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

UNIT – IV:

Lists: Creation of list objects, Accessing and traversing the elements of list. Important functions of list – len(), count(), index(), append(), insert(), extend(), remove(), pop(), reverse() and sort(). Basic Operations on list: Aliasing and Cloning of List objects, Mathematical Operators for list objects, Comparing list objects, Membership operators on list, Nested Lists, List Comprehensions. Illustrative examples on all the above topics.

Tuples: Creation of Tuple objects, Accessing elements of tuple, Mathematical operators for tuple, Important functions of Tuple – len(),count(),index(), sorted(), min(), max(), cmp().Tuple Packing and Unpacking. Illustrative examples on all the above topics.

Guido Van Rossum

Learning Mantra

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

Topics Covered:

1. Introduction
2. Creation of List Objects
3. Accessing elements of List
4. Traversing the elements of List
5. Important functions of List :
    i) To get information about list
    ii) Manipulating elements of List
    iii) Ordering elements of List
6. Aliasing and Cloning of List objects
7. Using Mathematical operators for List Objects
8. Comparing List objects
9. Membership operators
10. clear() function
11. Nested Lists
12. List Comprehensions

Agenda:

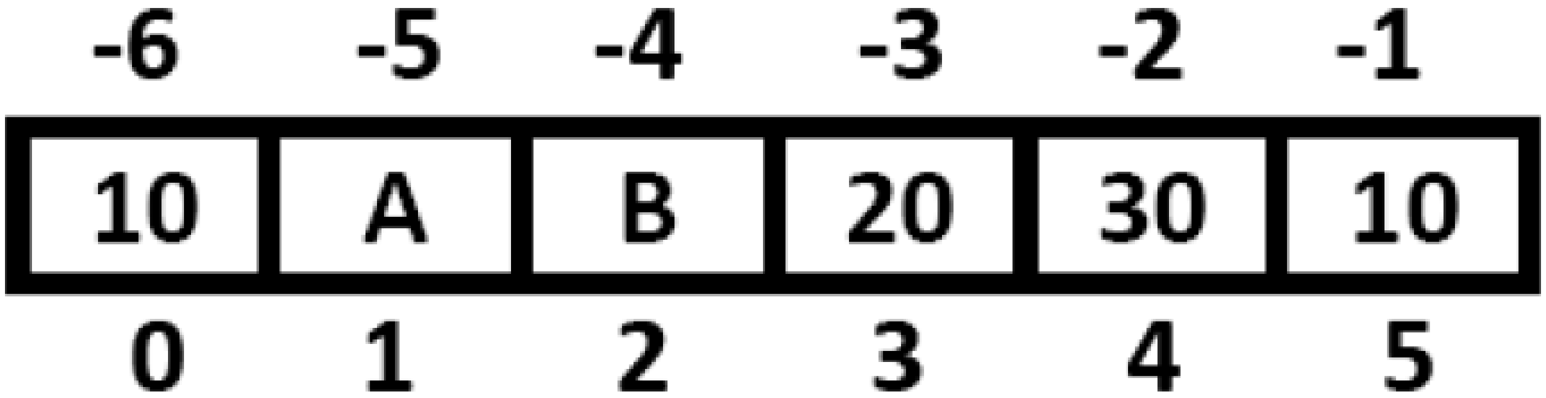
1. Introduction
2. Creation of List Objects
3. Accessing elements of List

INTRODUCTION

- If we want to represent a group of individual objects as a single entity where insertion order is preserved and duplicates are allowed, then we should go for List.
- Insertion order is preserved.
- Duplicate objects are allowed.
- Heterogeneous objects are allowed.
- List is dynamic because based on our requirement we can increase the size and decrease the size.
- In List the elements will be placed within square brackets and with comma separator.
- We can differentiate duplicate elements by using index and we can preserve insertion order by using index. Hence index will play very important role.

- Python supports both positive and negative indexes. +ve index means from left to right where as negative index means right to left.

  Eg: [10,"A","B",20, 30, 10]



-6	-5	-4	-3	-2	-1
10	A	B	20	30	10
0	1	2	3	4	5

- List objects are mutable. i.e., we can change the content.

2. Creation of List Objects:

1. We can create empty list object as follows...

```
list=[]
print(list)
print(type(list))
```

Output:

```
[]
<class 'list'>
```

2. If we know elements already then we can create list as followslist=[]

```
list = [10,20,30,40]

print(list)

print(type(list))
```

Output:

```
[10, 20, 30, 40]

<class 'list'>
```

3. With dynamic input:

```
list=(input("Enter List:"))        # Entire input is considered as string
print(list)
print(type(list))
```

