Create a Linked List Class

Let's create a `linked list` class. Every linked list should start out with a few basic properties: a `head` (the first item in your list) and a `length` (number of items in your list). Sometimes you'll see implementations of linked lists that incorporate a `tail` for the last element of the list, but for now we'll just stick with these two. Whenever we add an element to the linked list, our `length` property should be incremented by one.

We'll want to have a way to add items to our linked list, so the first method we'll want to create is the `add` method.

If our list is empty, adding an element to our linked list is straightforward enough: we just wrap that element in a `Node` class, and we assign that node to the `head` of our linked list.

But what if our list already has one or more members? How do we add an element to the list? Recall that each node in a linked list has a `next` property. To add a node to the list, find the last node in the list, and point that last node's `next` property at our new node. (Hint: you know you've reached the end of a linked list when a node's `next` property is `null`.)

Write an add method that assigns the first node you push to the linked list to the `head`; after that, whenever adding a node, every node should be referenced by the previous node's `next` property.

Note: Your list's `length` should increase by one every time an element is added to the linked list.

	Your `LinkedList` class should have an `add` method.
	Your `LinkedList` class should assign `head` to the first node added.

	The previous `node` in your `LinkedList` class should have a reference to the newest node created.
	The `size` in your `LinkedList` class should equal the amount of nodes in the linked list.