Shallow copy vs deep copy

Shallow copy :

a shallow copy creates a new object which stores the reference of original object , but shallow copy doesn’t create a new copy of nested object , instead it just copies the reference of nested objects.

We use copy() function for copy module.

Ex:

import copy as cp

a = [10,20,30]

b = [40,50,60]

c = [a,b]# nested list

d = cp.copy(c)

print(id(c))# print id of c

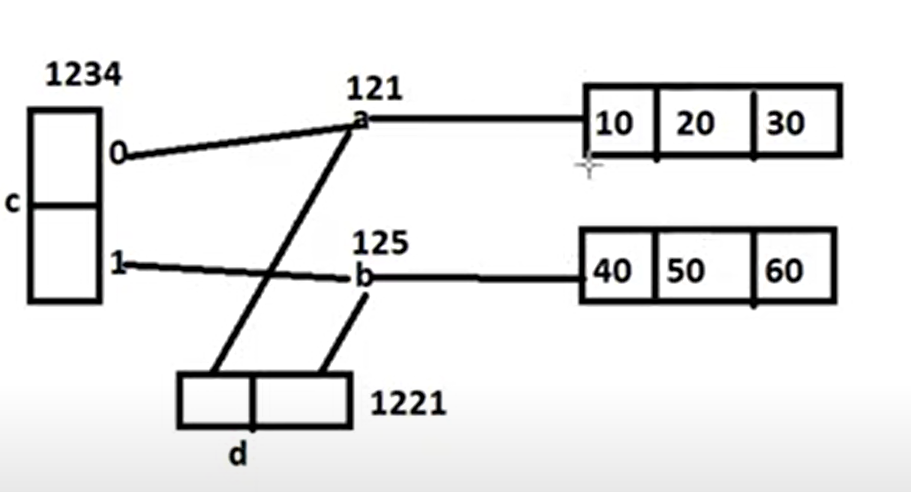
print(id(d))# print id of d , both are different objects.

print(id(c[0]))# print id for the first object inside c

print(id(d[0]))# print id for the first object inside d

# they both are equal because they pointing to same location i.e a

That means if we make any changes in c[0] the d[0] object is also change.

shallow copy diagram:

Deepcopy:

a deepcopy creates a new object which stores the reference of original object,as well as also creates new copy of nested object.

We use deepcopy() function to create deep copy.

Ex:

Import copy as cp

a = [10,20,30]

b = [40,50,60]

c = [a,b]

d = cp.deepcopy(c)

print(id(c))# printing id of c

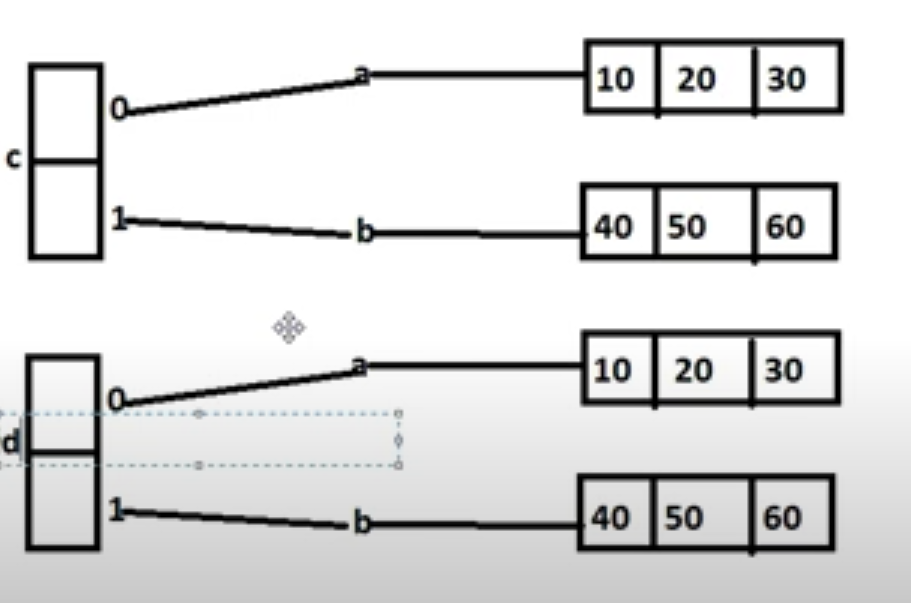
print(id(d)) # printing id of d, both are different objects

print(id(c[0])) # printing id of first nested element of c

print(id(d[0])) # printing id of first nested element of d

# both are different object with same value

Means if we change in c[0] , the d[0] object is not changed

Deepcopy diagram: