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.

**def** **append**(self, new\_element):

current = self.head

**if** self.head:

**while** current.next:

current = current.next

current.next = new\_element

**else**:

self.head = new\_element

Again, this part is really important, so don't rush through it. Take the code line-by-line and make sure everything makes sense. If the LinkedList already has a head, iterate through the next reference in every Element until you reach the end of the list. Set next for the end of the list to be the new\_element. Alternatively, if there is no head already, you should just assign new\_element to it and do nothing else.

**"""The LinkedList code from before is provided below.**

**Add three functions to the LinkedList.**

**"get\_position" returns the element at a certain position.**

**The "insert" function will add an element to a particular**

**spot in the list.**

**"delete" will delete the first element with that**

**particular value."""**

**class** **Element**(object):

**def** **\_\_init\_\_**(self, value):

self.value = value

self.next = **None**

**class** **LinkedList**(object):

**def** **\_\_init\_\_**(self, head=None):

self.head = head

**def** **append**(self, new\_element):

current = self.head

**if** self.head:

**while** current.next:

current = current.next

current.next = new\_element

**else**:

self.head = new\_element

**def** **get\_position**(self, position):

counter = 1

current = self.head

**if** position < 1:

**return** **None**

**while** current **and** counter <= position:

**if** counter == position:

**return** current

current = current.next

counter += 1

**return** **None**

**def** **insert**(self, new\_element, position):

counter = 1

current = self.head

**if** position > 1:

**while** current **and** counter < position:

**if** counter == position - 1:

new\_element.next = current.next

current.next = new\_element

current = current.next

counter += 1

**elif** position == 1:

new\_element.next = self.head

self.head = new\_element

**def** **delete**(self, value):

current = self.head

previous = **None**

**while** current.value != value **and** current.next:

previous = current

current = current.next

**if** current.value == value:

**if** previous:

previous.next = current.next

**else**:

self.head = current.next