Copy or Cloning

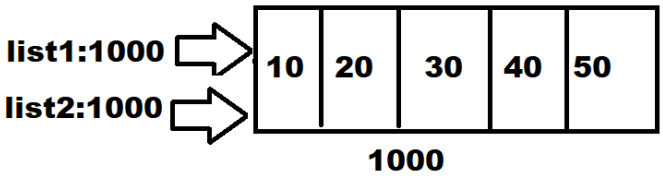
Python allows creating copy of collections.
It various methods or types of copies

1. Alias copy
2. Shallow copy
3. Deep copy

Alias copy

Assigning address of one list to another is called alias copy

list1=[10,20,30,40,50]



list2=list1

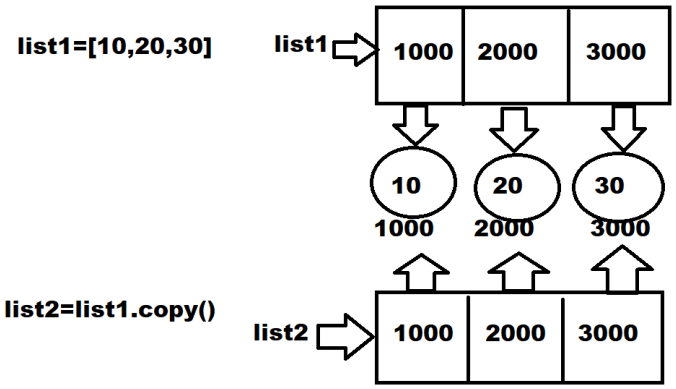
In Alias copy, same list object is assigned to multiple variables. In this approach new list object is not created.

```
>>> list1=[10,20,30,40,50]
>>> list2=list1
>>> id(list1)
1575085426944
>>> id(list2)
1575085426944
>>> print(list1)
[10, 20, 30, 40, 50]
>>> print(list2)
[10, 20, 30, 40, 50]
>>> list1[0]=99
>>> print(list1)
[99, 20, 30, 40, 50]
>>> print(list2)
```

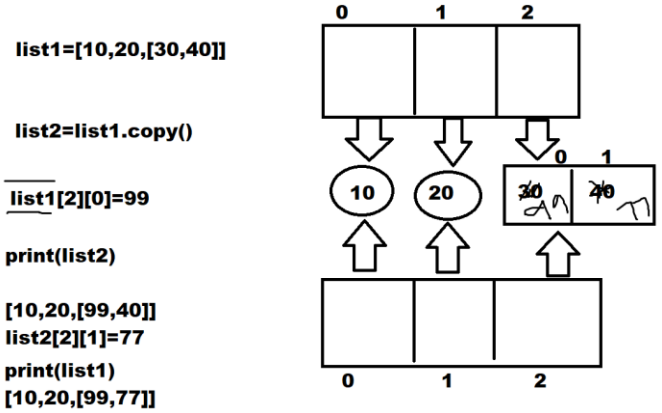
[99, 20, 30, 40, 50]
>>> del list1[0]
>>> print(list1)
[20, 30, 40, 50]
>>> print(list2)
[20, 30, 40, 50]

Shallow Copy OR Shallow Cloning

List provides a method called "copy".
This creates shallow copy of list.
In shallow copy a new list object is created by copying address of objects found in old list.



In shallow copy, if the objects exists inside list are mutable, the changes done in one list reflect to another list.

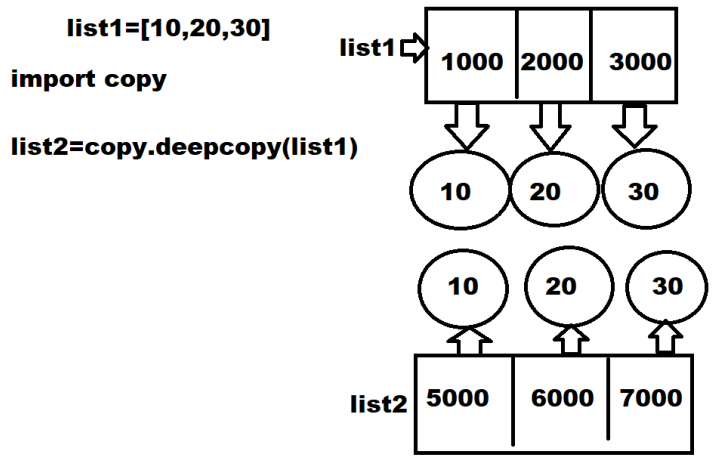


```
>>> list1=[10,20,[30,40]]
>>> list2=list1.copy()
>>> id(list1)
1575085243648
>>> id(list2)
1575085244224
>>> print(list1)
[10, 20, [30, 40]]
>>> print(list2)
[10, 20, [30, 40]]
>>>list1[2][0]=99
>>> print(lsit1)
Traceback (most recent call last):
  File "<pyshell#19>", line 1, in <module>
    print(lsit1)
NameError: name 'lsit1' is not defined. Did you mean: 'list1'?
>>> print(list1)
[10, 20, [99, 40]]
>>> print(list2)
[10, 20, [99, 40]]
>>> list2[2][1]=88
>>> print(list2)
[10, 20, [99, 88]]
>>> del list1[0]
>>> print(list1)
[20, [99, 88]]
>>> print(list2)
[10, 20, [99, 88]]
>>> del list2[2][0]
>>> print(list2)
[10, 20, [88]]
>>> print(list1)
[20, [88]]
```

Deep copy

In deep copy, a new list object created by copying objects found inside old list.

This deep copy is done using deepcopy function available in copy module.




```
>>> list1=[10,20,[30,40]]
>>> print(list1)
[10, 20, [30, 40]]
>>> import copy
>>> list2=copy.deepcopy(list1)
>>> print(list2)
[10, 20, [30, 40]]
>>> list1[2][0]=99
>>> print(list1)
[10, 20, [99, 40]]
>>> print(list2)
[10, 20, [30, 40]]
>>> list2[2][0]=88
>>> print(list2)
[10, 20, [88, 40]]
>>> print(list1)
[10, 20, [99, 40]]
```

What is difference between deep copy and shallow copy?

Shallow Copy	Deep Copy
In shallow copy a new list or collection is created by copying address of objects found in existing list or collection	In deep copy a new list or collection object is created by copying objects found existing list or collection
Changes done in one collection reflect to another collection	Changes done is one collection does not reflect to another collection
This copy is using "copy" method provided by list	This copy is done using deepcopy function provided by "copy" module/lib

function

```
def sorted():
```

```
class list:
  def sort():
  def remove()
  def insert()
  def append()
```

function/method

```
sorted()
```

```
list1.sort()
```

Sorting

Sorting is organization of elements/values in ascending or descending order. This sorting is done using two methods

1. Sort method of list
2. Sorted function

sort(*, key=None, reverse=False)

This method sorts the list in place, using only < comparisons between item. Sort method of list is mutable; sorting is done in same list.
Sort method of list by default organizes elements in ascending order.

```
>>> list1=[4,8,1,2,6,3,9,1,4,3]
>>> print(list1)
```

```
[4, 8, 1, 2, 6, 3, 9, 1, 4, 3]
>>> list1.sort()
>>> print(list1)
[1, 1, 2, 3, 3, 4, 4, 6, 8, 9]
>>> list1.sort(reverse=True)
>>> print(list1)
[9, 8, 6, 4, 4, 3, 3, 2, 1, 1]
```