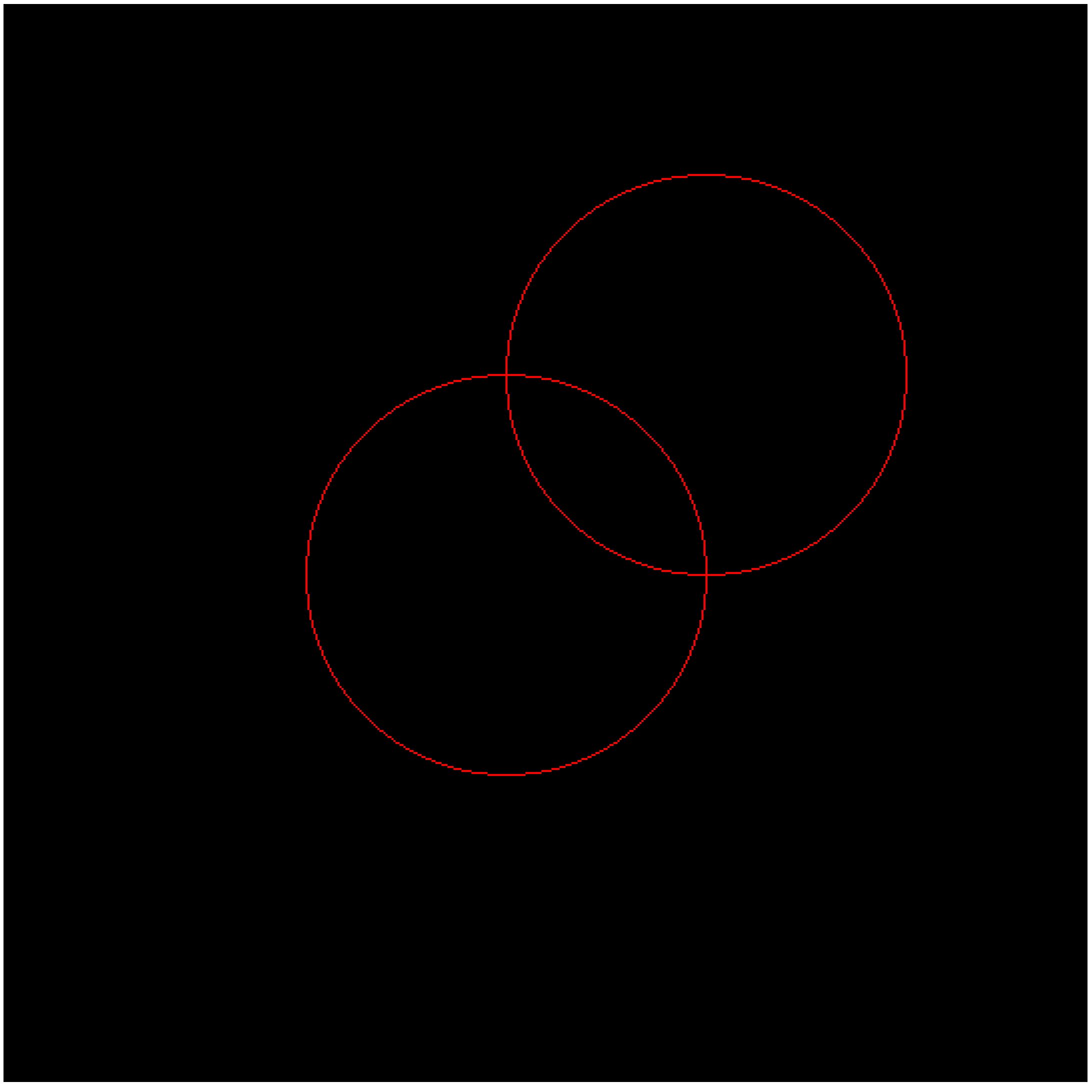
```
def eightway_symetry(x, y, a, b):
    addPixel(x + a, y + b)
    addPixel(y + a, x + b)
    addPixel(y + a, -x + b)
    addPixel(x + a, -y + b)
    addPixel(-x + a, -y + b)
    addPixel(-y + a, -x + b)
    addPixel(-y + a, x + b)
    addPixel(-x + a, y + b)

def midpoint_circle(radius,centerX = 0, centerY = 0):
    gl.glBegin(gl.GL_POINTS)
    x = 0
    y = radius
    d = 1 - radius
    while x <= y:
        eightway_symetry(x,y,centerX,centerY)
        if d < 0:
            d += 2 * x + 3
        else:
            d += 2 * (x - y) + 5
            y -= 1
        x += 1
    gl.glEnd()



# Let's render something!
gl.glClear(gl.GL_COLOR_BUFFER_BIT)


gl.glColor3f(1,0,0)
gl.glPointSize(1)

midpoint_circle(100)
midpoint_circle(100,100,100)
```



```
from math import radians as rad
from math import sin, cos
def circles(n = 8, r = 200, centerx = 0, centery = 0):
    midpoint_circle(r,centerx, centery)
    theta = 0
    dt = 360/n
    small_r = r//2
    for i in range(n):
      newCenterx = small_r*cos(rad(theta))
      newCentery = small_r*sin(rad(theta))
      midpoint_circle(small_r,newCenterx + centerx, newCentery + centery)
      theta += dt
```

```
gl.glClear(gl.GL_COLOR_BUFFER_BIT)
gl.glColor3f(1,0,0)
gl.glPointSize(1)

circles(5, r =  150)

img_buf = gl.glReadPixelsub(0, 0, WIDTH, HEIGHT, gl.GL_RGB,
gl.GL_UNSIGNED_BYTE)
img = np.frombuffer(img_buf, np.uint8).reshape(HEIGHT, WIDTH, 3)[::-1]
show.image(img/255.0)
```