Special Methods

Instead of creating classes with empty or default values, we can set these values when we create the instance. This ensures that we don't miss an important value and avoids a lot of unnecessary lines of code. To do this, we use a special method called a **constructor**. Below is an example of an Apple class with a constructor method defined.

```
>>> class Apple:
...     def __init__(self, color, flavor):
...         self.color = color
...         self.flavor = flavor
```

When you call the name of a class, the constructor of that class is called. This constructor method is always named **___init___**. You might remember that special methods start and end with two underscore characters. In our example above, the constructor method takes the self variable, which represents the instance, as well as color and flavor parameters. These parameters are then used by the constructor method to set the values for the current instance. So we can now create a new instance of the Apple class and set the color and flavor values all in go:

```
>>> jonagold = Apple("red", "sweet")
>>> print(jonagold.color)
Red
```

In addition to the **___init___** constructor special method, there is also the **___str___** special method. This method allows us to define how an instance of an object will be printed when it's passed to the print() function. If an object doesn't have this special method defined, it will wind up using the default representation, which will print the position of the object in memory. Not super useful. Here is our Apple class, with the **___str___** method added:

```
>>> class Apple:
...     def __init__(self, color, flavor):
...         self.color = color
...         self.flavor = flavor
...     def __str__(self):
...         return "This apple is {} and its flavor is {}".format(self.color, self.flavor)
...
```

Now, when we pass an Apple object to the print function, we get a nice formatted string:

```
>>> jonagold = Apple("red", "sweet")
>>> print(jonagold)
This apple is red and its flavor is sweet
```

This apple is red and its flavor is sweet

It's good practice to think about how your class might be used and to define a __str__ method when creating objects that you may want to print later.