We build a self-referential class

1. Questions?

2. Let's build a new container class that is composed of a number of Element objects that are linked together.

3. Each Element:
   (a) Carries a little bit of data, in .value.
   (b) Connects to a chain of other Elements through .next.
   (c) When .next is None, we take that to mean “there are no other elements following this element.”
   (d) We'd like to be able to have properties for value and next.
   (e) We'd like to be able to set the value of next.
   (f) Support for str and repr.

4. We'll build a wrapper class, called LinkedList. This class is the public interface to organizing Element objects.
   (a) Contains a single reference, head, to the first Element of the list that holds its data.
   (b) If the head is None, the list is considered empty.
   (c) Has an implementation of append(item).
   (d) Has an implementation of .iter_, which generates all the values stored in the elements of the list, in order. Great for for loops.
   (e) Support for len(l) (._len_), computing the length of the list.
   (f) Support for in (._contains_), detecting membership,
   (g) Support for comparing LinkedLists via .eq_.
   (h) Support for indexing (._getitem_) or setting (._setitem_) element i.
   (i) Support for str (._str_) and repr (._repr_).

*