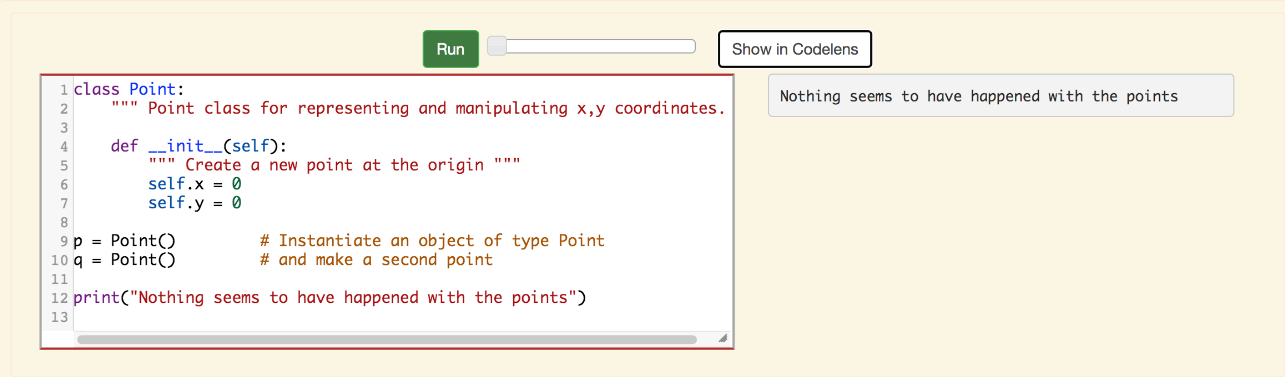
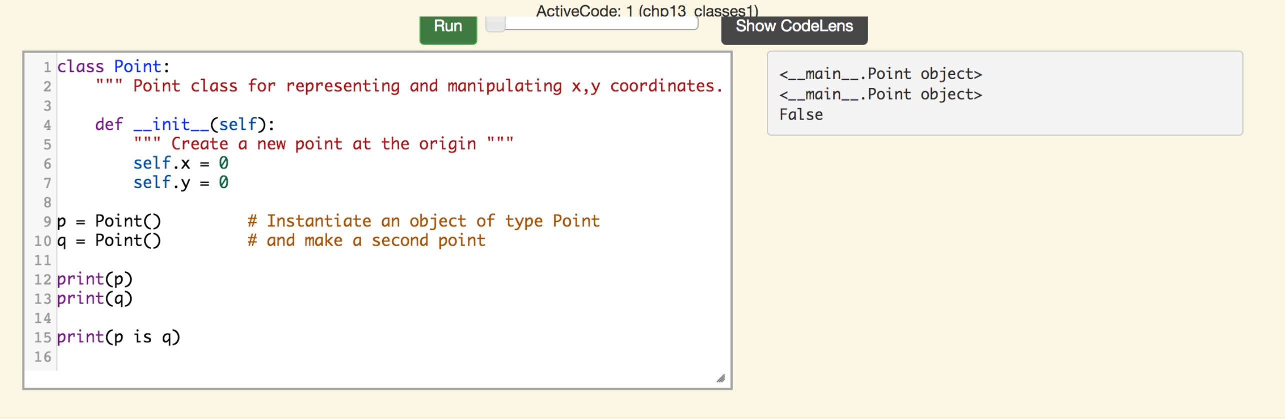
So let's use our new Point class now.



```python
class Point:
    """ Point class for representing and manipulating x,y coordinates.

    def __init__(self):
        """ Create a new point at the origin """
        self.x = 0
        self.y = 0

p = Point()         # Instantiate an object of type Point
q = Point()         # and make a second point

print("Nothing seems to have happened with the points")

