

translate
& rotate

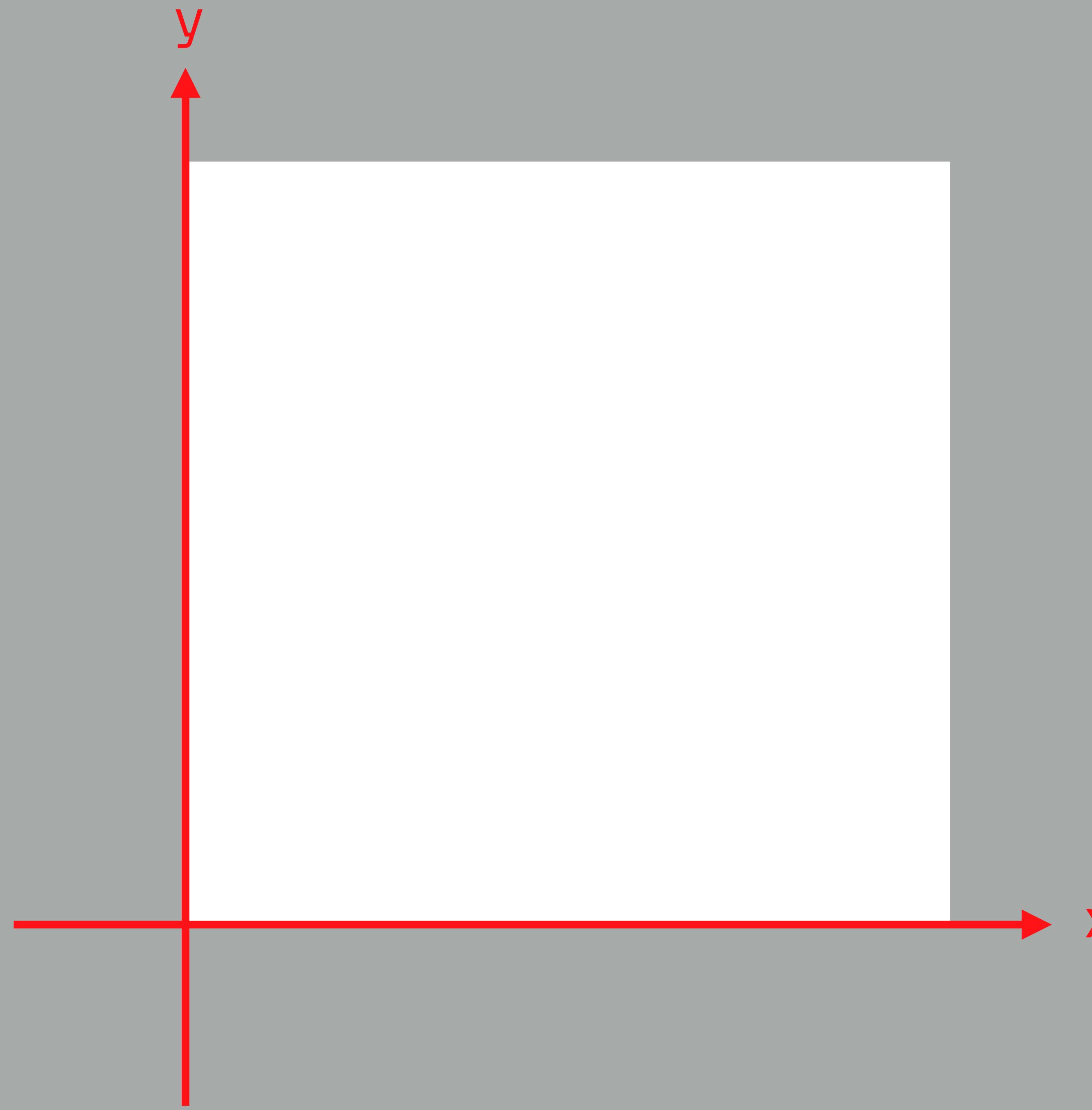
Untitled 7

Run Comment Uncomment Indent Dedent

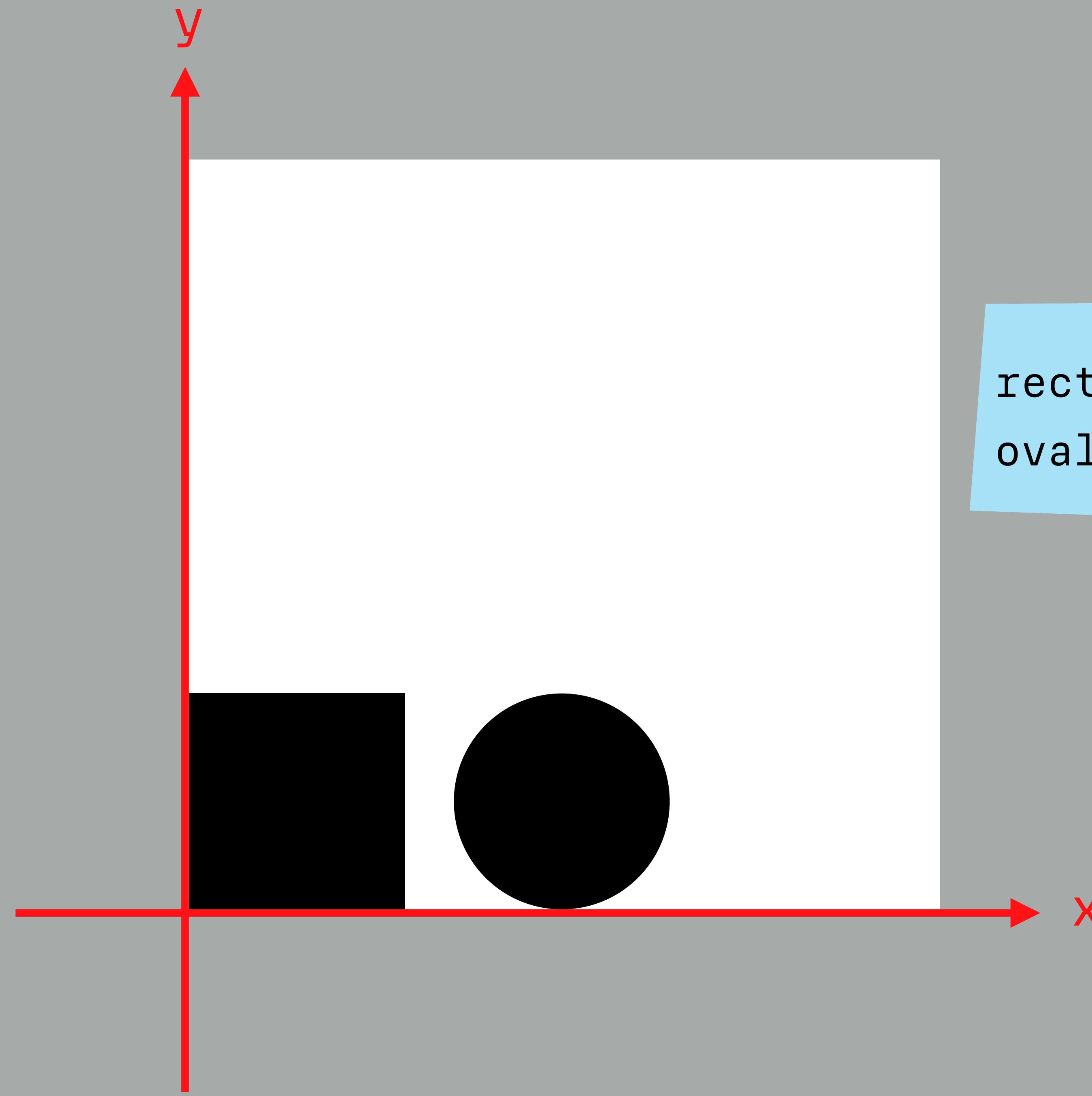
```
rotate(45)
rect(0, 0, 300, 300)
oval(400, 0, 300, 300)
```

wtf?

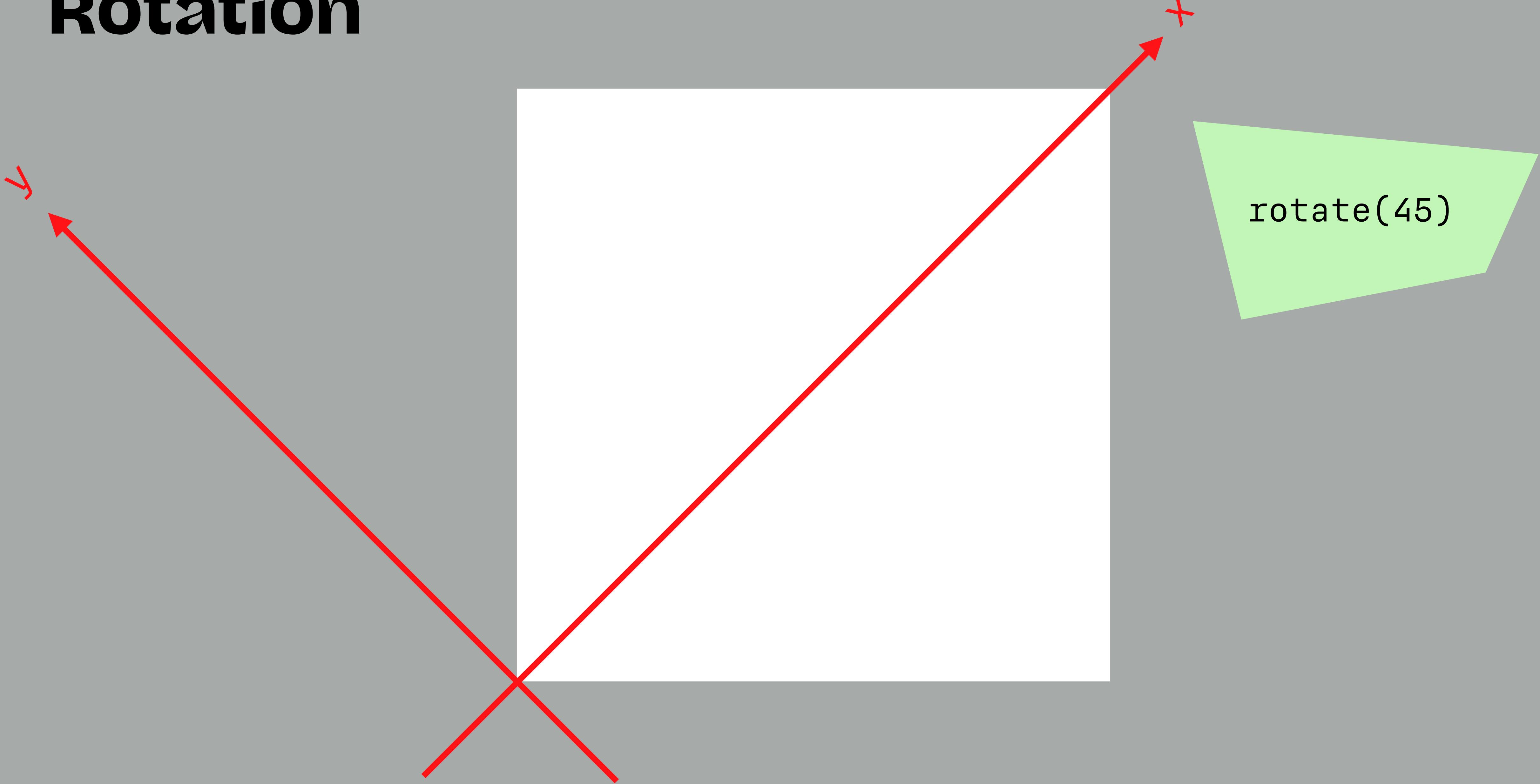
Rotation



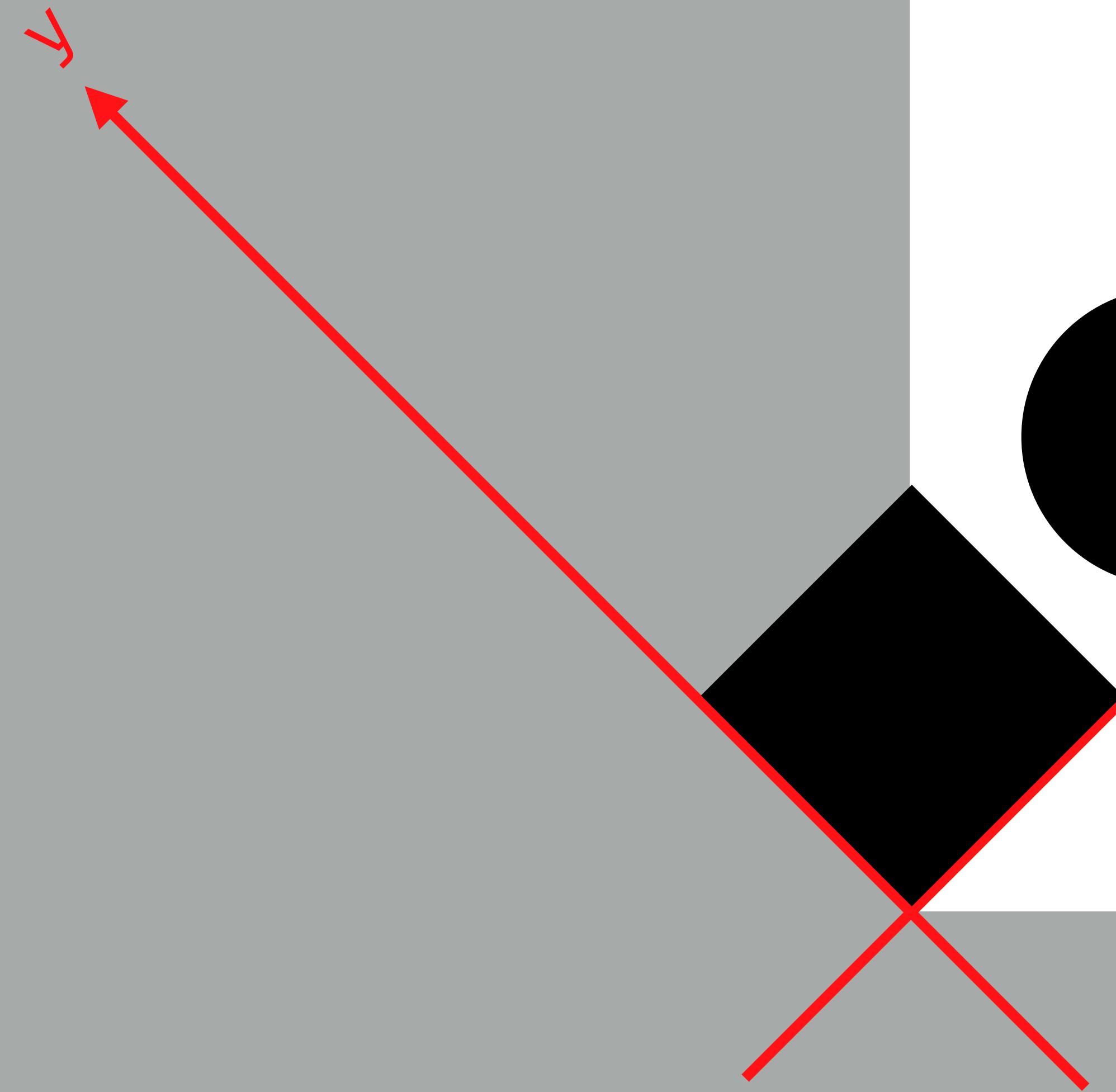
Rotation



Rotation



Rotation



