Notes on Chapter 5 - Structured Types, Mutability, and High Order Functions

Swarup Tripathy [*]

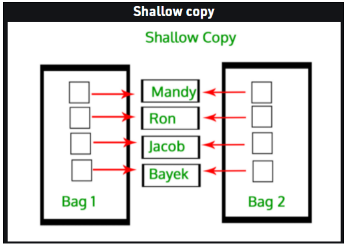
February 2022

A curated list of important points for my reference.

1. Literals of type **tuples** are written by enclosing a comma separated list of elements within parenthesis.
2. Like strings, tuples can be concatenated, indexed and sliced.
3. Sequences and Multiple Assignments
   - Executing the statement x,y=(3,4) where x will be bound to 3 and y to 4
   - The statement a,b,c = 'xyz' will bind x to a, y to b and z to c.
4. Built-in-function **id**, which returns a unique integer identifier for an object.
5. Deep and Shallow copy in python

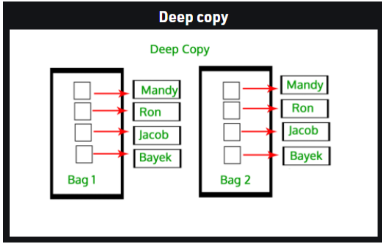
```
import copy
copy.copy(x)        #shallow copy
copy.deepcopy(x)    #deep copy
```

   - Shallow Copy → A shallow copy creates a new object which stores the reference of the original elements. So, a shallow copy doesn't create a copy of nested objects, instead it just copies the reference of nested objects. This means, a copy process does not recurse or create copies of nested objects itself.



[*] John V Guttag

- Deep Copy → A deep copy creates a new object and recursively adds the copies of nested objects present in the original elements. The deep copy creates independent copy of original object and all its nested objects.



6. Python supports creation of anonymous functions (i.e., functions which are not bound to a name), using the reserved word **lambda**. The general form of a lambda expression is

```
    lambda <sequence of variable names>: <expression>

L=[]
for i in map(lambda x,y: x**y,[1,2,3,4],[3,2,1,0]):
    L.append(i)
print(L)

prints out [1, 4, 3, 1]
```

7. IMMUTABLE datatypes (e.g. int, float, bool, str, tuple, unicode) cant be changed after they are created,

8. MUTABLE datatypes (e.g. list, set, dict, byte array) can be changed. An object whose internal state can be changed is mutable

9. In python programming, the second name given to a piece of data is known as an alias. Aliasing happens when the value of one variable is assigned to another variable because variables are just names that store references to actual value. reference: Aliasing