Output:

```
Enter List:10,20,30,40
10,20,30,40
<class 'str'>
```

Eg:

```
list=eval(input("Enter List:"))
print(list)
print(type(list))
```

Output:

```
Enter List:[10,20,30,40]
[10, 20, 30, 40]
<class 'list'>
```

Eg:

```
list=eval(input("Enter List:"))
print(list)
print(type(list))
```

Output:

```
Enter List:[ram,raj]
NameError: name 'ram' is not defined
```

Eg:

```
list=eval(input("Enter List:"))
print(list)
print(type(list))
```

Output:

```
Enter List:['ram','raj']
['ram', 'raj']
<class 'list'>
```

4. With list() function:

```
l=list(range(0,10,2))
print(l)
```

Output:

```
[0, 2, 4, 6, 8]
```

```
s=“karthi"
l=list(s)
print(l)
```

Output:

```
[‘k', ‘a', 'r', ‘t', ‘h‘,’i']
```

5. With split() function:

```
s="Learning Python is very very easy !!!"
l=s.split()
print(l)
print(type(l))
```

Output:

```
['Learning', 'Python', 'is', 'very', 'very', 'easy', '!!!']
<class 'list'>
```

Note:

- Sometimes we can take list inside another list, such type of lists are called nested lists. For Example, [10,20,[30,40]].

3. Accessing elements of List:

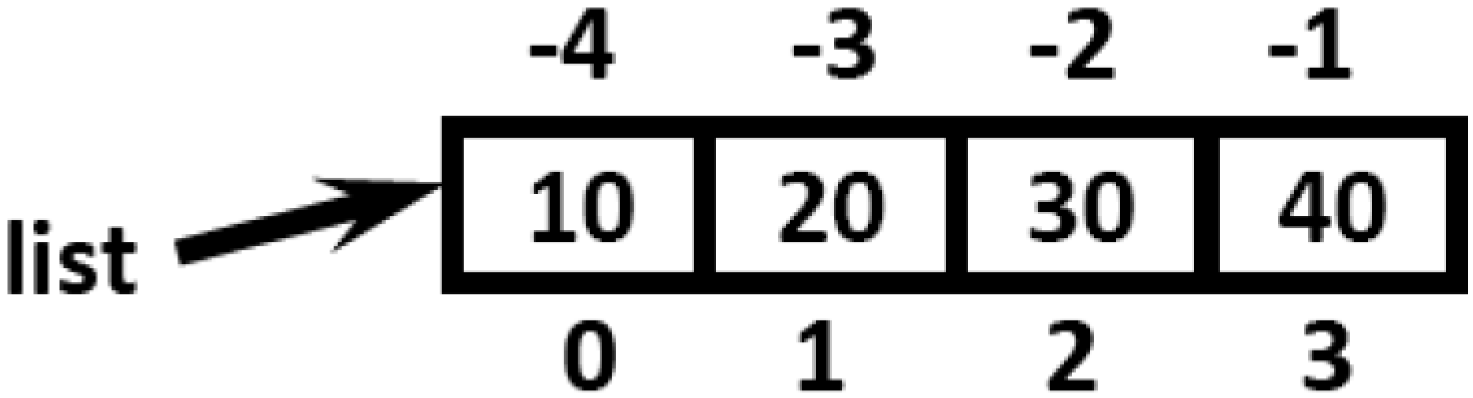
- We can access elements of the list either by using index or by using slice operator.

1. By using index:

- List follows zero based index. i.e., index of first element is zero.
- List supports both Positive and Negative indexes.
- Positive index meant for Left to Right.
- Negative index meant for Right to Left.

Eg:

list=[10,20,30,40]



Eg:

```
list=[10,20,30,40]
print(list[0])        ➔10
print(list[-1])       ➔40
print(list[-4])       ➔10
print(list[10])       ➔IndexError: list index out of range
```

Eg:

```
list = [10,20,[30,40]]
print(list[2])        ➔[30,40]
print(list[2][1])     ➔40
```

2. By using slice operator:

Syntax:

list2= list1[start:stop:step]

start ➔ It indicates the index where slice has to start default value is 0

stop ➔ It indicates the index where slice has to end default value is max allowed index of list i.e., length of the list.

step ➔ Increment value

Note: step default value is 1

Eg:

```
l = [10,20,30,40,50,60]
print(l[::])        ➔[10, 20, 30, 40, 50, 60]
```

Eg:

```
l = [10,20,30,40,50,60]
print(l[::2])       ➔[10, 30, 50]
```

```
l = [10,20,30,40,50,60]
print(l[::-1])      ➔ [60, 50, 40, 30, 20, 10]
```

```
l = [10,20,[30,40],50,60]
print(l[0:3:])      ➔[10, 20, [30, 40]]
```

Eg:

```
n=[1,2,3,4,5,6,7,8,9,10]
print(n[2:7:2])          ➔ [3,5,7]
print(n[4::2])           ➔ [5,7,9]
print(n[3:7])            ➔ [4,5,6,7]
print(n[8:2:-2])         ➔ [9,7,5]
print(n[4:100])          ➔ [5,6,7,8,9,10]
```

List vs mutability:

- Once we creates a List object, we can modify its content. Hence List objects are mutable.

Eg:

```
n=[10,20,30,40]
print(n)          ➔ [10, 20, 30, 40]
n[1]=777
print(n)          ➔[10, 777, 30, 40]
```

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