

20.8. Instances as Return Values

Functions and methods can return objects. This is actually nothing new since everything in Python is an object and we have been returning values for quite some time. (You can also have lists or tuples of object instances, etc.) The difference here is that we want to have the method create an object using the constructor and then return it as the value of the method.

Suppose you have a point object and wish to find the midpoint halfway between it and some other target point. We would like to write a method, let's call it `halfway`, which takes another `Point` as a parameter and returns the `Point` that is halfway between the point and the target point it accepts as input.

```
1 class Point:
2     def __init__(self, initX, initY):
3         self.x = initX
4         self.y = initY
5     def getX(self):
6         return self.x
7     def getY(self):
8         return self.y
9     def distanceFromOrigin(self):
10        return ((self.x ** 2) + (self.y ** 2)) ** 0.5
11    def __str__(self):
12        return "x = {}, y = {}".format(self.x, self.y)
13    def halfway(self, target):
14        mx = (self.x + target.x)/2
15        my = (self.y + target.y)/2
16        return Point(mx, my)
17 p = Point(3,4)
18 q = Point(5,12)
19 mid = p.halfway(q)
20 # note that you would have exactly the same result if you instead wrote
21 # mid = q.halfway(p)
22 # because they are both Point objects, and the middle is the same no matter what
23 print(mid)
24 print(mid.getX())
25 print(mid.getY())
26
```

ActiveCode: 1 (chp13_classesmid1)

The resulting Point, `mid`, has an x value of 4 and a y value of 8. We can also use any other methods on `mid` since it is a `Point` object.

Mark as completed

