20.11. Thinking About Classes and Instances

You can now imagine some reasons you may want to define a class. You have seen examples of creating types that are more complicated or specific than the ones built in to Python (like lists or strings). `Turtle`, with all the instance variables and methods you learned about using earlier in the semester, is a class that programmers defined which is now included in the Python language. In this chapter, we defined `Point` with some functionality that can make it easier to write programs that involve `x,y` coordinate `Point` instances. And shortly, you'll see how you can define classes to represent objects in a game.

You can also use self-defined classes to hold data – for example, data you get from making a request to a REST API.

Before you decide to define a new class, there are a few things to keep in mind, and questions you should ask yourself:

- **What is the data that you want to deal with?** (Data about a bunch of songs from iTunes? Data about a bunch of tweets from Twitter? Data about a bunch of hashtag searches on Twitter? Two numbers that represent coordinates of a point on a 2-dimensional plane?)
- **What will one instance of your class represent?** In other words, which sort of new *thing* in your program should have fancy functionality? One song? One hashtag? One tweet? One point? The answer to this question should help you decide what to call the class you define.
- **What information should each instance have as instance variables?** This is related to what an instance represents. See if you can make it into a sentence. *"Each instance represents one < song > and each < song > has an < artist > and a < title > as instance variables."* Or, *"Each instance represents a < Tweet > and each < Tweet > has a < user (who posted it) > and < a message content string > as instance variables."*
- **What instance methods should each instance have?** What should each instance be able to *do*? To continue using the same examples: Maybe each song has a method that uses a lyrics API to get a long string of its lyrics. Maybe each song has a method that returns a string of its artist's name. Or for a tweet, maybe each tweet has a method that returns the length of the tweet's message. (Go wild!)
- **What should the printed version of an instance look like?** (This question will help you determine how to write the `__str__` method.) Maybe, "Each song printed out will show the song title and the artist's name." or "Each Tweet printed out will show the username of the person who posted it and the message content of the tweet."

After considering those questions and making decisions about how you're going to get started with a class definition, you can begin to define your class.

Remember that a class definition, like a function definition, is a general description of what *every instance of the class should have*. (Every Point has an `x` and a `y`.) The class instances are specific: e.g. the Point with *a specific x and y* >. You might have a Point with an `x` value of 3 and a `y` value of 2, so for that particular *instance* of the *class* `Point`, you'd pass in `3` and `2` to the constructor, the `__init__` method, like so: `new_point = Point(3,2)`, as you saw in the last sections.

You have attempted 1 of 1 activities on this page

20.10. Class Variables and Instance Variables">

20.12. Testing classes">

Class Variables and Instance Variables">

✔ Completed. Well Done!

20.12. Testing classes">Next Section - 20.12. Testing classes

| Back to top