Deep copy Vs Shallow Copy

```
#  = , copy(), deepcopy()

lst1 = [1,2,3,4]
lst2 = lst1 # lst2 is referring to same memory as lst1.
```

```
lst2[1] = 1000

lst2 # [1,1000,3,4]

lst1 # [1,1000,3,4]
```

copy() - Shallow copy

```
lst1 = [1,2,3,4]
lst2 = lst1.copy() # this copy method will create other memory location
                                    # for lst2, so any changes in lst2 doesn't affect lst1
```

```
lst2[1] = 1000

lst2 # [1,1000,3,4]

lst1 # [1,2,3,4]
```

Shallow copy nested list

```
lst1 = ([1,2,3,4], [5,6,7,8])
lst2 = lst1.copy()
```

```
lst1[1][0] = 100

lst1 # ([1,2,3,4], [100,6,7,8])

lst2 # ([1,2,3,4], [100,6,7,8])
```

```
lst1.append([2,3,4,5])

lst1 # ([1,2,3,4], [100,6,7,8], [2,3,4,5])
```

```
lst2 # ([1,2,3,4], [100,6,7,8])
```

Deep copy

```
import copy

lst1 = [1,2,3,4]
lst2 = copy.deepcopy(lst1)
```

```
lst2[1] = 1000

lst2 # [1,1000,3,4]

lst1 # [1,2,3,4]
```

Deep copy in nested list

```
lst1 = ([1,2,3,4], [5,6,7,8])
lst2 = copy.deepcopy(lst1)
```

```
lst1[1][0] = 100

lst1 # ([1,2,3,4], [100,6,7,8])

lst2 # ([1,2,3,4], [5,6,7,8])
```