

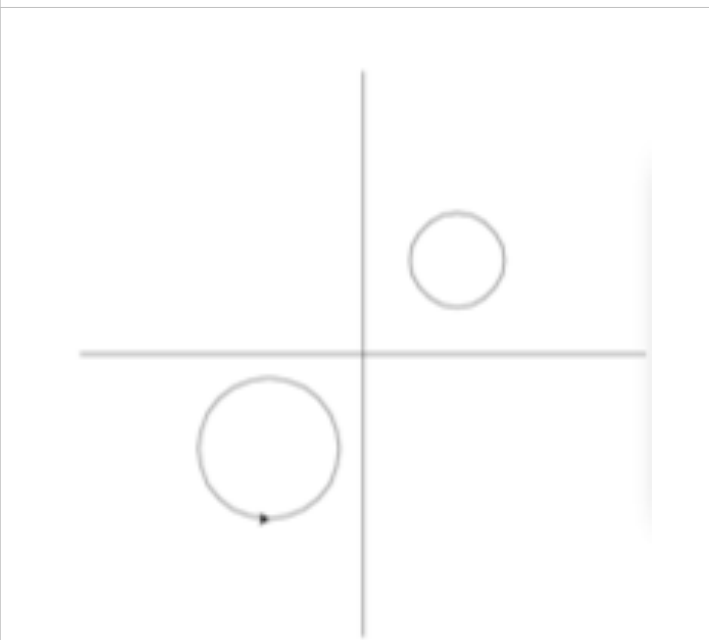
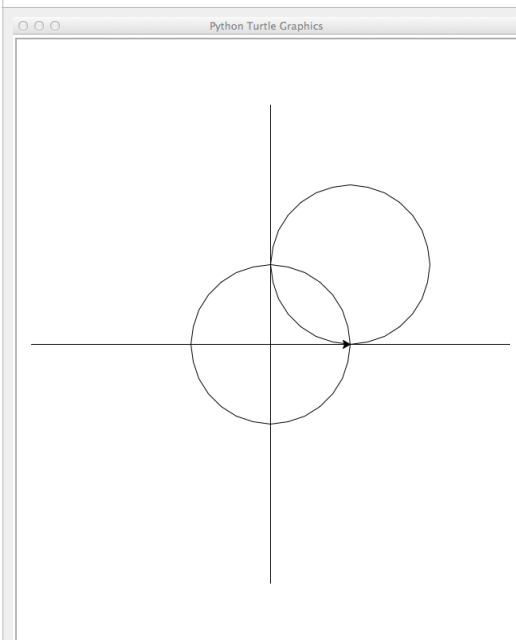
Overlapping Circles Hint

Sample Run:

```
>>> import overlap
>>> overlap.circles()
Enter x for Circle 1's center: 0
Enter y for Circle 1's center: 0
Enter radius for Circle 1: 100
Enter x for Circle 2's center: 100
Enter y for Circle 2's center: 100
Enter radius for Circle 2: 100
Circles overlap.
0
>>> quit()
```

Sample Run:

```
>>> import overlap
>>> overlap.circles()
Enter x for Circle 1's center: 100
Enter y for Circle 1's center: 100
Enter radius for Circle 1: 50
Enter x for Circle 2's center: -100
Enter y for Circle 2's center: -100
Enter radius for Circle 2: 75
Circles do not overlap.
0
```



Sample run with an error:

```
import overlap
>>> overlap.circles()
Enter x for Circle 1's center: 0
Enter y for Circle 1's center: 0
Enter radius for Circle 1: 0
Radii must be greater than zero.
```

Code:

```
import math
import turtle

def circles():
    t = turtle.Turtle()

    # draw the x and y axes

    t.goto(0, 300)
    t.down()
    t.goto(0, -300)
    t.up()
    t.goto(-300, 0)
    t.down()
    t.goto(300, 0)

    circle1x = input("Enter x for Circle 1's center: ")
    circle1y = input("Enter y for Circle 1's center: ")

    circle1Radius = input("Enter radius for Circle 1: ")

    if circle1Radius <= 0:
        print "Radii must be greater than zero."
        quit() # Is there a better way?

    t.up()
    t.goto(0,0)

    t.goto(circle1x, circle1y - circle1Radius)
    t.down()

    t.circle(circle1Radius)

    t.up()

    circle2x = input("Enter x for Circle 2's center: ")

    circle2y = input("Enter y for Circle 2's center: ")
    circle2Radius = input("Enter radius for Circle 2: ")

    if circle2Radius <= 0:
        print "Radii must be greater than zero."
        quit() # Is there a better way?

    t.goto(circle2x, circle2y-circle2Radius)

    t.down()

    t.circle(circle2Radius)

    sumOfRadii = circle1Radius + circle2Radius

    distance = math.sqrt(math.pow(circle1x - circle2x,2) + math.pow(circle1y - circle2y,2))

    if distance <= sumOfRadii:
        print "Circles overlap."
    else:
        print "Circles do not overlap."

    return 0
```

Additional work is still required, specifically:

- Support the "use case" where the circles are kissing. . .