

[Practice](#)[GATE CS](#)[Placements](#)[Videos](#)[Contribute](#)[Login/Register](#)**Quick Links for Python**[Recent Articles](#)[MCQ / Quizzes](#)[Practice Problems](#)**Basics**[Introduction](#)[New Generation Language](#)[Keywords, Set 1 Set 2](#)[Explore More...](#)**Variables**[Variables, Expressions & Functions](#)[Global and Local Variables](#)[Type Conversion](#)[Explore More...](#)**Operators**[Increment and Decrement Operator](#)[Teranry Operator & Divison Operator](#)[Logical and Bitwise Not Operators on Boolean](#)[Any & ALL](#)

Operator Functions Set 1 & Set 2
Data Types
Introduction
Arrays Set 1, Set 2
String Methods Set 1, Set 2, Set 3
String Template Class & String Formatting using %
List Methods Set 1, Set 2, Set 3
Tuples & Sets
Dictionary Methods Set 1, Set 2
ChainMap
Explore More...
Control Flow
Loops and Control Statements
Counters & Accessing Counters
Iterators & Iterator Functions Set 1, Set 2
Generators
Explore More...
Functions
Function Decorators
Returning Multiple Values
Yield instead of Return
Python Closures & Coroutine
Explore More...
Modules
Introduction



Numeric Functions & Logarithmic and Power functions
Calender Functions Set 1, Set 2
Complex Numbers Introduction & Important functions
Explore More...
Object Oriented Concepts
Class, Object and Members
Data Hiding and Object Printing
Inheritance, Subclass and super
Class method vs static method & Class or Static Variables
Explore More...
Exception Handling
Exception Handling
User-Defined Exceptions
Built-in Exceptions
Libraries and Functions
Timeit
Numpy Set 1, Set 2
Get and Post
import module & reload module
Collection Modules Deque, Namedtuple & Heap
Explore More...
Machine Learning with Python
Classifying data using Support Vector Machines(SVMs) in Python



K means Clustering
How to get synonyms/antonyms from NLTK WordNet in Python?
Explore More...
Misc
Sql using Python & MongoDB and Python
Json formatting & Python Virtual environment
Metaprogramming with Metaclasses in Python
Python Input Methods for Competitive Programming
Explore More...
Applications and Projects
Creating a proxy webserver Set 1, Set 2
Send Message to FB friend
Twitter Sentiment Analysis & Whatsapp using Python
Desktop Notifier & Junk File Organizer
Explore More...

copy in Python (Deep Copy and Shallow Copy)

Python defines a module which allows to deep copy or shallow copy mutable object using the inbuilt functions present in the module "**copy**".

Assignment statements in Python do not copy objects, they create bindings between a target and an object. For collections that are mutable or contain mutable items, a copy is sometimes needed so one can change one copy without changing the other.

Deep copy



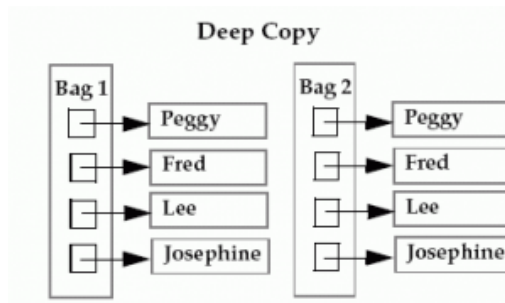


Image Source : <http://docs.roguewave.com/sourcepro/11.1/html/toolsug/6-4.html>

In case of deep copy, a copy of object is copied in other object. It means that **any changes** made to a copy of object **do not reflect** in the original object.

In python, this is implemented using “**deepcopy()**” function.

```
# Python code to demonstrate copy operations

# importing "copy" for copy operations
import copy

# initializing list 1
li1 = [1, 2, [3,5], 4]

# using deepcopy to deep copy
li2 = copy.deepcopy(li1)

# original elements of list
print ("The original elements before deep copying")
for i in range(0,len(li1)):
    print (li1[i],end=" ")

print("\r")

# adding and element to new list
li2[2][0] = 7

# Change is reflected in l2
print ("The new list of elements after deep copying ")
for i in range(0,len( li1)):
    print (li2[i],end=" ")

print("\r")

# Change is NOT reflected in original list
# as it is a deep copy
print ("The original elements after deep copying")
for i in range(0,len( li1)):
    print (li1[i],end=" ")
```

Run on IDE

Output:

```
The original elements before deep copying
1 2 [3, 5] 4
The new list of elements after deep copying
1 2 [7, 5] 4
The original elements after deep copying
1 2 [3, 5] 4
```



In the above example, the change made in the list **did not** effect in other list, indicating the list is deep copied.

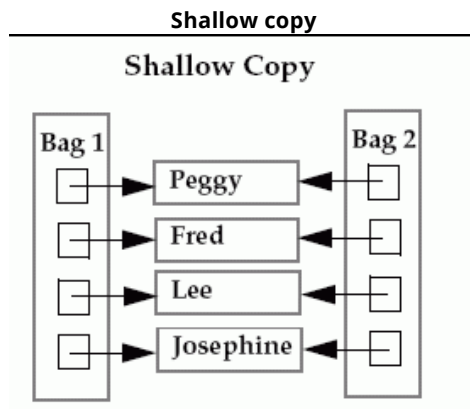


Image Source : <http://docs.roguewave.com/sourcepro/11.1/html/toolsug/6-4.html>

In case of shallow copy, a reference of object is copied in other object. It means that **any changes** made to a copy of object **do reflect** in the original object.

In python, this is implemented using "**copy()**" function.

```
# Python code to demonstrate copy operations
# importing "copy" for copy operations
import copy

# initializing list 1
li1 = [1, 2, [3,5], 4]

# using copy to shallow copy
li2 = copy.copy(li1)

# original elements of list
print ("The original elements before shallow copying")
for i in range(0,len(li1)):
    print (li1[i],end=" ")

print("\r")

# adding and element to new list
li2[2][0] = 7

# checking if change is reflected
print ("The original elements after shallow copying")
for i in range(0, len( li1)):
    print (li1[i],end=" ")
```

Run on IDE

Output:

```
The original elements before shallow copying
1 2 [3, 5] 4
The original elements after shallow copying
1 2 [7, 5] 4
```



In the above example, the change made in the list **did** effect in other list, indicating the list is shallow copied.

Important Points:

The difference between shallow and deep copying is only relevant for compound objects (objects that contain other objects, like lists or class instances):

- A shallow copy constructs a new compound object and then (to the extent possible) inserts references into it to the objects found in the original.
- A deep copy constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original.

Reference: [Python official Documentation](#)

This article is contributed by **Manjeet Singh**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

GATE CS Corner Company Wise Coding Practice

Python

Recommended Posts:

[Import module in Python](#)

[Python | Set 2 \(Variables, Expressions, Conditions and Functions\)](#)

[GET and POST requests using Python](#)

[Reloading modules in Python](#)

[Sets in Python](#)



([Login](#) to Rate and Mark)

3.3

Average Difficulty : **3.3/5.0**
Based on **3** vote(s)



Add to TODO List



Mark as DONE

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!

@geeksforgeeks, Some rights reserved

[Contact Us!](#)

[About Us!](#)

[Careers!](#)

[Privacy Policy](#)

