## 20.7. Converting an Object to a String

When we're working with classes and objects, it is often necessary to print an object (that is, to print the state of an object). Consider the example below.

```
Save & Run
                            Download
                                         Load History
                                                         Show CodeLens
 1 class Point:
       """ Point class for representing and manipulating x,y coordinates. """
 2
 3
       def __init__(self, initX, initY):
           self.x = initX
 4
 5
           self.y = initY
 6
      def getX(self):
 7
           return self.x
 8
      def getY(self):
 9
           return self.y
10
      def distanceFromOrigin(self):
           return ((self.x ** 2) + (self.y ** 2)) ** 0.5
12p = Point(7,6)
13 print(p)
14
                            ActiveCode: 1 (chp13 classesstr1)
```

The print function shown above produces a string representation of the Point  $\,p$ . The default functionality provided by Python tells you that  $\,p$  is an object of type Point . However, it does not tell you anything about the specific state of the point.

We can improve on this representation if we include a special method call \_\_str\_\_ . Notice that this method uses the same naming convention as the constructor, that is two underscores before and after the name. It is common that Python uses this naming technique for special methods.

The \_\_str\_\_ method is responsible for returning a string representation as defined by the class creator. In other words, you as the programmer, get to choose what a Point should look like when it gets printed. In this case, we have decided that the string representation will include the values of x and y as well as some identifying text. It is required that the \_\_str\_\_ method create and *return* a string.

Whatever string the \_\_str\_\_ method for a class returns, that is the string that will print when you put any instance of that class in a print statement. For that reason, the string that a class's \_\_str\_\_ method sasArgtetterschafte

Take a look at the code below.

1 of 3 11/12/19, 8:22 pm

```
Save & Run
                            Download
                                         Load History
                                                        Show CodeLens
 1 class Point:
       """ Point class for representing and manipulating x,y coordinates. """
 2
 3
      def __init__(self, initX, initY):
           self.x = initX
 4
 5
           self.y = initY
 6
      def getX(self):
 7
           return self.x
      def getY(self):
 8
 9
           return self.y
      def distanceFromOrigin(self):
10
           return ((self.x ** 2) + (self.y ** 2)) ** 0.5
11
12
      def __str__(self):
           return "x = \{\}, y = \{\}".format(self.x, self.y)
13
14 p = Point(7,6)
15 print(p)
16
```

ActiveCode: 2 (chp13\_classesstr2)

When we run the program above you can see that the print function now shows the string that we chose.

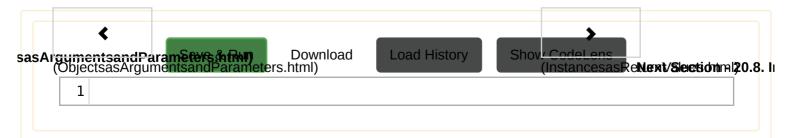
Now, you ask, don't we already have a str type converter that can turn our object into a string? Yes we do!

And doesn't print automatically use this when printing things? Yes again!

However, as we saw earlier, these automatic mechanisms do not do exactly what we want. Python provides many default implementations for methods that we as programmers will probably want to change. When a programmer changes the meaning of a method we say that we **override** the method. Note also that the str type converter function uses whatever \_\_str\_\_ method we provide.

## **Check Your Understanding**

1. Create a class called Cereal that accepts three inputs: 2 strings and 1 integer, and assigns them to 3 instance variables in the constructor: name, brand, and fiber. When an instance of Cereal is printed, the user should see the following: "[name] cereal is produced by [brand] and has [fiber integer] grams of fiber in every serving!" To the variable name c1, assign an instance of Cereal whose name is "Corn Flakes", brand is "Kellogg's", and fiber is 2. To the variable name c2, assign an instance of Cereal whose name is "Honey Nut Cheerios", brand is "General Mills", and fiber is 3. Practice printing both!



2 of 3 11/12/19, 8:22 pm

ActiveCode: 3 (ac\_ch13\_classstr\_01)

Mark as completed

© Copyright 2017 bradleymiller. Created using Runestone (http://runestoneinteractive.org/) 3.2.15.

 $\textbf{username: sanjaysheel 1997} @ \textbf{gmail.com} \mid \texttt{Back to top}$ 

sasArgumentsandParameters.html)
(ObjectsasArgumentsandParameters.html)

**>** 

(InstancesasReNexit/Sleetsionml20.8. II