6.009: Fundamentals of Programming

Week 8 Lecture: Custom Types and Linked Structures

Adam Hartz

hz@mit.edu

6 April 2020

6.009: Last Time

Implementing custom types using the `class` keyword

Updating our environment model to handle classes and instances

Review: Variable and Attribute Lookup

Looking up a *variable* (in a function call, etc):

1. look in the current frame first
2. if not found, look in the *parent* frame
3. if not found, look in that's frame's parent frame
4. ...
5. if not found, look in the global frame
6. if not found, look in the builtins
7. if not found, raise a `NameError`

Review: Variable and Attribute Lookup

Looking up an *attribute* (in an object with ”dot” notation):

1. look in the object itself (the instance)
2. if not found, look in the object’s *class*
3. if not found, look in that class’s superclass
4. if not found, look in *that* class’s superclass
5. ...
6. if not found and no more superclasses, raise an `AttributeError`

`self`

Additional weirdness: when looking up a class's method by way of an instance, that instance will automatically be passed in as the first argument.

For example, the following two pieces of code will do the same thing, if x is an instance of the class *Thing*:

```
Thing.foo(x, 1, 2, 3)
```

```
x.foo(1, 2, 3)
```

By convention, this first parameter is usually called `self`. It's a good idea to follow that convention, even though it's not strictly necessary to do so.

Integrating More Closely With Python

Python offers ways to integrate things more tightly into the language: “magic” methods or “dunder” methods. For example:

- `print(x)` is translated implicitly to `print(x.__str__())`
- `abs(x)` is translated implicitly to `x.__abs__()`
- `x + y` is translated implicitly to `x.__add__(y)`
- `x - y` is translated implicitly to `x.__sub__(y)`
- `x[y]` is translated implicitly to `x.__getitem__(y)`
- `x[y] = z` is translated implicitly to `x.__setitem__(y, z)`

For a full list, see:

`https://docs.python.org/3/reference/datamodel.html`, Section 3.3

Today: Linked List

Different way of implementing lists. Example of a ”linked” data structure.

Example: a linked list containing 4, 8, 15, 16, 23, and 42: