**Doubly Linked Lists**

Objectives:

* Understand the differences between singly linked lists and doubly linked lists
* Understand the pros and cons of each
* Construct a doubly linked list using OOP

**Disclaimer**: If you're ahead and have already finished all the mandatory assignments, please work on this assignments before moving on to Flask. Many students mention how they were expected to do linked list on the whiteboard during their technical interviews. If you're on schedule or not so ahead, you can skip this assignment and come back when you have more time.

These exercises are designed to help you prepare for technical interviews and to reinforce concepts you've learned about OOP. If you want to be better prepared for technical interviews, it's helpful to know linked lists and how they are used. Some interesting puzzles can be solved using linked lists (and you may be asked to solve problems using linked lists in technical interviews).

In technical interviews, our alumni are commonly asked problems involving linked lists. Learn about *doubly-linked lists*, also known as *DLists*. We started with *singly-linked lists* because they are simpler, but here's an opportunity to stretch your understanding and learn more by researching doubly-linked lists.

<http://en.wikipedia.org/wiki/Doubly_linked_list> is a great start.

Once you have learned about linked lists, build a class in Python and demonstrate how you can:

* add a new node to the end of the list,
* delete an existing node,
* insert a node between existing nodes (eg. before a given value, at a certain index, etc.)

You should have two classes for this: DoublyLinkedList and Node. Have DoublyLinkedList be the class that allows you to add a new node, delete an existing node, insert a new node between existing nodes, print out the values in the linked list. Have Node be the class that has the necessary properties for the node.

Please also think about the following:

1. How would you know if you have a circular linked list?
2. How would you get to the middle of the linked list?
3. How would you remove duplicate values in the list?
4. How would you reverse the values in the list?

Think hard about these puzzles and how you could potentially use multiple runners to tackle some of these tasks.

Spend up to 5 hours on this assignment.