```

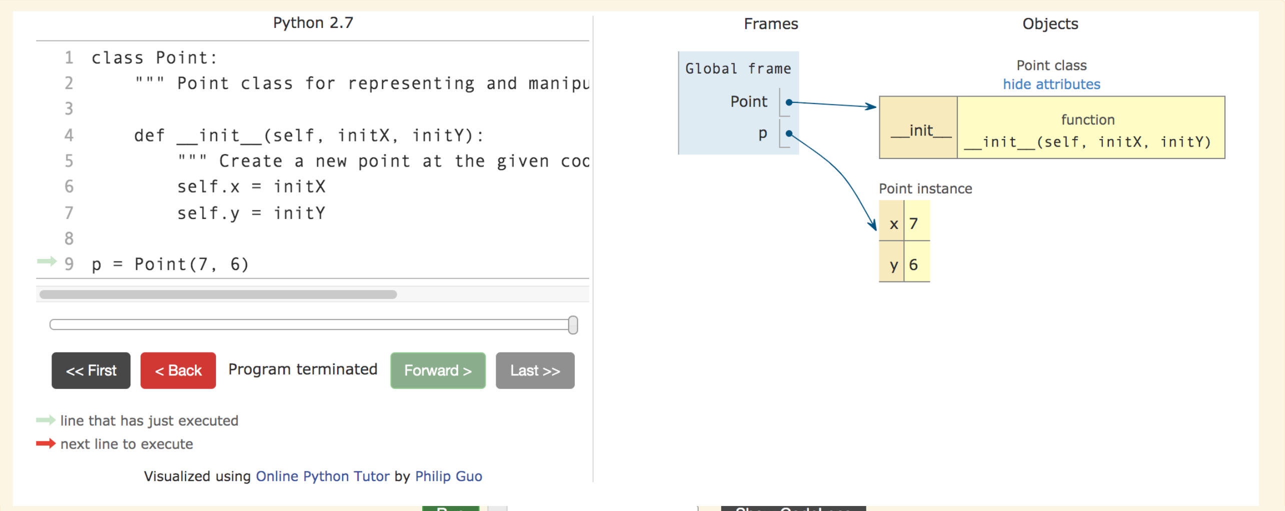
```
Nothing seems to have happened with the points
```

ActiveCode: 1 (chp13_classes1)



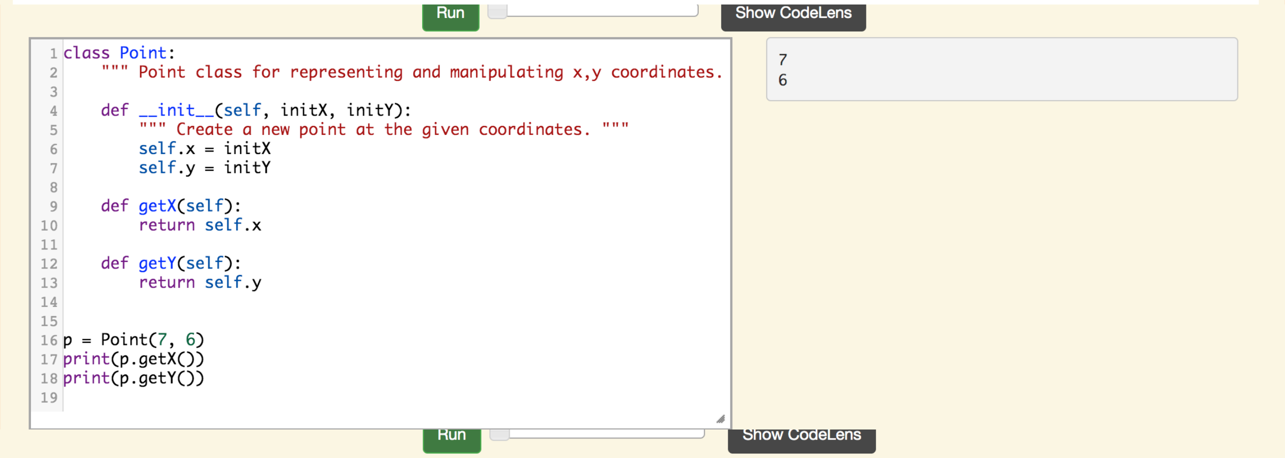
```python
class Point:
    """ Point class for representing and manipulating x,y coordinates.

    def __init__(self):
        """ Create a new point at the origin """
        self.x = 0
        self.y = 0

p = Point()         # Instantiate an object of type Point
q = Point()         # and make a second point

print(p)
print(q)

print(p is q)

