

# Lab 05: Linked Lists



## Implement a Singly Linked List Data Structure

### Specifications

- Read all of these instructions carefully. Name things exactly as described.
- Do all your work in a public repository called [data-structures-and-algorithms](#), with a well-formatted, detailed top-level README.md.
- Create a new branch in your repo called [linked\\_list](#).
- Your top-level readme should contain a “Table of Contents” navigation to all of your challenges and implementations so far. (Don’t forget to update it!)
- Place this implementation in your [Data-Structures](#) folder within your repository.
- On your branch, create...
  - *C#*: a new console .NET core project named [LinkedList.cs](#).
  - *JavaScript*: a folder named [linkedList](#) which contains a file called [linked-list.js](#)
  - *Python*: a folder named [linked\\_list](#) which contains a file called [linked\\_list.py](#)
  - *Java*: a package named [linkedList](#) which contains a file called [LinkedList.java](#)
- Include any language-specific configuration files required for this challenge to become an individual component, module, library, etc.
  - *NOTE: You can find an example of this configuration for your course in your class lecture repository.*

### Features

- Create a Node class that has properties for the value stored in the Node, and a pointer to the next Node.
- Within your LinkedList class, include a head property. Upon instantiation, an empty Linked List should be created.
  - Define a method called [insert](#) which takes any value as an argument and adds a new node with that value to the [head](#) of the list with an O(1) Time performance.
  - Define a method called [includes](#) which takes any value as an argument and returns a boolean result depending on whether that value exists as a Node’s value somewhere within the list.

- Define a method called `toString` (or `__str__` in Python) which takes in no arguments and returns a string representing all the values in the Linked List.
- At no time should an exception or stack trace be shown to the end user. Catch and handle any such exceptions and return a printed value or operation which cleanly represents the state and either stops execution cleanly, or provides the user with clear direction and output.
- Be sure to follow your language/frameworks standard naming conventions (e.g. C# uses PascalCasing for all method and class names).

## Structure and Testing

Utilize the Single-responsibility principle: any methods you write should be clean, reusable, abstract component parts to the whole challenge. You will be given feedback and marked down if you attempt to define a large, complex algorithm in one function definition.

Write tests to prove the following functionality:

1. Can successfully instantiate an empty linked list
2. Can properly insert into the linked list
3. The head property will properly point to the first node in the linked list
4. Can properly insert multiple nodes into the linked list
5. Will return true when finding a value within the linked list that exists
6. Will return false when searching for a value in the linked list that does not exist
7. Can properly return a collection of all the values that exist in the linked list

Ensure your tests are passing before you submit your solution.

## Stretch Goal

Create a new branch called `doubly_linked_list`, and, using the resources available to you online, implement a doubly linked list (completely separate from your singly linked list).

## Documentation: Your README.md

```
# Singly Linked List
<!-- Short summary or background information -->

## Challenge
<!-- Description of the challenge -->

## Approach & Efficiency
```

```
<!-- What approach did you take? why? What is the Big O space/time for this? -->

## API

<!-- Description of each method publicly available to your Linked List -->
```

## Submission Instructions

1. Create a pull request from your branch to your master branch
2. In your open pull request, leave as a comment a [checklist](#) of the specifications and tasks above, with the actual steps that you completed checked off
3. Submitting your completed work to Canvas:
  1. Copy the link to your open pull request and paste it into the corresponding Canvas assignment
  2. Leave a description of how long this assignment took you in the comments box
  3. Add any additional comments you like about your process or any difficulties you may have had with the assignment
4. Merge your branch into master, and delete your branch (don't worry, the PR link will still work)