Python Programming

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. Ordering elements of List

2. Aliasing and Cloning of List objects

III. Ordering elements of List:

1. reverse():

- We can use to reverse() order of elements of list.

Eg:

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

III. Ordering elements of List:

2. sort() function:

- In list by default insertion order is preserved. If you want to sort the elements of list according to default natural sorting order then we should go for sort() method.

  -For numbers ➔ default natural sorting order is Ascending Order.

  -For Strings ➔ default natural sorting order is Alphabetical Order.

Eg:

```
n=[20,5,15,10,0]
n.sort()
print(n)          ➔ [0, 5, 10, 15, 20]
```

III. Ordering elements of List:

Eg:

```
s=["Dog","Banana","Cat","Apple"]
s.sort()
print(s)        ➔ ['Apple', 'Banana', 'Cat', 'Dog']
```

Eg:

```
s=["Dog","Banana","Cat","apple"]
s.sort()        # Unicode values are used during comparison of alphabets
print(s)        ➔ ['Banana', 'Cat', 'Dog', 'apple']
```

III. Ordering elements of List:

Note:

- To use sort() function, compulsory list should contain only homogeneous elements, otherwise we will get **TypeError.**

Eg:

```
n=[20,10,"A","B"]
n.sort()
print(n)
```

➔**TypeError:** '<' not supported between instances of 'str' and 'int'

III. Ordering elements of List:

Note: In Python 2 if List contains both numbers and Strings then sort() function first sort numbers followed by strings.

Eg:

```
n=[20,"B",10,"A"]
n.sort()
print(n)      # [10,20,'A','B'] It is valid in Python 2, but in Python 3 it is Invalid
```

In Python3:

TypeError: '<' not supported between instances of 'str' and 'int'

III. Ordering elements of List:

How to sort the elements of list in reverse of default natural sorting order?

One Simple Way:

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

III. Ordering elements of List:

How to sort the elements of list in reverse of default natural sorting order?

Another Way:

- We can sort according to reverse of default natural sorting order by using **reverse = True** argument.

```
n=[40,10,30,20]
n.sort()
print(n)                    ➔ [10,20,30,40]
n.sort(reverse=True)
print(n)                    ➔ [40,30,20,10]
n.sort(reverse=False)
print(n)                    ➔ [10,20,30,40]
```

III. Ordering elements of List:

Eg:

```
s=["Dog","Banana","Cat","Apple"]
s.sort(reverse= True)                    # Reverse of Alphabetical order
print(s)                    ➔ ['Dog', 'Cat', 'Banana', 'Apple']
```

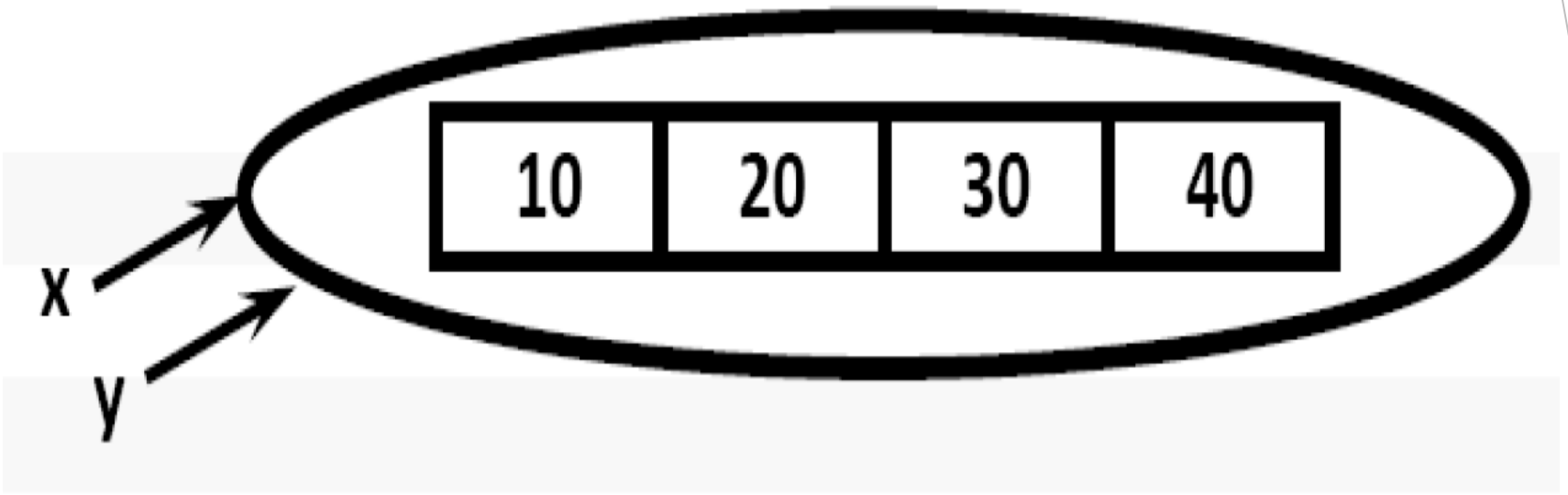
6 . Aliasing and Cloning of List objects:

- The process of giving another reference variable to the existing list is called aliasing.

Eg:

```
x=[10,20,30,40]
y=x
print(id(x))        ➔1709842944648
print(id(y))        ➔1709842944648
```

6 . Aliasing and Cloning of List objects:



- The problem in this approach is by using one reference variable if we are changing content, then those changes will be reflected to the other reference variable.

6 . Aliasing and Cloning of List objects:

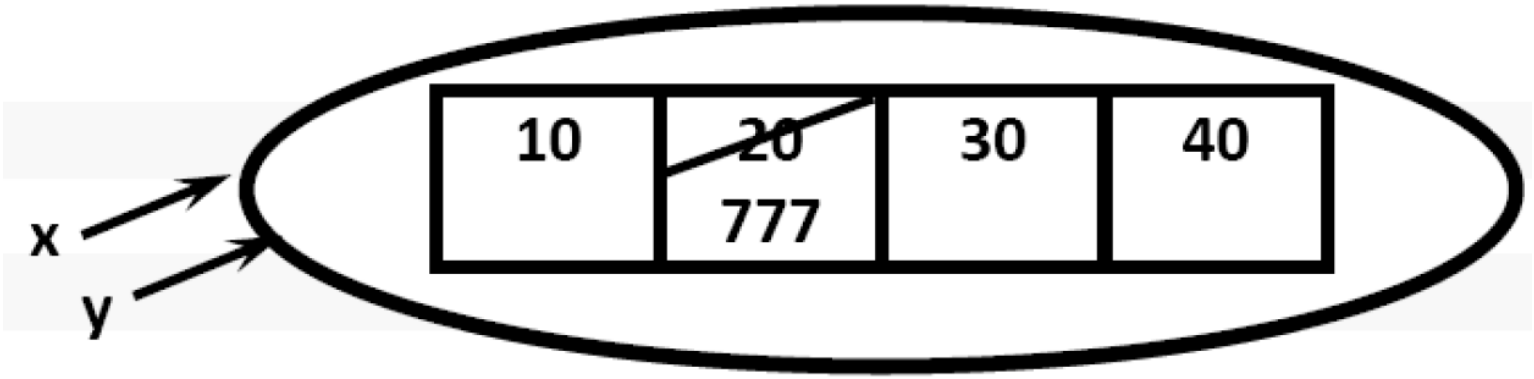
Eg:

x=[10,20,30,40]

y=x

y[1]=777

print(x) ➔[10,777,30,40]



To overcome this problem we should go for **cloning.**

6 . Aliasing and Cloning of List objects:

Cloning: The process of creating exactly duplicate independent object is called cloning.