```

```
<__main__.Point object>
<__main__.Point object>
False
```



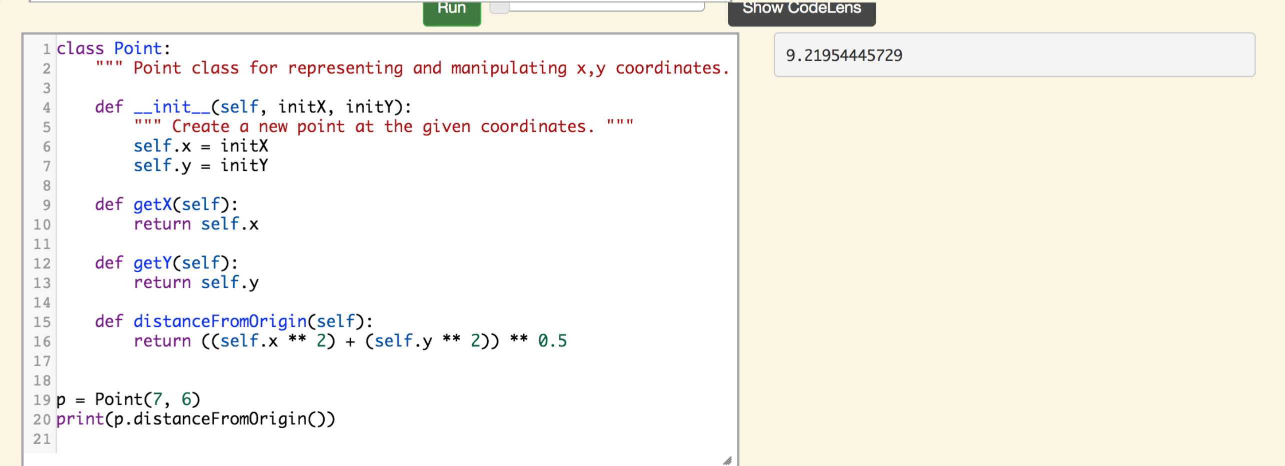
```python
class Point:
    """ Point class for representing and manipu

    def __init__(self, initX, initY):
        """ Create a new point at the given coc
        self.x = initX
        self.y = initY

p = Point(7, 6)
```

Visualized using Online Python Tutor by Philip Guo



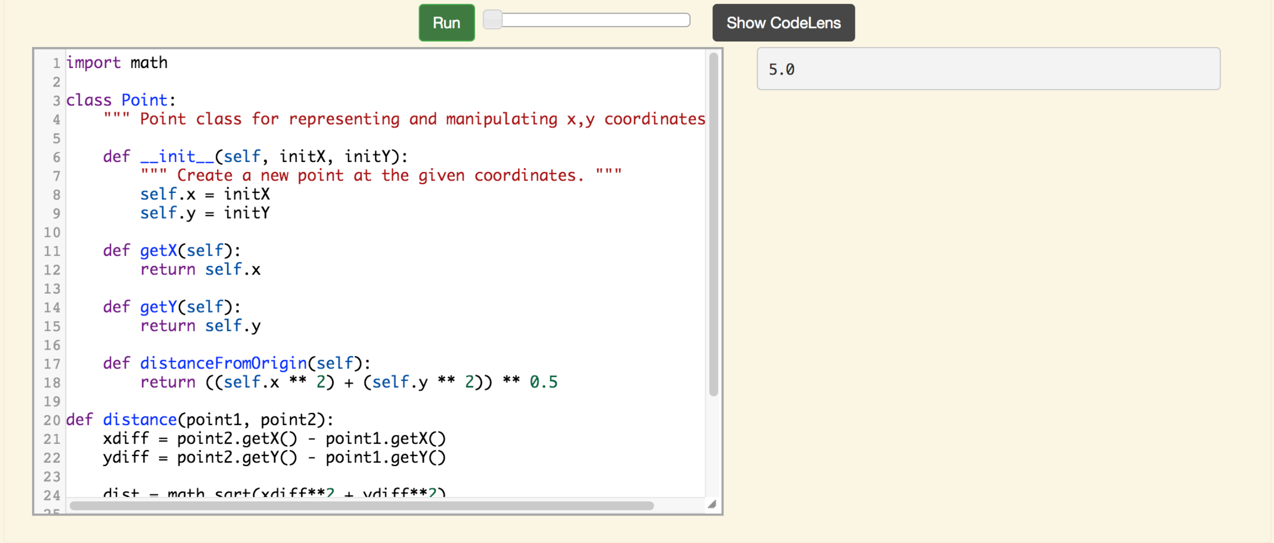
```python
class Point:
    """ Point class for representing and manipulating x,y coordinates.

    def __init__(self, initX, initY):
        """ Create a new point at the given coordinates. """
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y


p = Point(7, 6)
print(p.getX())
print(p.getY())

