

## A Linked Class



<https://seedotnet.wordpress.com/tag/singly-linked-list/>

### Objectives.

---

- To practice writing linked lists

### Overview

---

Linked lists consist of nodes that have 2 attributes: the data being stored and a link to the next node. Nodes are also called self-referential data types because they are objects contain a link (reference) to another object of the same class. You will use the ListNode class developed in the lecture class and attached to the assignment to create a LinkedList class.

Your LinkedList class should have the following methods:

- `__init__()`: The constructor should not take any arguments. It should initialize the one attribute of your linked list, head, to None.
- `insert_front(item)`: This method should create a new node that contains the item passed to the method and insert the node at the beginning of the linked list.
- `insert_position(item, position)`: This method should create a new node that contains the item passed to the method and insert the node at the position passed. Positions should work like indices in python lists: the first node will be position 0.
- `insert_end(item)`: This method should create a new node that contains the item passed to the method and insert the node at the end of the linked list.
- `delete_front()`: This method should remove the first node in the linked list. The python garbage collector will then remove the node from memory.
- `delete_end()`: This method should remove the last node in the linked list. The python garbage collector will then remove the node from memory.
- `delete_item(item)`: This method should remove the **first occurrence** of item from the linked list. If there are duplicates farther down the linked list, they will remain.
- `delete_list()`: This method should remove all nodes in the linked list.
- `__str__()`: This method should return a string that consists of all data items stored in the linked list and separated by a comma followed by a space

## Instructions

---

The assignment has been divided into 2 parts to help you know where to start. You must finish both parts before turning it in. A driver program that uses your class has been provided. You may not change the driver file except to comment out lines that test methods that you haven't written. If there is a problem, you must change your class so that it works with the driver.

### Part 1

Write the class with the constructor, `insert_front`, `delete_front`, and `__str__` methods.

### Part 2

Finish the rest of the class methods. You might want to leave `insert_position` and `delete_item` for last as they are the most complex.

## Requirements

---

For the code:

- Your program should have the correct comment block at the top (see last assignment)
- Use appropriate comments throughout the code. Be sure to include comments before each function definition explaining the interface (what must be sent, what is returned) and the purpose of the function. Also, don't forget to make appropriate comments within the method definition.
- Make good use of whitespace. Be sure to include a blank line of space before each comment. Two lines of space between functions is nice, but one will suffice – whichever you choose, be consistent. Follow the Python programming style guide: <https://www.python.org/dev/peps/pep-0008/#introduction>

For the lab report, follow the Lab Report Format Guide and complete the following sections:

- Title Page
- Analysis and Conclusions
- Appendix B - code (Be sure that you copy and paste the code into your word document, don't take a screen shot of it and paste that – it ends up being too hard to read.)

## Deliverables

---

Electronic submission in myCourses:

- Code (due by the end of the class period)
- Lab Report (due by the start of the next time that your lab meets) -- either a Microsoft Word document or a pdf

Paper submission in class:

- Lab Report (due at the start of lab the next time that it meets)

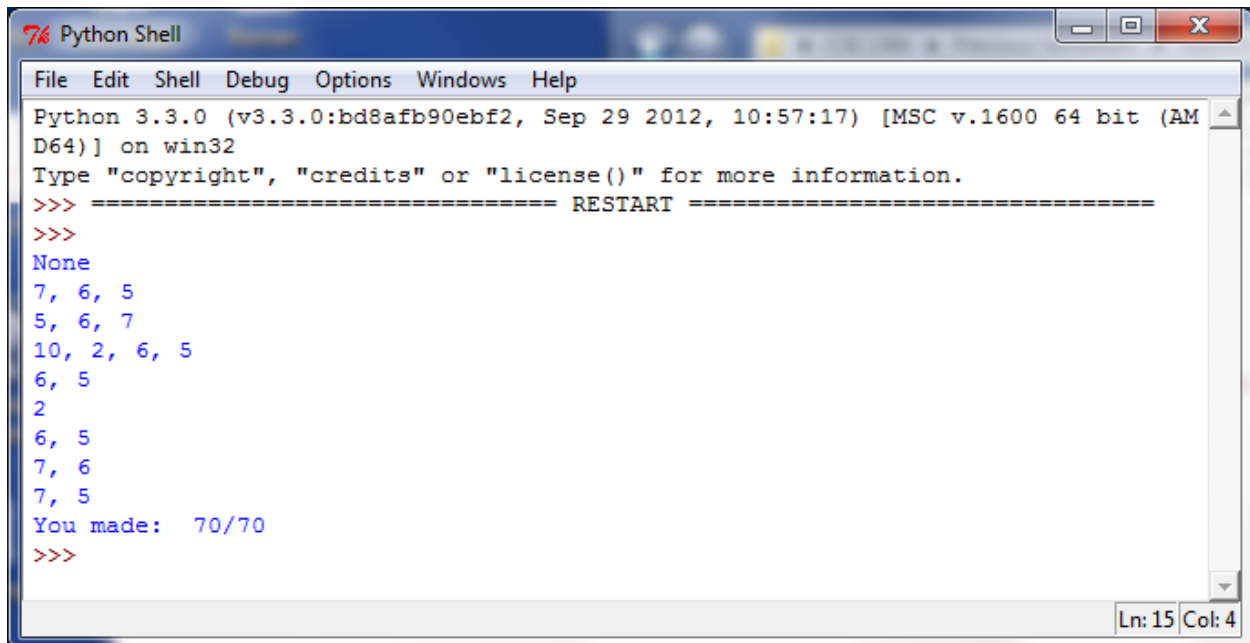
## Grading

---

Task	Points
Lab Report	30 points
Title Page	5 points
Analysis and Conclusions	10 points
Coding Style	15 points
Code – see grader program	70 points

## Sample Execution

---



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:57:17) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
None
7, 6, 5
5, 6, 7
10, 2, 6, 5
6, 5
2
6, 5
7, 6
7, 5
You made: 70/70
>>>
```