```
print(n)# [10,20,'A','B']
```

But in Python 3 it is invalid.

To sort in reverse of default natural sorting order:

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

Eg:

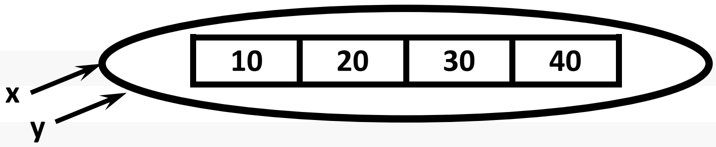
```
n=[40,10,30,20]
n.sort()
print(n) ==>[10,20,30,40]
n.sort(reverse=True)
print(n) ===>[40,30,20,10]
n.sort(reverse=False)
print(n) ==>[10,20,30,40]
```

Aliasing and Cloning of List objects:

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

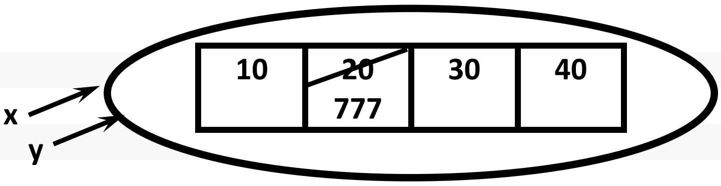
Eg:

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



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.

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

We can implement cloning by using slice operator or by using copy() function