```

```
7
6
```



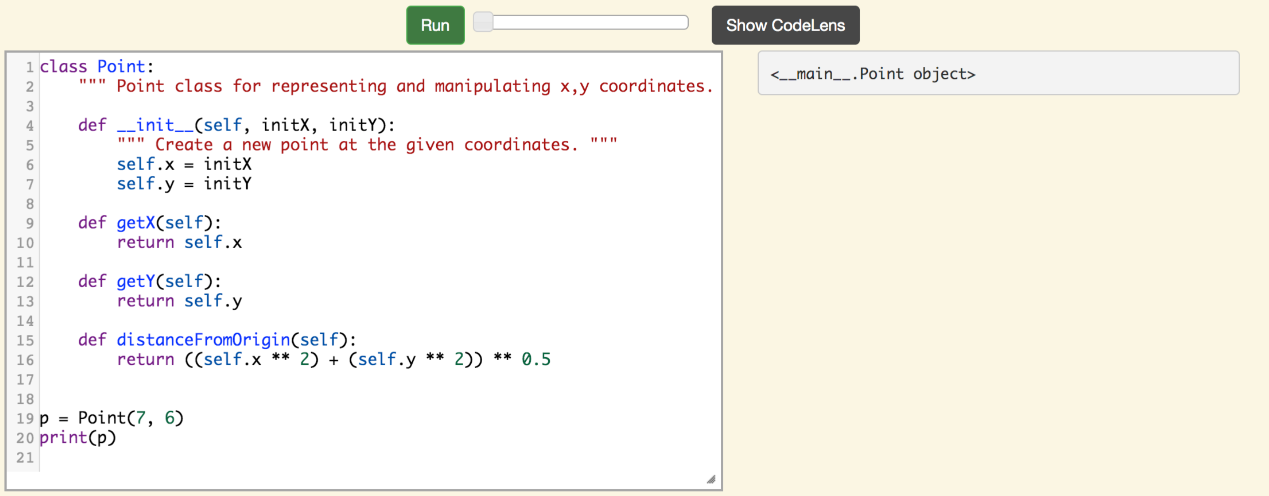
```python
class Point:
    """ Point class for representing and manipulating x,y coordinates.

    def __init__(self, initX, initY):
        """ Create a new point at the given coordinates. """
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def distanceFromOrigin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5


p = Point(7, 6)
print(p.distanceFromOrigin())

```

```
9.21954445729
```



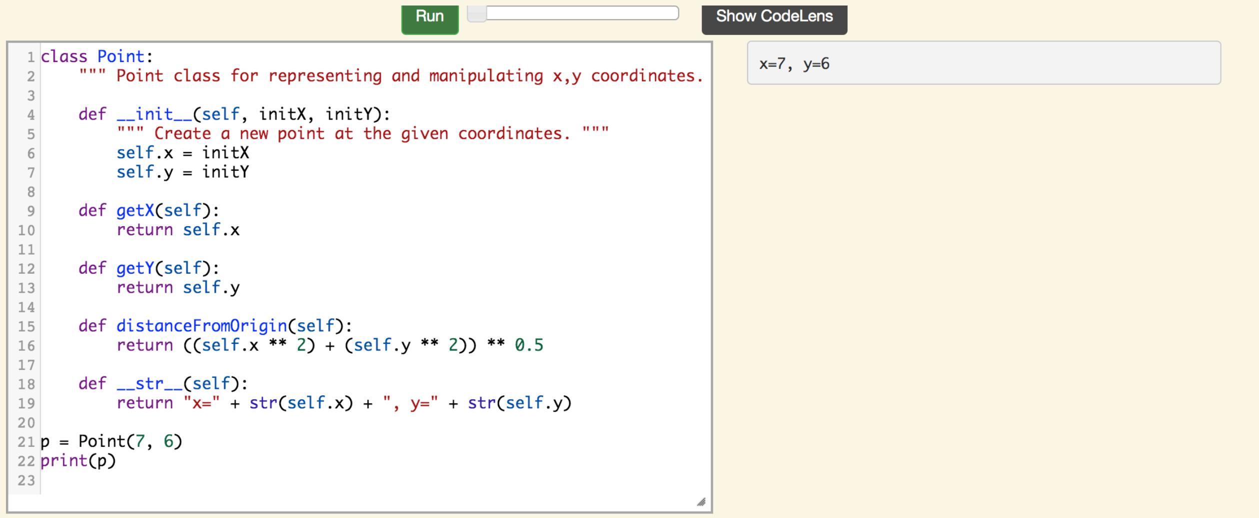
```python
import math

class Point:
    """ Point class for representing and manipulating x,y coordinates

    def __init__(self, initX, initY):
        """ Create a new point at the given coordinates. """
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def distanceFromOrigin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5

def distance(point1, point2):
    xdiff = point2.getX() - point1.getX()
    ydiff = point2.getY() - point1.getY()

    dist = math.sqrt(xdiff**2 + ydiff**2)
```

```
5.0
```