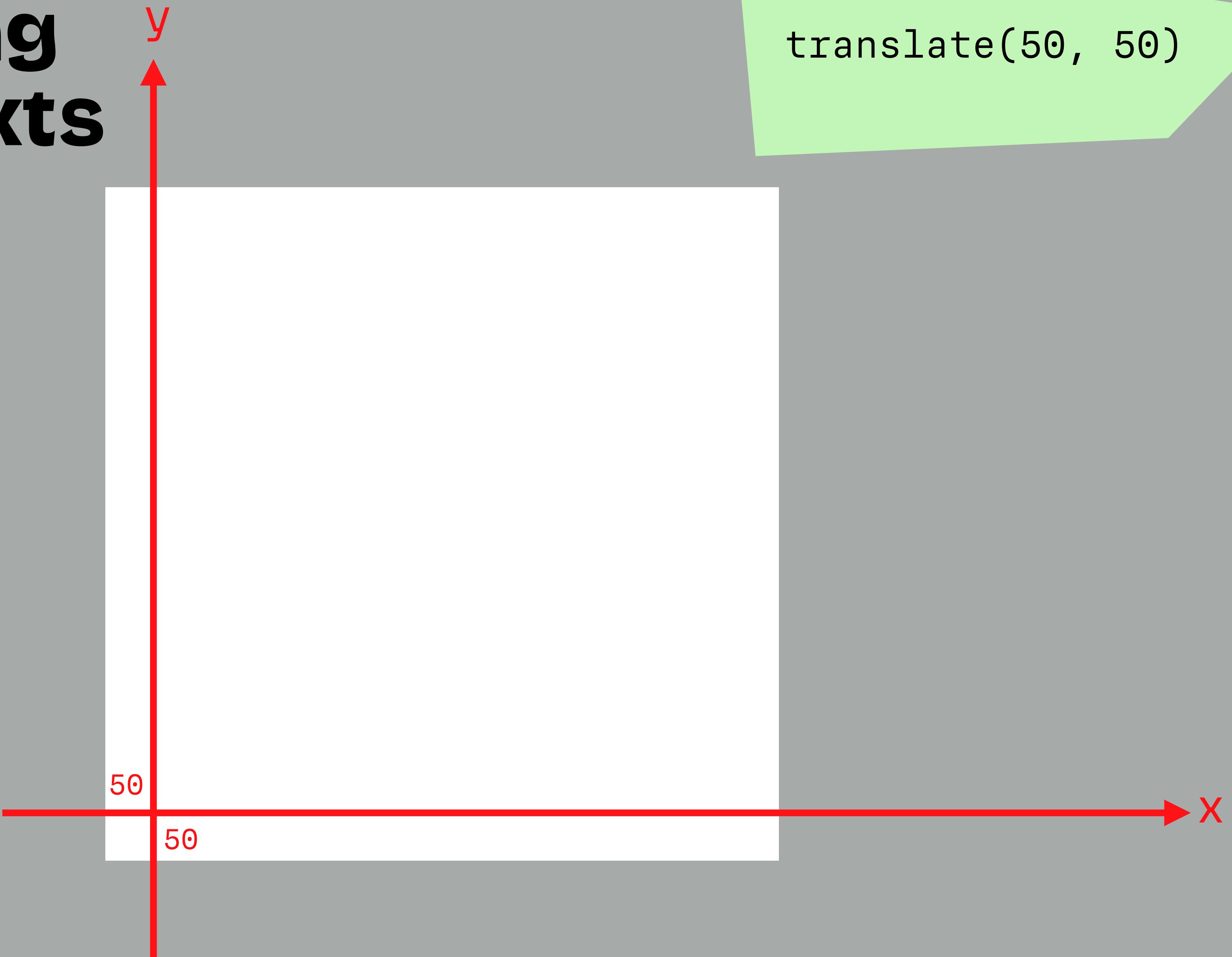
x

+

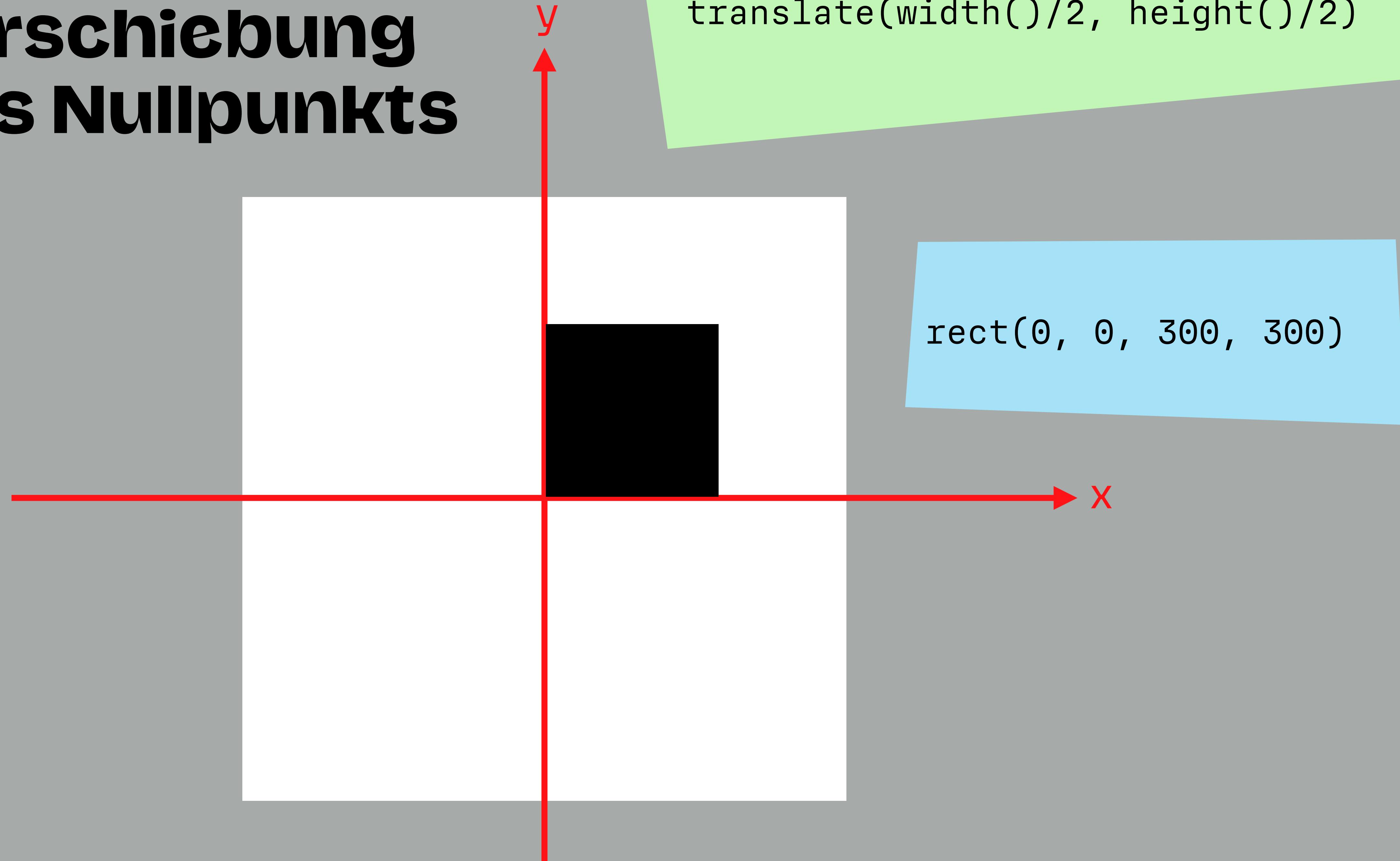
rotate(45)

```
rect(0, 0, 300, 300)  
oval(400, 0, 300, 300)
```

