Challenge 4: Find the Length of a Linked List

Let's write a function which can tell us the length of a linked list.

We'll cover the following

- Problem Statement
  - Input
  - Output
  - Sample Input
  - Sample Output
- Coding Exercise

Problem Statement

In this problem, you have to implement the `length()` function which will find the length of a given linked list.

Input

A linked list.

Output

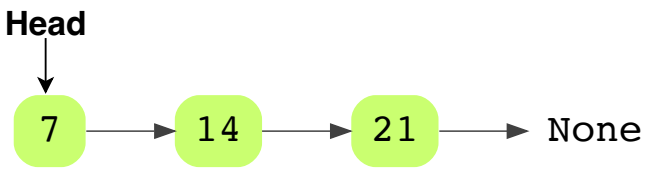
The number of nodes in the list.

Sample Input

```
linkedlist = 0->1->2->3->4
```

Sample Output

```
5
```



Length of List = # of Nodes = 3

Coding Exercise

Hopefully, this isn't a very tough task because we've seen much more complex algorithms in the previous lessons. Try it out on a piece of paper to get the logic right.

The full `LinkedList` and `Node` classes are available to you along with all their member functions.

Good Luck!

main.py

LinkedList.py

Node.py

```
from Node import Node
from LinkedList import LinkedList
# Access head_node => list.get_head()
# Check if list is empty => list.is_empty()
# Delete at head => list.delete_at_head()
# Delete by value => list.delete(value)
# Search for element => list.search()
# Node class attributes: {data, next_element}


def length(lst):
    # Write - Your - Code
    count = 0
    if lst.is_empty():
        return count
    else:
        current_node = lst.get_head()
        while current_node is not None:
            count += 1
            current_node = current_node.next_element
    return count
```


Back

Singly Linked Lists vs. Doubly Linked ...

Next

Solution Review: Find the Length of a ...

Completed