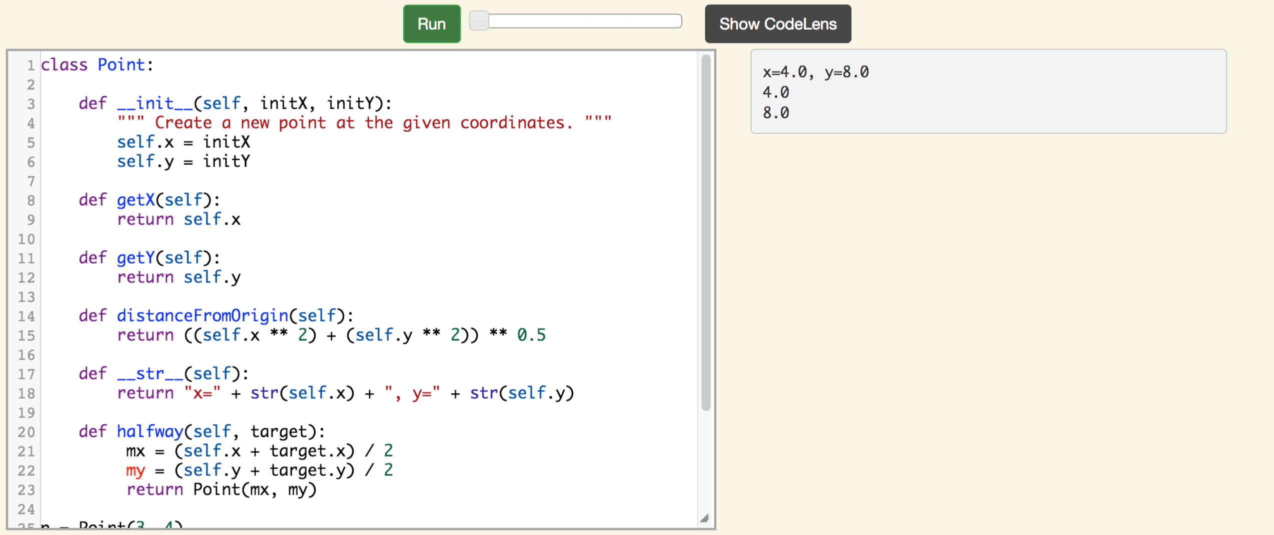
```python
class Point:
    """ Point class for representing and manipulating x,y coordinates.

    def __init__(self, initX, initY):
        """ Create a new point at the given coordinates. """
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def distanceFromOrigin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5


p = Point(7, 6)
print(p)

```

```
<__main__.Point object>
```



```python
class Point:
    """ Point class for representing and manipulating x,y coordinates.

    def __init__(self, initX, initY):
        """ Create a new point at the given coordinates. """
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def distanceFromOrigin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5

    def __str__(self):
        return "x=" + str(self.x) + ", y=" + str(self.y)

p = Point(7, 6)
print(p)

```

```
x=7, y=6
```



```python
class Point:

    def __init__(self, initX, initY):
        """ Create a new point at the given coordinates. """
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y

    def distanceFromOrigin(self):
        return ((self.x ** 2) + (self.y ** 2)) ** 0.5

    def __str__(self):
        return "x=" + str(self.x) + ", y=" + str(self.y)

    def halfway(self, target):
         mx = (self.x + target.x) / 2
         my = (self.y + target.y) / 2
         return Point(mx, my)

p = Point(3, 4)
```

```
x=4.0, y=8.0
4.0
8.0
```

ActiveCode: 1 (chp12_classesmid1)