Verschiebung des Nullpunkts

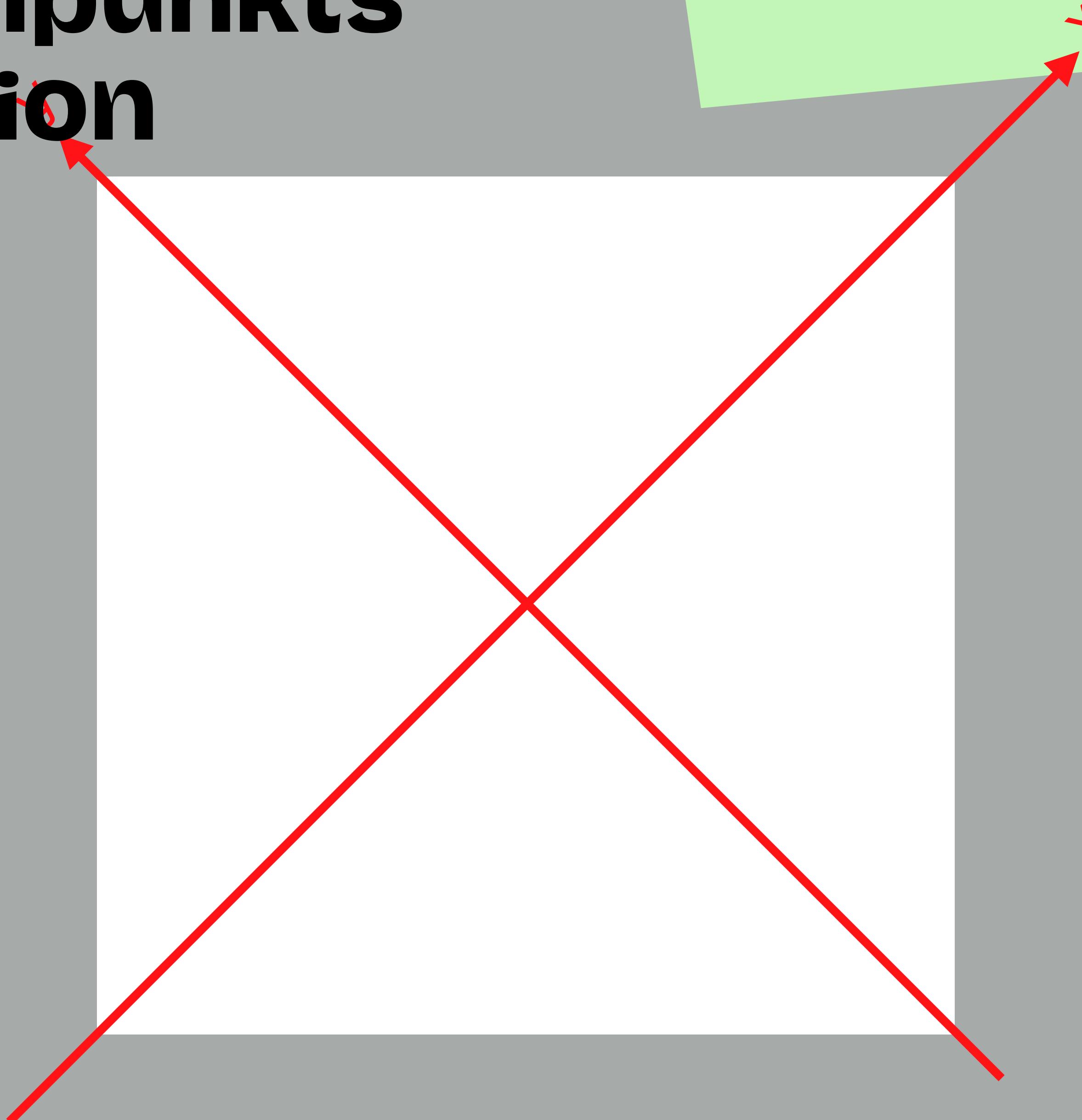


verschiebung des Nullpunkts

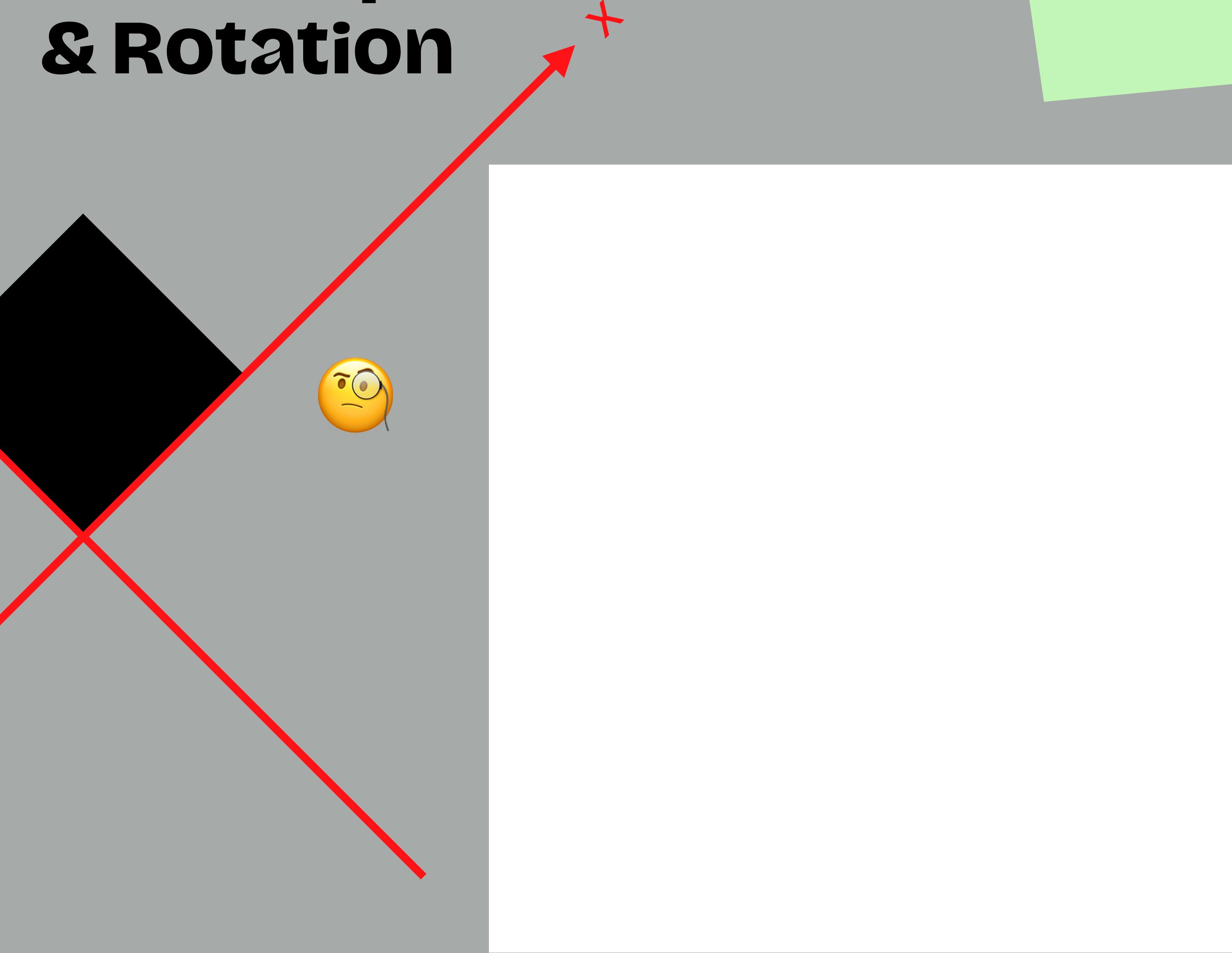


Verschiebung des Nullpunkts & Rotation

translate(width()/2, height()/2)
rotate(45)



Verschiebung des Nullpunkts & Rotation

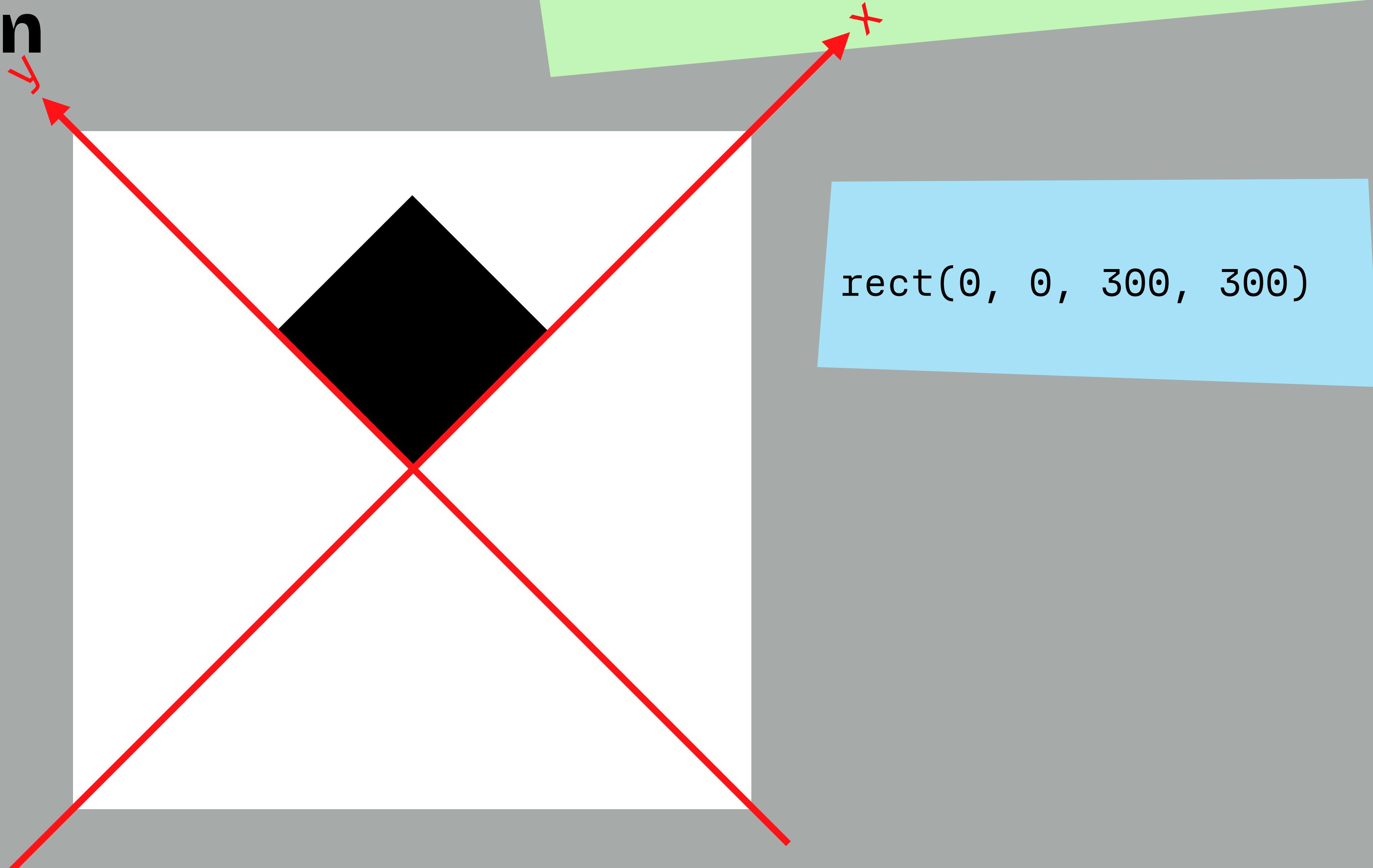


```
rotate(45)  
translate(width() / 2, height() / 2)
```

