CSI33 Data Structures

Sharon Persinger

Fall 2019

Day 6

Python memory model – how names and values are stored

- Most programming languages: A variable is a name for a memory location that holds a value.
- Assign x = 37.2

- Python doesn't have variables, but instead has names. A name holds a reference to an object.
- Technically, every assignment of a name to an object gets stored in a Python namespace dictionary as a key:value pair. The name is the key and a reference to the object is the value.

Examples

- In the Python shell:
- Assign
  - x = “some text”
  - y = x
- Look at the namespace dictionary locals()
- Look at id(x) and id(y).
- Assign
  - y = “different”
- What happens to id(x) and id(y)?

- One string object has two names.
- Aliasing

- id(x) is the reference(address) of the object with name x.
- Assignment changes what object a name refers to. It does not change the object.

Removing objects

- The Python interpreter keeps a count of the number of references to an object. This is called the object's reference count. When an object's reference count becomes 0, the memory for the object is automatically deallocated, so it can be used for storing other objects. Python has automatic garbage collection.

- It is also possible to remove the mapping for name, with
  - del y
- Look again at the namespace dicctionary.

Advantages of Python's memory model

- A name contains a reference to an object and all references are the same size (4 bytes or 8 bytes).
- The type of data belongs to the object, not its name, so names can be reassigned.
- Container objects - lists, tuples, dictionaries – can be heterogeneous since they just store references to the varied objects.

Aliasing and mutable objects

- Simple data in Python is immutable. Once created, it can't be changed in the same memory location. Examples? int, float, tuple, string

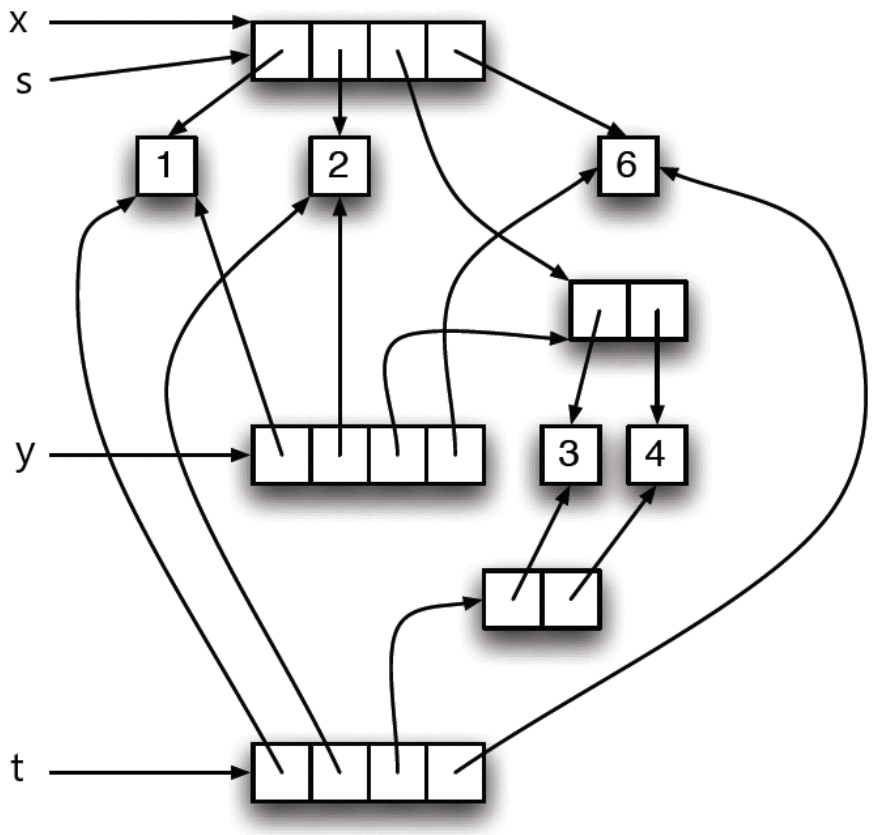
- A list can be changed in the same memory location.
  - Look at some examples.
- A list can have multiple names – aliasing.

- Any changes made using one name will show up as changes with the other name.

Copying a list to avoid the effects of aliasing

- Python has a module copy for copying objects.
- It has a shallow copy method copy() and deep copy method deepcopy()
- Create a list
  - x = [1, 2, [3, 4], 5] and make both a shallow copy y and a deep copy t of the list.
- Make some changes and look at the effect. Look especially at what happens when you change the object at x[2].
- Compare x, y, and t using is and ==
- What's happening?

Shallow copy and deep copy



Pictorial representation of shallow and deep copies

Linked Lists

- In an array, the data elements are placed in a sequence of contiguous memory locations.

