**Using the Method deepcopy from the Module copy**

A solution to the described problems is to use the module "copy". This module provides the method "copy", which allows a complete copy of a arbitrary list, i.e. shallow and other lists.   
  
The following script uses our example above and this method: 

from copy import deepcopy

lst1 = ['a','b',['ab','ba']]

lst2 = deepcopy(lst1)

lst2[2][1] = "d"

lst2[0] = "c";

**Copy with the Slice Operator**

It's possible to completely copy shallow list structures with the slice operator without having any of the side effects, which we have described above:

>>> list1 = ['a','b','c','d']

>>> list2 = list1[:]

>>> list2[1] = 'x'

>>> print list2

['a', 'x', 'c', 'd']

>>> print list1

['a', 'b', 'c', 'd']

>>>

But as soon as a list contains sublists, we have the same difficulty, i.e. just pointers to the sublists.

>>> lst1 = ['a','b',['ab','ba']]

>>> lst2 = lst1[:]

**Shallow copy:**

L=[1,2,3]

L2=L that is called shallow copy

**Deepcopy :**

L=[1,2,3]

L2=L[:]

If list contain sublist in that case its not work like ..

L=[1,2,[3,4]]

L1=L[:]

L[2][0]=10

Write one is :

**Import copy**

**L=[1,2,[3,4]]**

**L2=copy.deepcopy(L)**