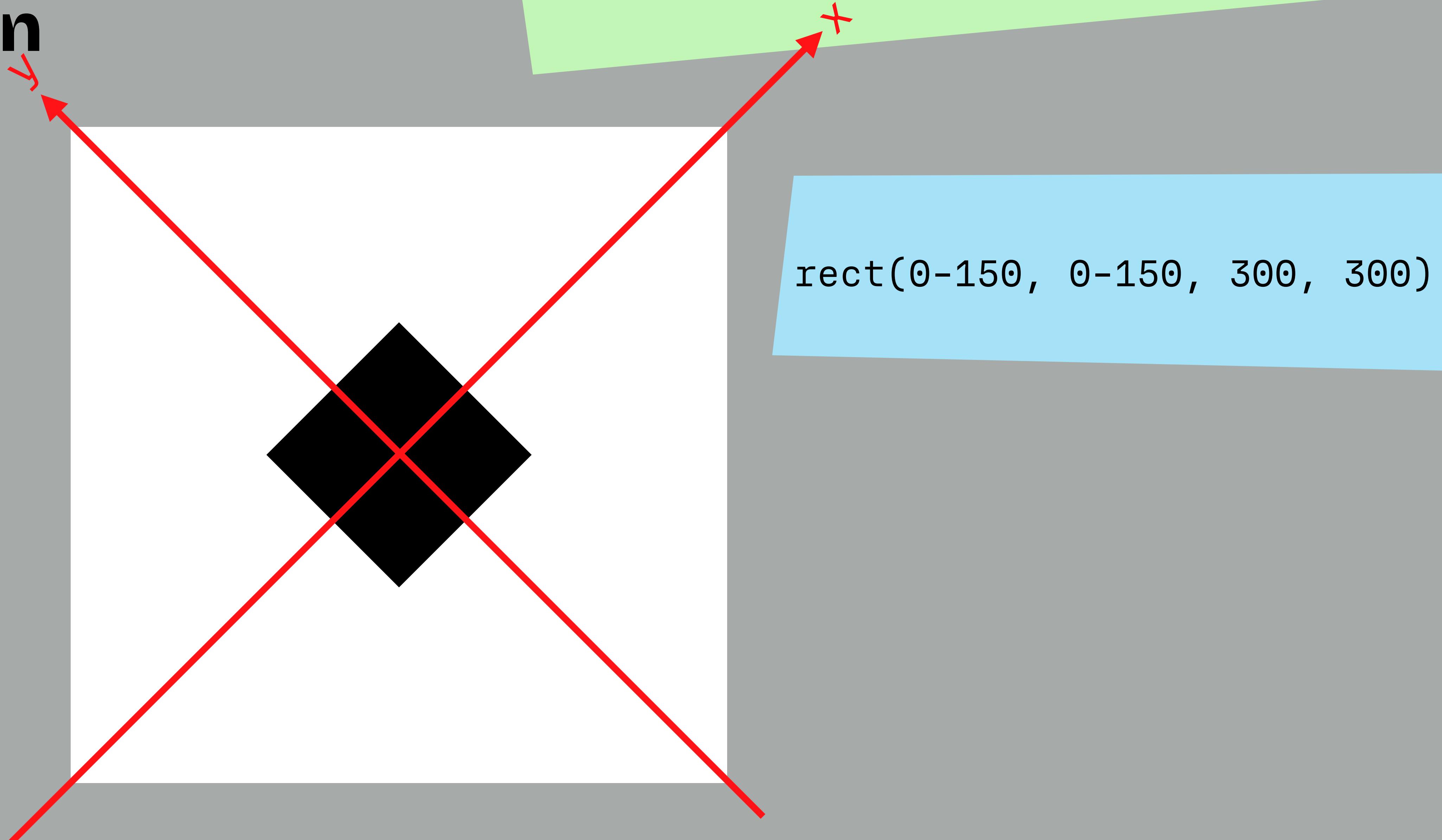
```
rect(0, 0, 300, 300)
```

**Achtung bei
der Reihenfolge!**

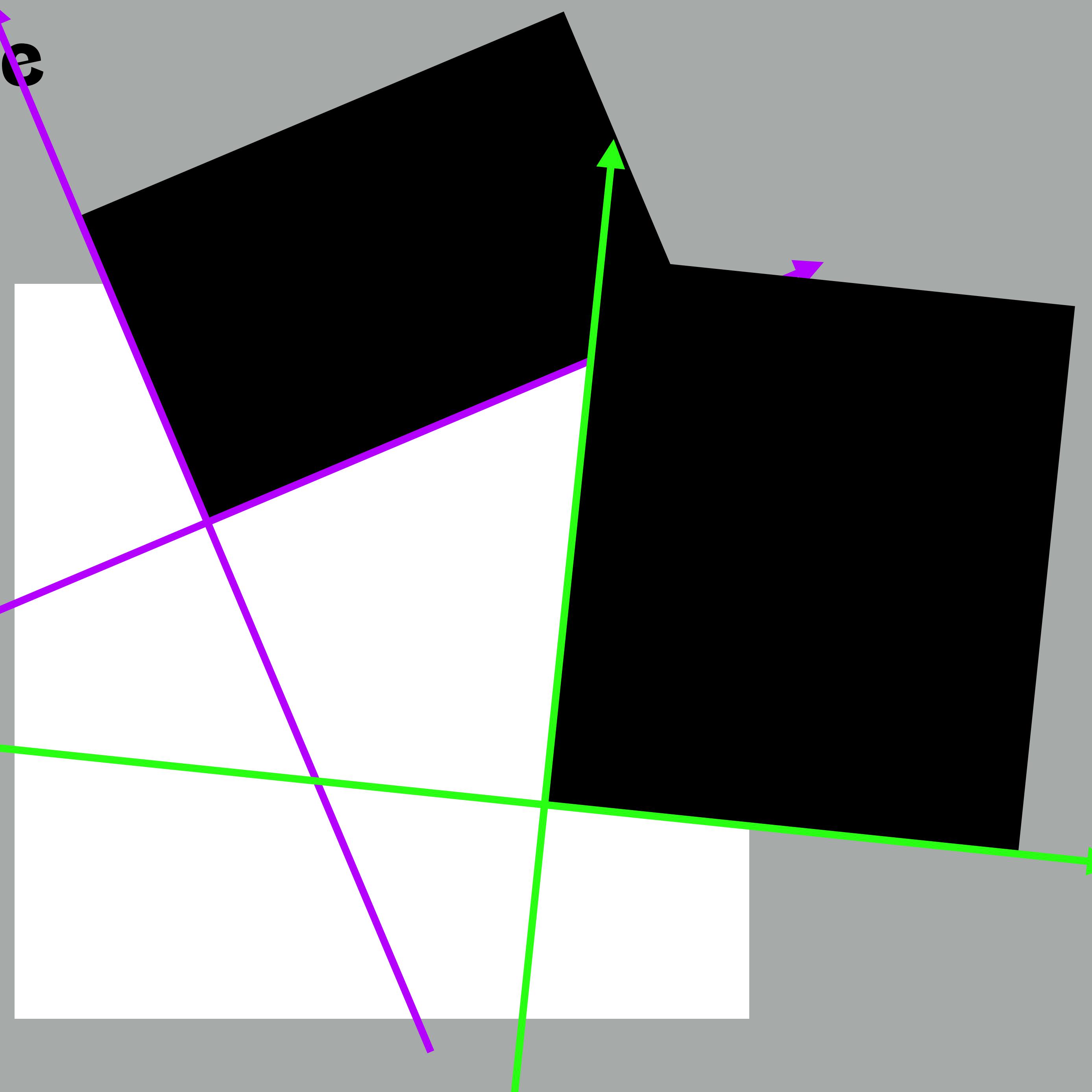
Verschiebung des Nullpunkts & Rotation



Verschiebung des Nullpunkts & Rotation



Komplexere Fälle



Die Rotate-Funktion vereinfacht

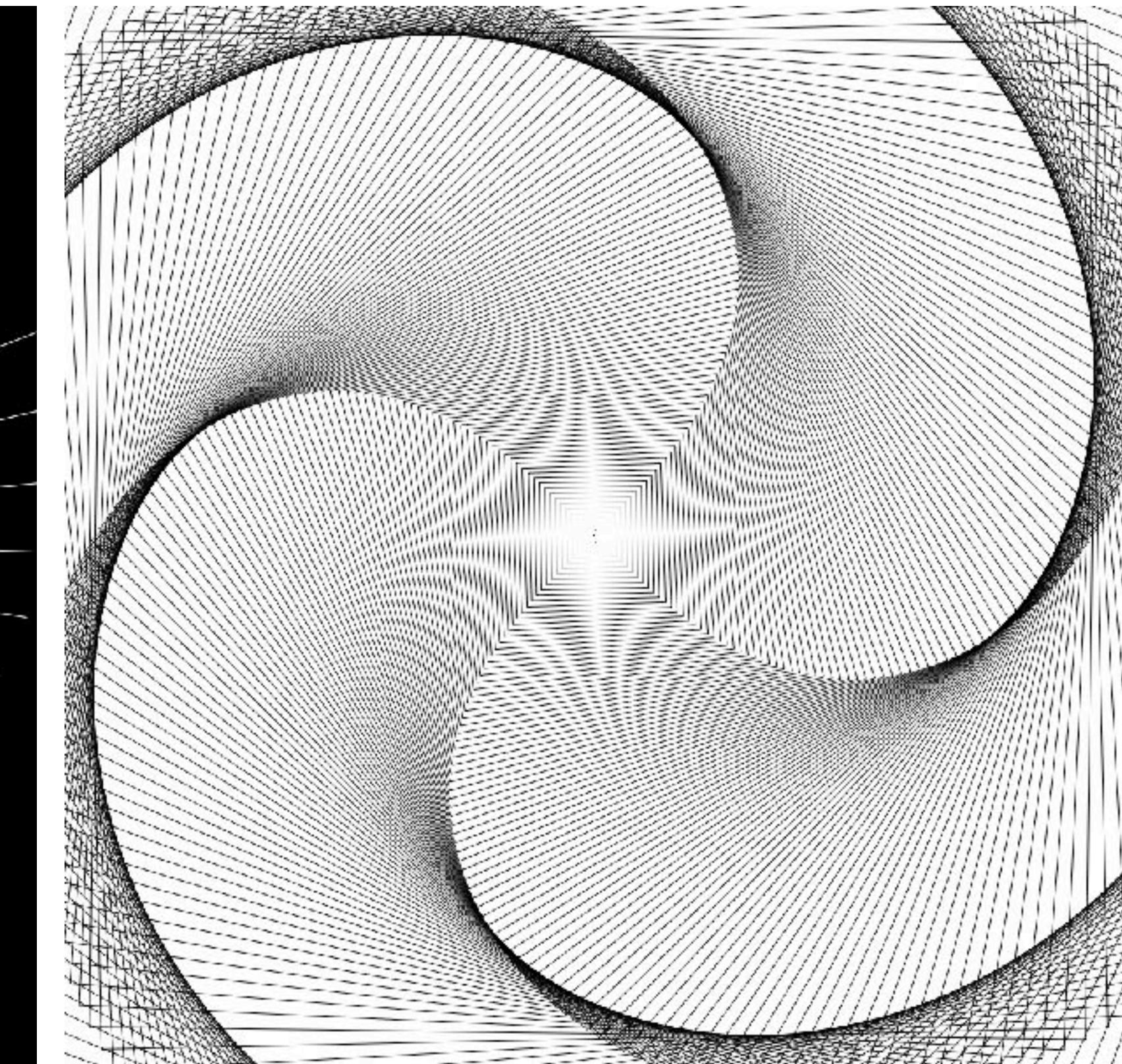
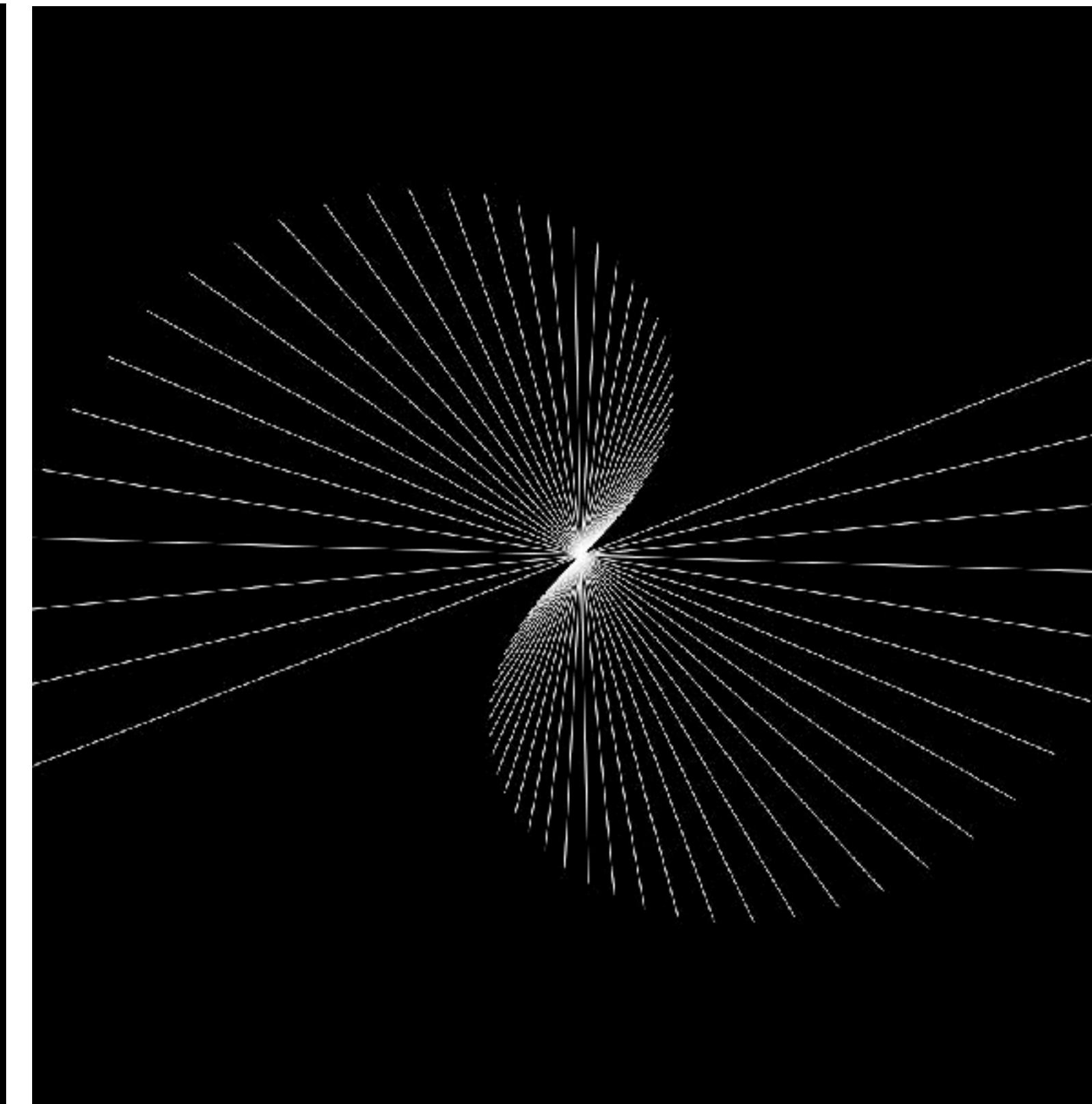
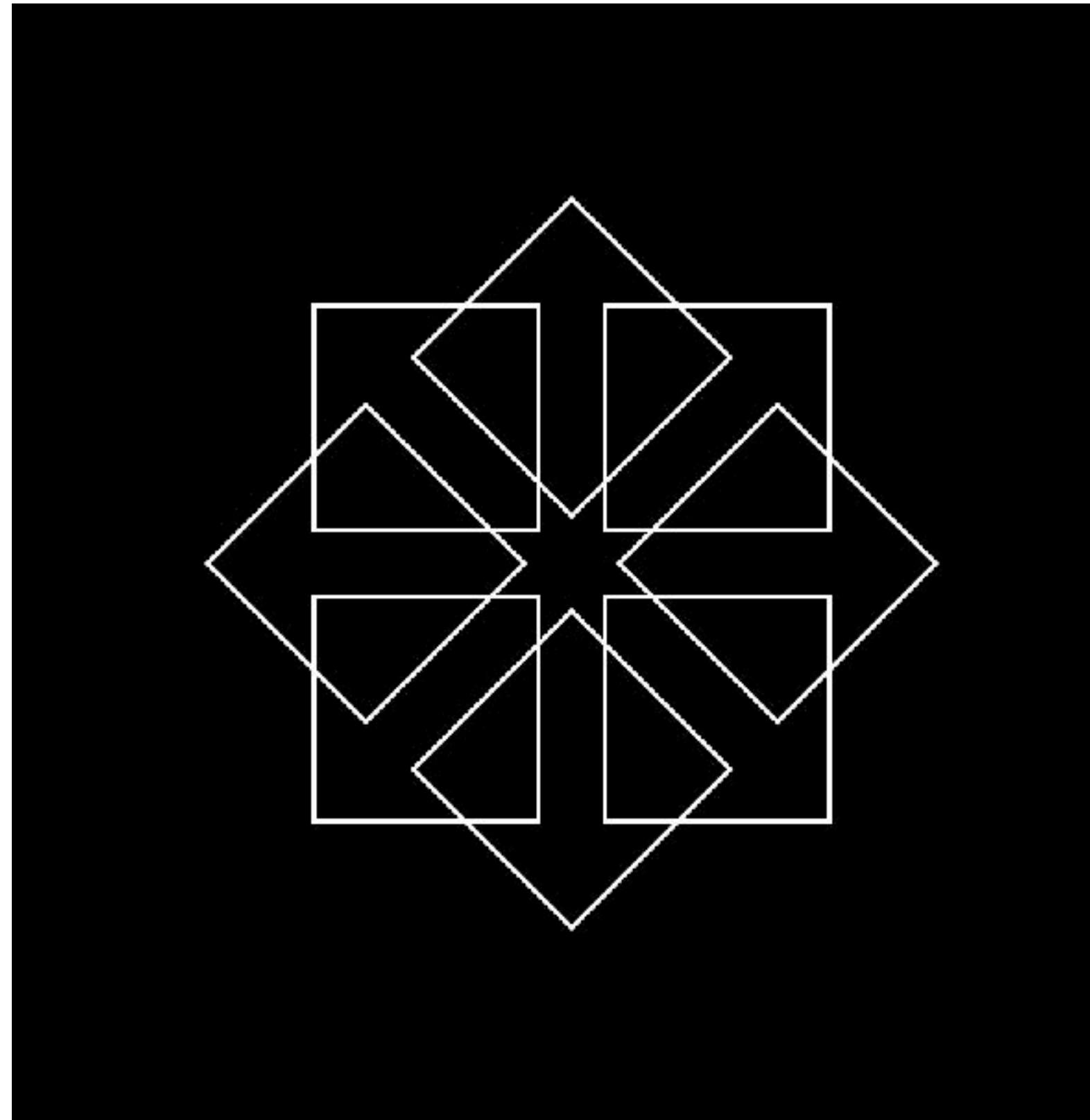
```
translate(100, 100)
```

```
rotate(-45)
```

Kurzschreibweise

```
rotate(-45, (100, 100))
```

Probier mal, etwas zu rotieren



am besten mit einem Loop ...

