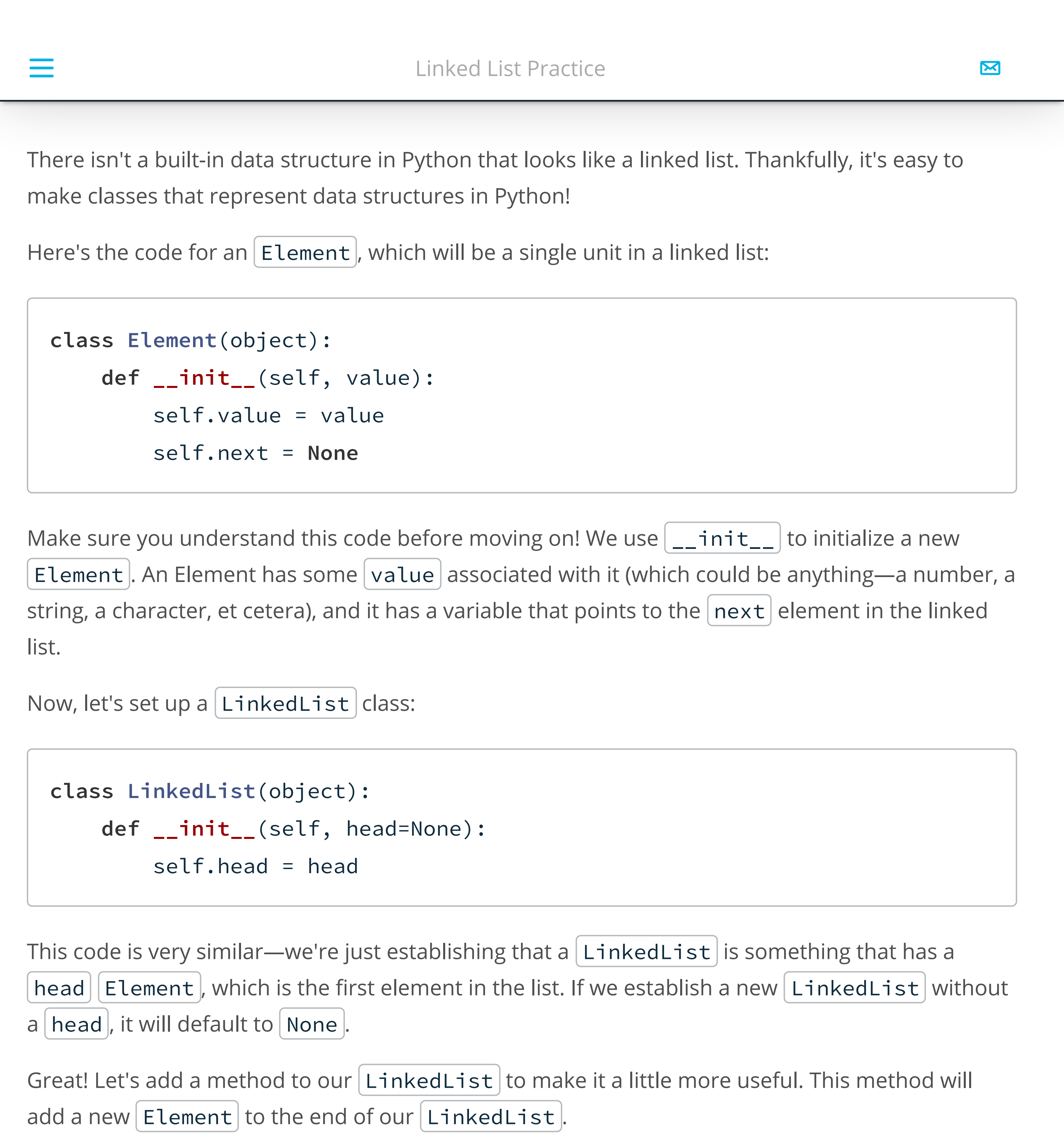

Linked List Practice

There isn't a built-in data structure in Python that looks like a linked list. Thankfully, it's easy to make classes that represent data structures in Python!

Here's the code for an `Element`, which will be a single unit in a linked list:

```
class Element(object):
    def __init__(self, value):
        self.value = value
        self.next = None
```

Make sure you understand this code before moving on! We use `__init__` to initialize a new `Element`. An Element has some `value` associated with it (which could be anything—a number, a string, a character, et cetera), and it has a variable that points to the `next` element in the linked list.

Now, let's set up a `LinkedList` class:

```
class LinkedList(object):
    def __init__(self, head=None):
        self.head = head
```

This code is very similar—we're just establishing that a `LinkedList` is something that has a `head` `Element`, which is the first element in the list. If we establish a new `LinkedList` without a `head`, it will default to `None`.

Great! Let's add a method to our `LinkedList` to make it a little more useful. This method will add a new `Element` to the end of our `LinkedList`.