We can implement cloning by using the following ways:

1. slice operator
2. copy() function

6 . Aliasing and Cloning of List objects:

1. By using slice operator:

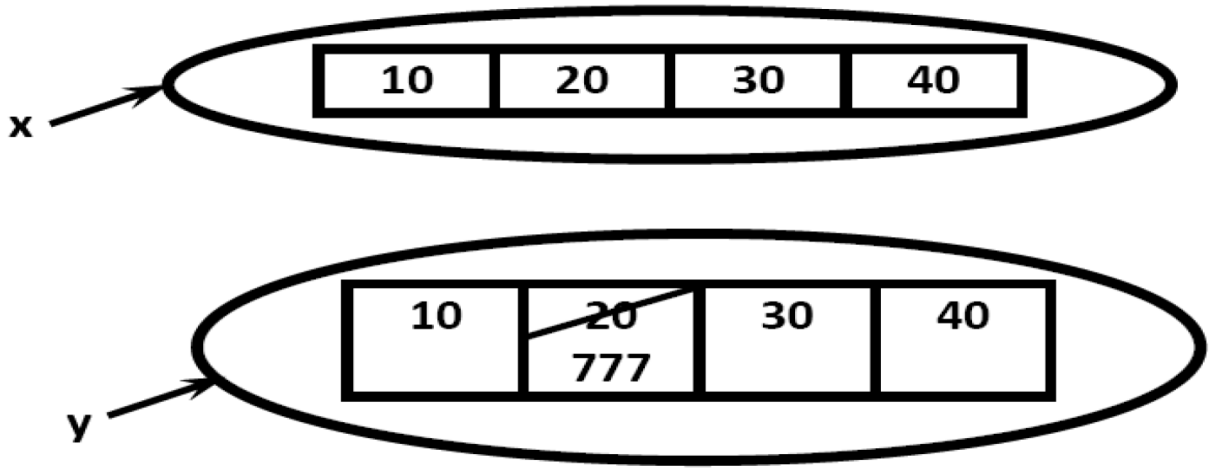
x=[10,20,30,40]

y=x[:]

y[1]=777

print(x) ➔ [10,20,30,40]

print(y) ➔ [10,777,30,40]

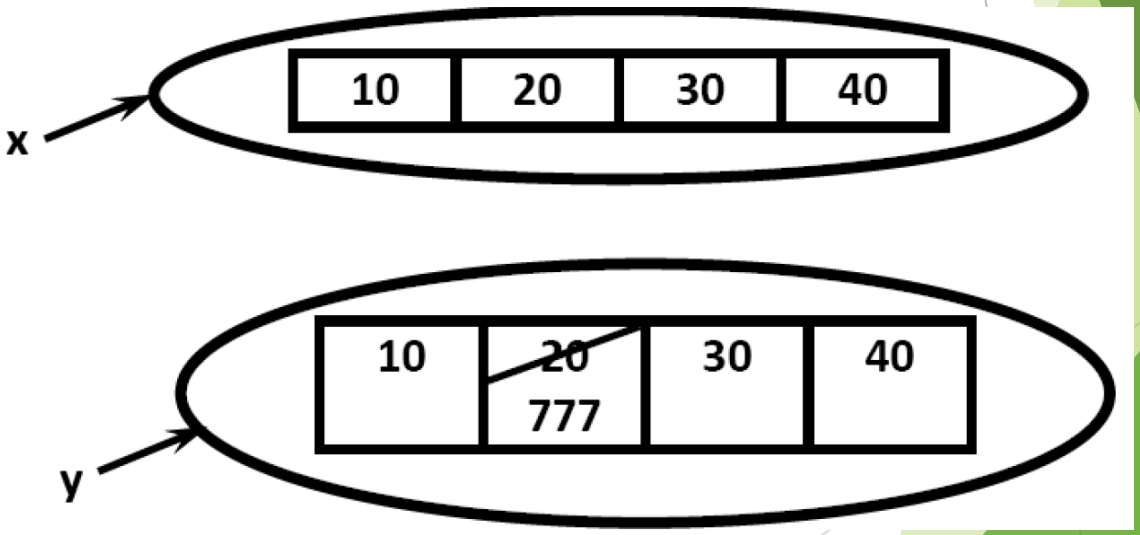


6 . Aliasing and Cloning of List objects:

2. By using copy() function:

Eg:

```
x=[10,20,30,40]
y=x.copy()
y[1]=777
print(x)    ➔ [10,20,30,40]
print(y)    ➔ [10,777,30,40]
```



6 . Aliasing and Cloning of List objects:

Q. What is the difference between = operator and copy() function?

- '=' operator meant for aliasing, 'copy() function' meant for cloning.

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