sawed state

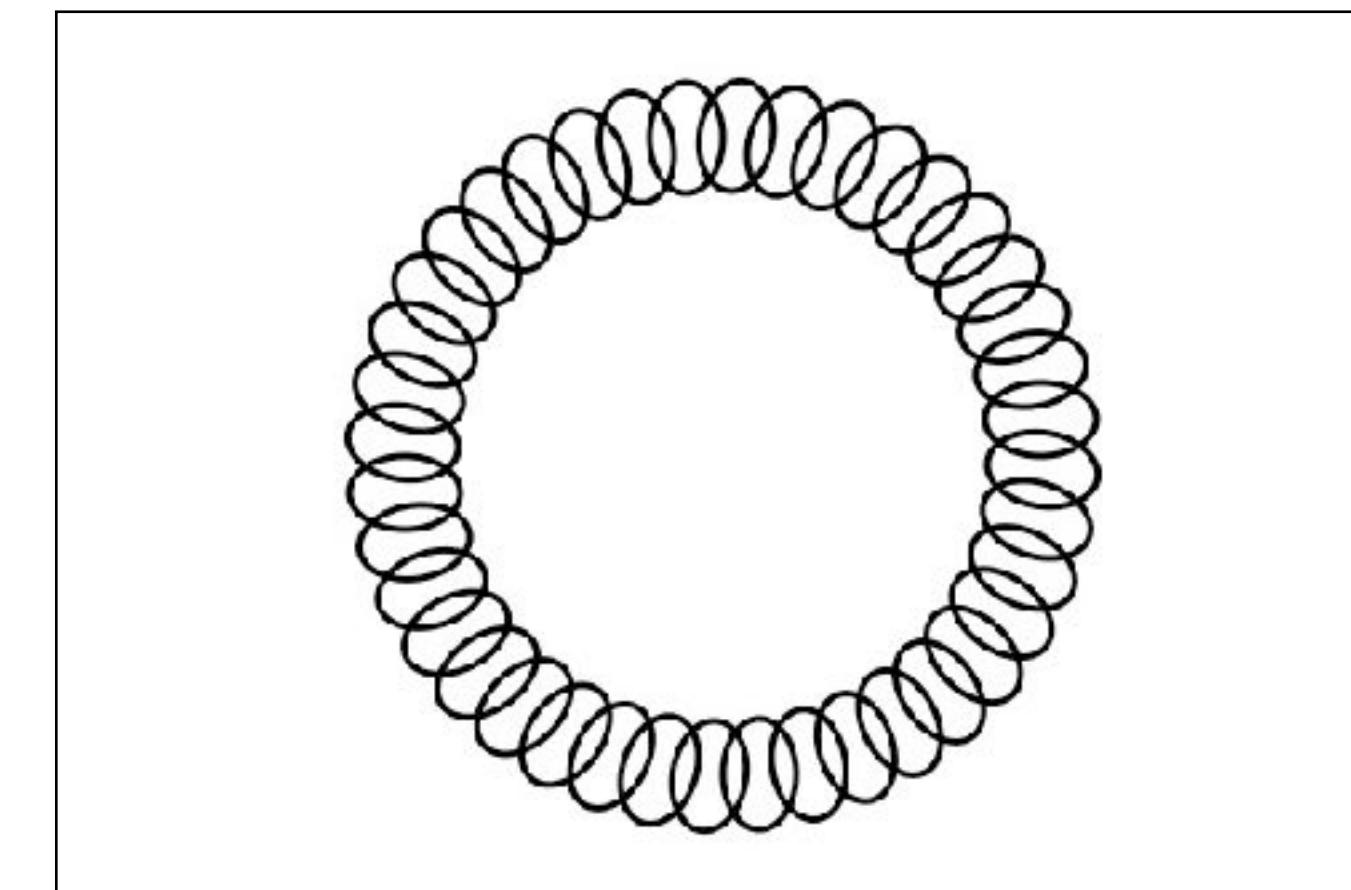
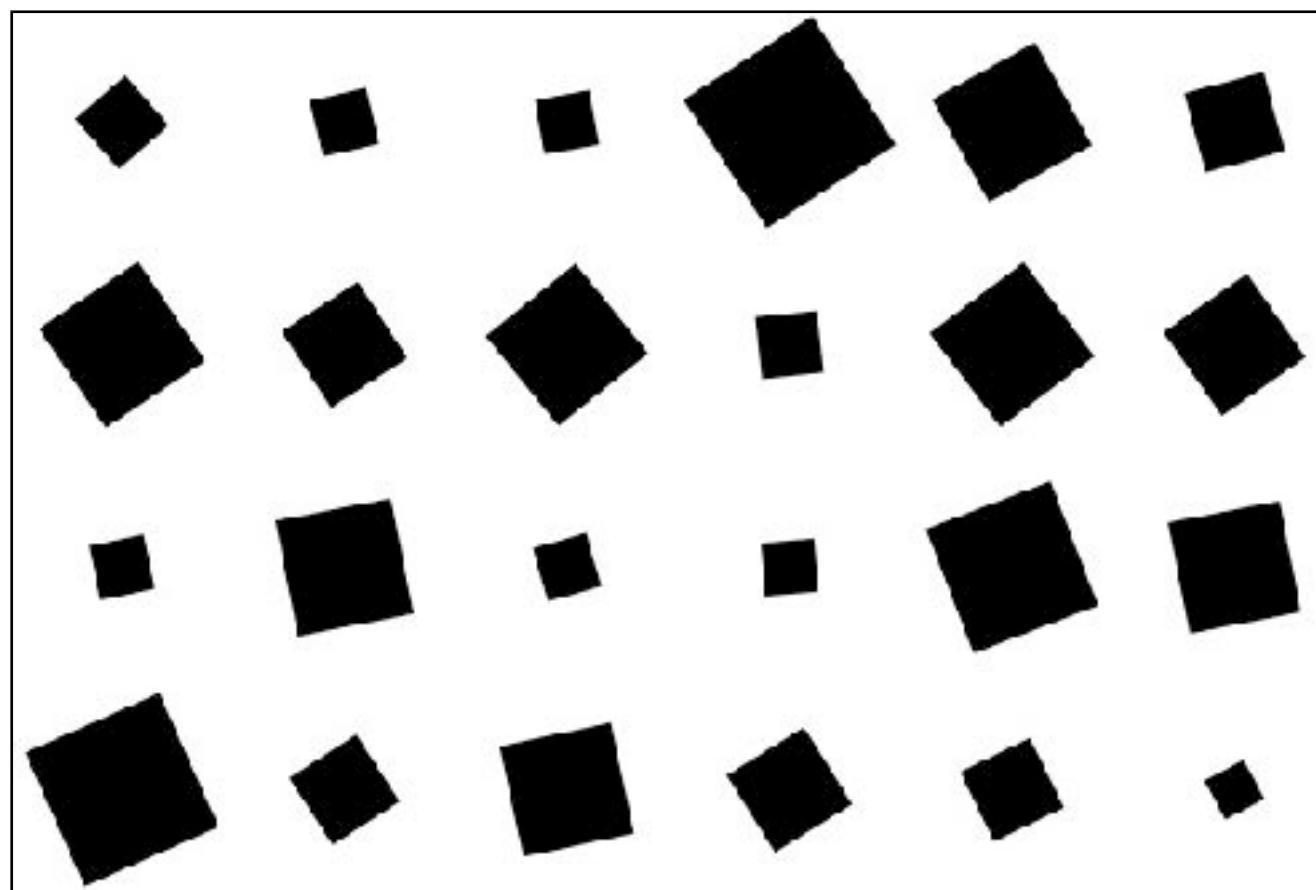
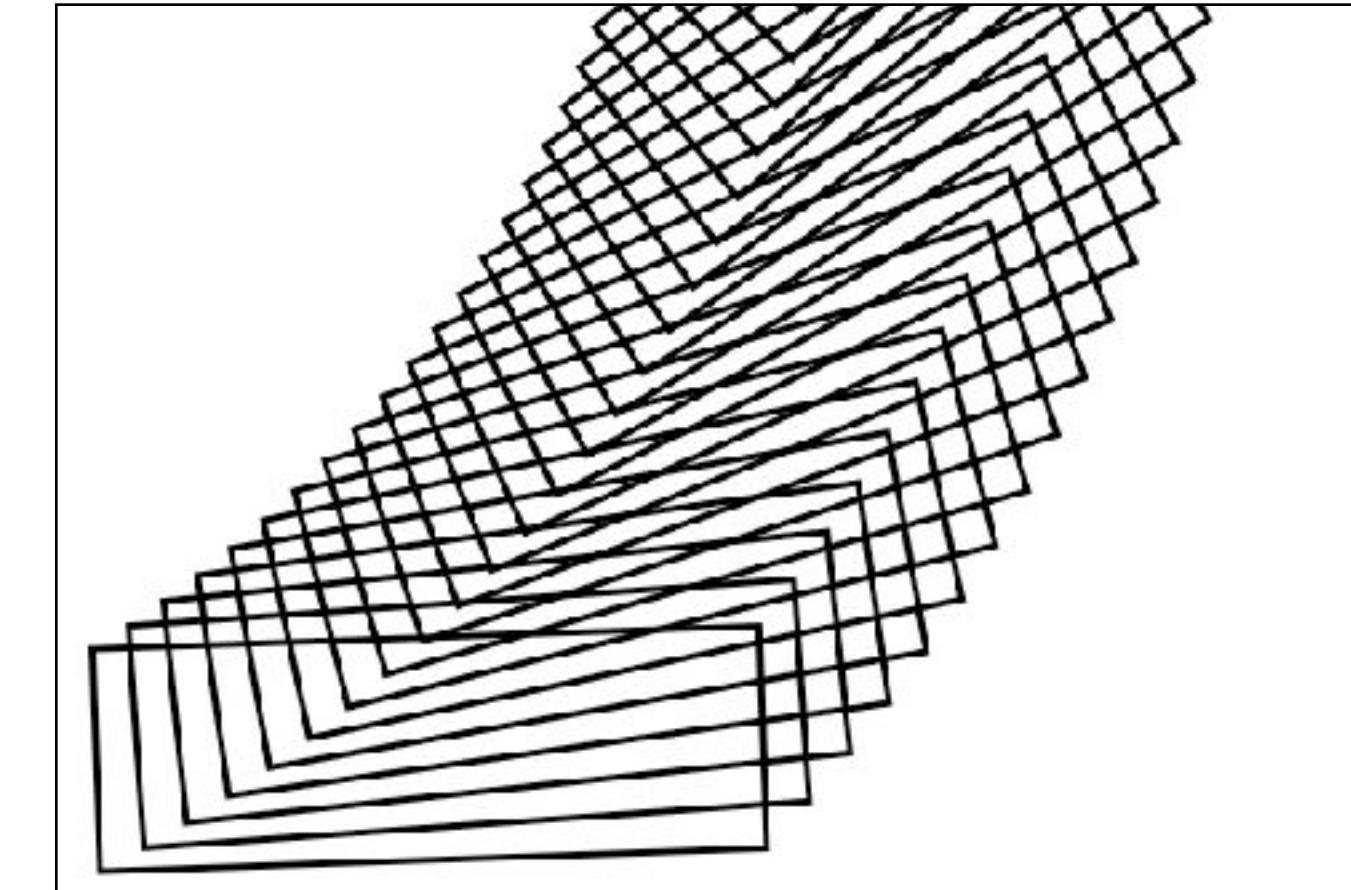
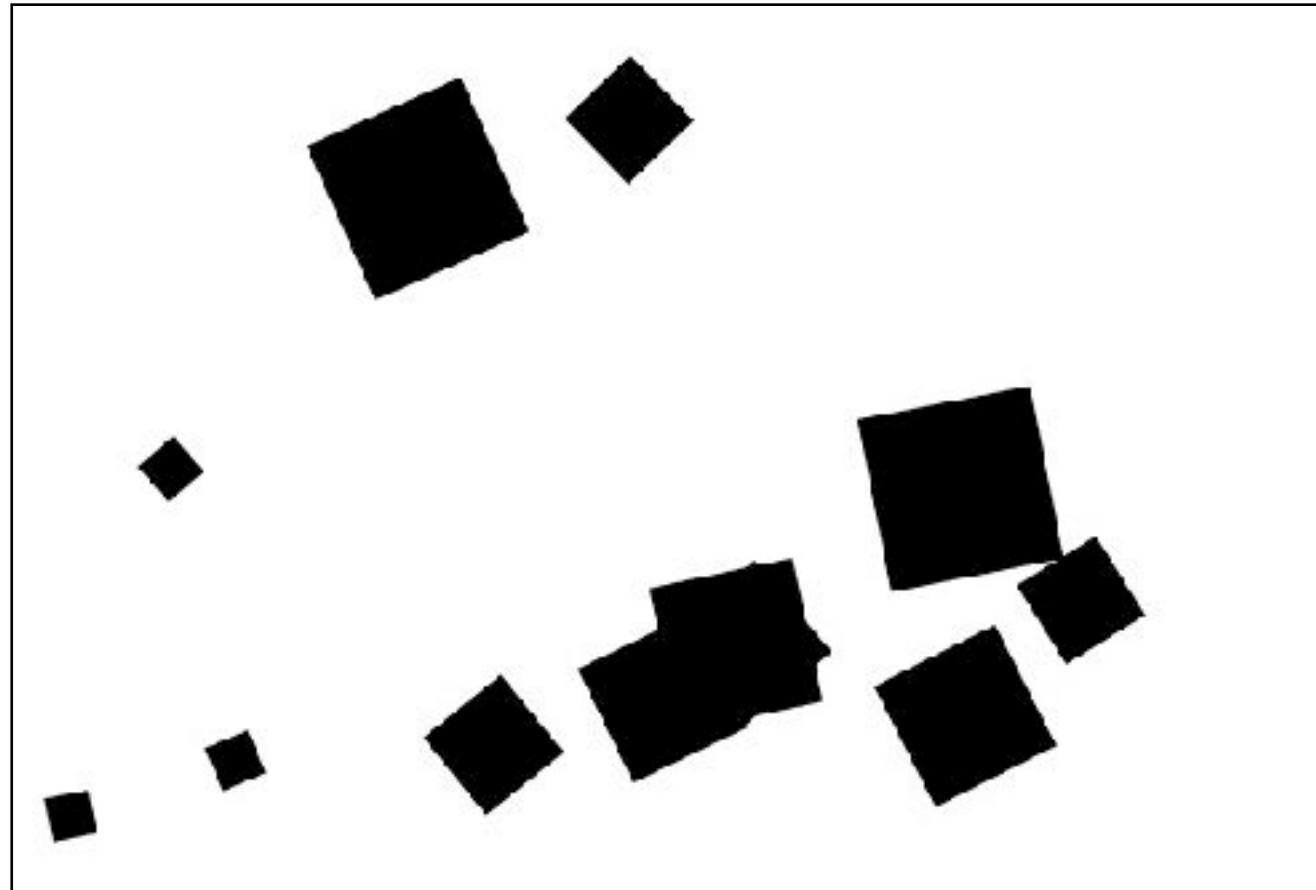
Ausgangszustand wiederherstellen

```
with savedState():  
    translate(300, 300)  
    rotate(30)  
    fill(1, 0, 0)  
    rect(0, 0, 400, 400)  
  
rect(0, 0, 300, 300)
```