- In a Python list, the references to data objects are placed into an array, so organized into neighboring memory locations.

- A linked list is a data structure in which data elements are organized into a sequence by links. Each element of a linked list is called a node. A node consists of the data and a link (reference) to the next node in the list.

- A linked list is a linear data structure, but the elements do not have to be stored in consecutive memory locations. The links provide a means of moving from one node to the following node.

List Node class

- Before we created a linked list ADT, we will make a simple class for a List Node object _-

```
class ListNode(object):


    def __init__(self, item = None, link = None):
        '''creates a ListNode with the specified data value and link
        post: creates a ListNode with the specified data value and link'''

        self.item = item
        self.link = link
```

Connecting together ListNode objects

- Create a ListNode object n3 that has data 3 and a link to None.
- Create another ListNode object n2 and insert it in front of n1. Do this again.
- What does the sequence of ListNode objects look like?

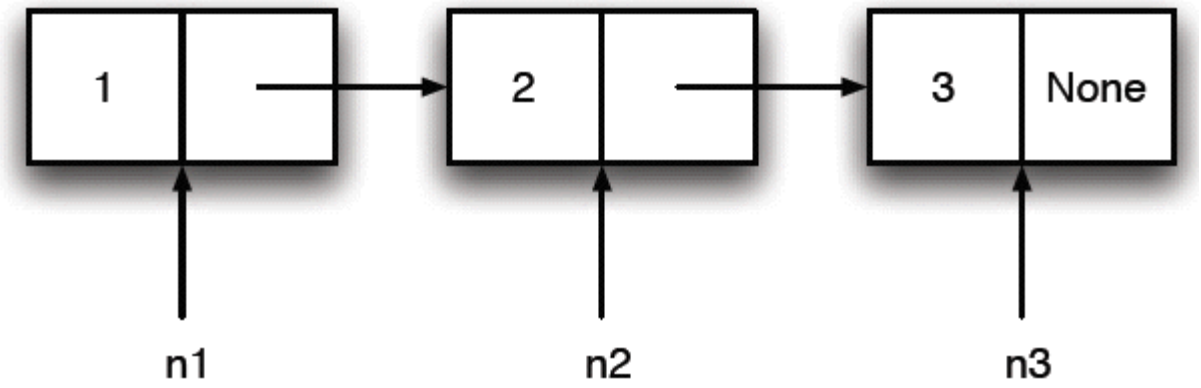
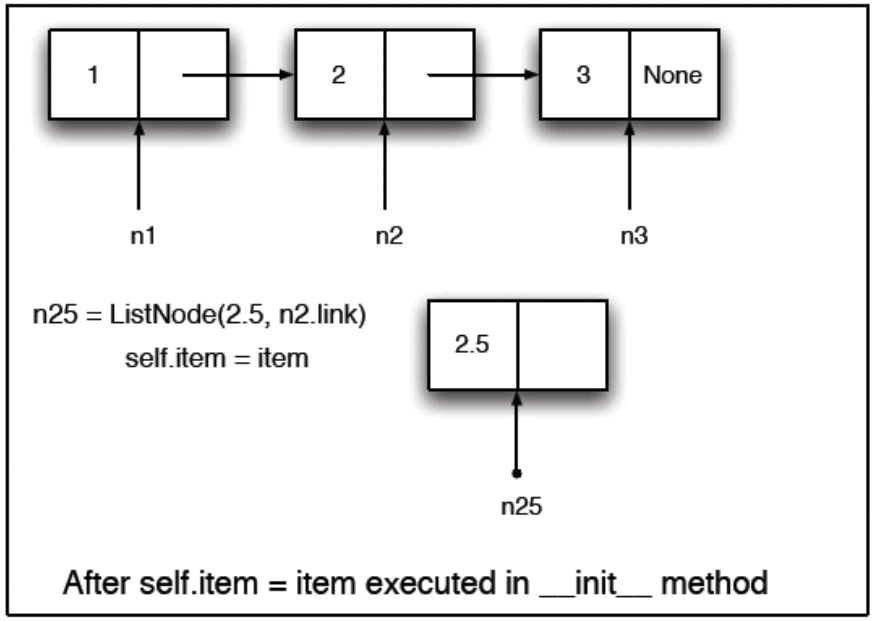


Figure 4.6: Three `ListNodes` linked together

Inserting in the middle of a list

- Create a ListNode object n25 that has data 2.5.
- Insert n25 into the list so that it comes after node n2 and before node n3.
  - You have to be careful with the order of changes to the links. Why?



Delete the node n2 from the linked list.