

Computer Science 134C

Introduction to Computer Science, in Python

Lecture #20 (Classes V)

Octo-boo!-r 31

Keywords linked lists

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__`).