negative index: ",i-x)
```

Output:

```
A is available at positive index: 0 and at negative index: -3
B is available at positive index: 1 and at negative index: -2
C is available at positive index: 2 and at negative index: -1
```

5. Important functions of List

What is the difference between function and method?

- In Python you can use both these terms interchangeably.

-Function:

- Function by default considered as method also.
- If a function is declaring outside a class is called as function.

- Method :

- If you are declaring a function inside a class is called as a method.
- In other words, if you are calling any function with object reference is called as method.

Note: Python is both functional oriented as well as object oriented programming language.

I. To get information about list:

1. len():

- It returns the number of elements present in the list.

```
n=[10,20,30,40]
print(len(n))           ➔ 4
```

2. count():

- It returns the number of occurrences of specified item in the list.

```
n=[1,2,2,2,2,3,3]
print(n.count(1))   ➔1
print(n.count(2))   ➔3
print(n.count(3))   ➔2
print(n.count(4))   ➔0
```

I. To get information about list:

3. index() function:

- It returns the index of first occurrence of the specified item.

Eg:

```
n=[1,2,2,2,2,3,3]
print(n.index(1))        → 0
print(n.index(2))        →1
print(n.index(3))        →5
print(n.index(4))        →ValueError: 4 is not in list
```

I. To get information about list:

Note:

- If the specified element not present in the list then we will get ValueError.
- Hence before index() method we have to check whether item present in the list or not by using in operator.

Eg:

```
n=[1,2,2,2,2,3,3]
print(4 in n))        ➔False
```

I. To get information about list:

Eg:

```
l = [10,20,30,40,10,20,10,10]
target = int(input('Enter value to search : '))
if target in l:
    print(target,'available and its first occurrence is at ',l.index(target))
else:
    print(target,' is not available')
```

Output:

```
Enter value to search : 50
50 is not available
```

I. To get information about list:

Eg:

```
l = [10,20,30,40,10,20,10,10]
target = int(input('Enter value to search : '))
if target in l:
    print(target,'available and its first occurrence is at ',l.index(target))
else:
    print(target,' is not available')
```

Output:

```
Enter value to search : 20
20 available and its first occurrence is at 1
```

I. To get information about list:

Eg:

```
l = [10,20,30,40,10,20,10,10]
target = int(input('Enter value to search : '))
if target in l:
    print(target,'available and its first occurrence is at ',l.index(target))
else:
    print(target,' is not available')
```

Output:

```
Enter value to search : 10
10 available and its first occurrence is at 0
```

II. Manipulating Elements of List:

1. append() function:

- We can use append() function to add item at the end of the list.
- By using this append function, we always add an element at last position.

Eg:

```
list=[]
list.append("A")
list.append("B")
list.append("C")
print(list)              ➔['A', 'B', 'C']
```

II. Manipulating Elements of List:

Eg: To add all elements to list up to 100 which are divisible by 10.

```
list=[]
for i in range(101):
    if i%10==0:
        list.append(i)
print(list)        ➔ [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

Another Way:

```
list= []
for i in range(0,101,10):
    list.append(i)
print(list)        ➔ [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

II. Manipulating Elements of List:

2. insert() function:

- It is used to insert item at specified index position.

Eg:

```
n=[1,2,3,4,5]

n.insert(1,888)

print(n)          ➔[1, 888, 2, 3, 4, 5]
```

II. Manipulating Elements of List:

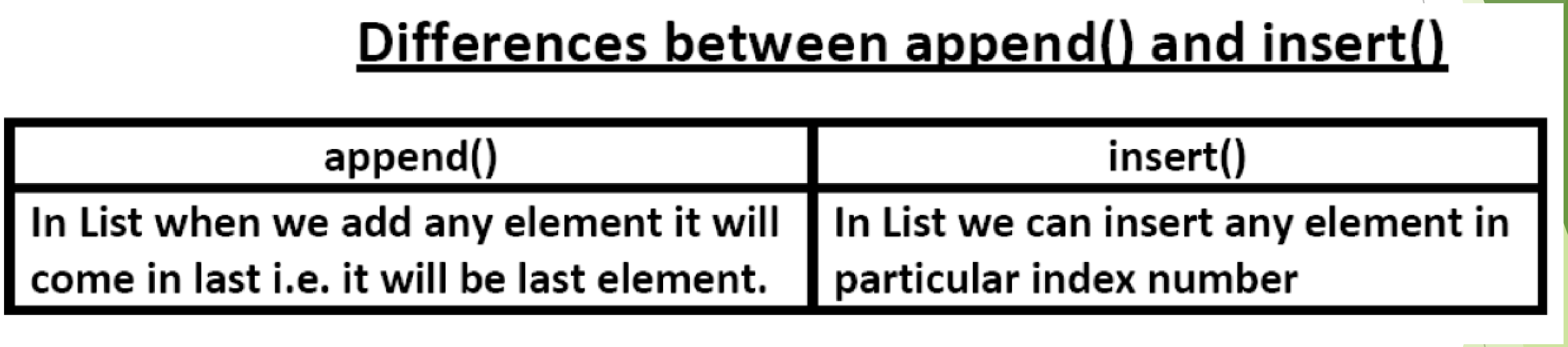
Eg:

```
n=[1,2,3,4,5]
n.insert(10,777)
n.insert(-10,999)
print(n)              ➔[999, 1, 2, 3, 4, 5, 777]
print(n.index(777)) ➔ 6
print(n.index(999)) ➔ 0
```

Note:

- If the specified index is greater than max index then element will be inserted at last position.
- If the specified index is smaller than min index then element will be inserted at first position.

II. Manipulating Elements of List:



Differences between append() and insert()

append()	insert()
In List when we add any element it will come in last i.e. it will be last element.	In List we can insert any element in particular index number

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