Shallow Copy v/s Deep Copy in Python

Presented by:

Shreyansh Mohnot & Arush Sharma

Shallow Copy

- Shallow copy is only relevant for compound objects, for example: list of lists, etc.
- It copies only the reference of original object rather than copying the original object.
- Changes made to the original object is reflected in new object and vice-versa.

Example

- Multiple methods to perform Shallow Copy: list2 = list1, list2 = list(list1), list2 = list1[:]
- Changing specific items in list1 affects list2 items.
- Objects Memory id for both the list can be different but they are kept binded together in shallow copy process.

```
list1 = [[10,20,30], ["how", "Are", "You"], ["python", "is", "boring"]]

# list2 = list1
# list2 = list(list1)
list2 = list1[:]

list2[2][2] = "fun"

print("old list {}: {}\nnew list {}: {}".format(id(list1), list1, id(list2), list2))
# old list 1786455363464: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'fun']]
# new list 1786452225352: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'fun']]

list1.append([1,2,3])
print("old list {}: {}\nnew list {}: {}".format(id(list1), list1, id(list2), list2))
# old list 1786455363464: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'fun'], [1, 2, 3]]
# new list 1786452225352: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'fun']]
```

Deep Copy

- Deep copy is also relevant for compound objects
- It constructs the new object and recursively copies the content of object found in the original
- Changes made to the original object is NOT reflected in new object and vice-versa.

Example

- To perform deepcopy, we need to import copy module and use deepcopy()
- Changing one object does not modify another object.
- The memory id for both the objects are different in deep copy process.

```
# Deep Copy Example

list1 = [[10,20,30], ["how", "Are", "You"], ["python", "is", "boring"]]

list2 = copy.deepcopy(list1)

list2[2][2] = "fun"

print("old list {}: {}\nnew list {}: {}".format(id(list1), list1, id(list2), list2))
# old list 1786455301000: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'boring']]
# new list 1786455363464: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'fun']]

list1.append([1,2,3])
print("old list {}: {}\nnew list {}: {}".format(id(list1), list1, id(list2), list2))
# old list 1786455301000: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'boring'], [1, 2, 3]]
# new list 1786455363464: [[10, 20, 30], ['how', 'Are', 'You'], ['python', 'is', 'fun']]
```