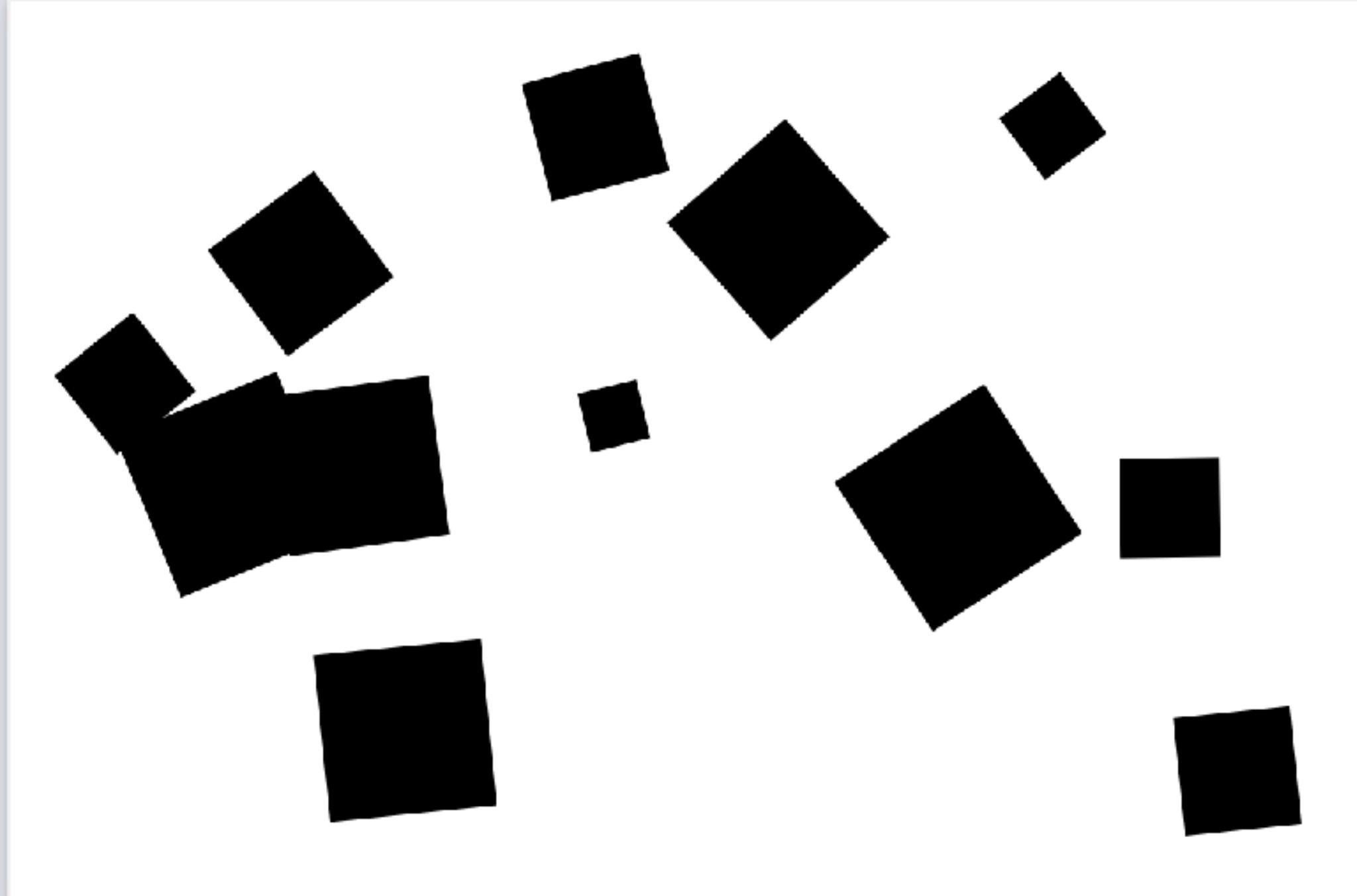
Diese Veränderungen gelten nur im eingerückten Block.
Danach werden Nullpunkt, Rotation und Farbe zurückgesetzt.

Für welche Bilder braucht es savedState?



Untitled 5

Run Comment Uncomment Indent Dedent

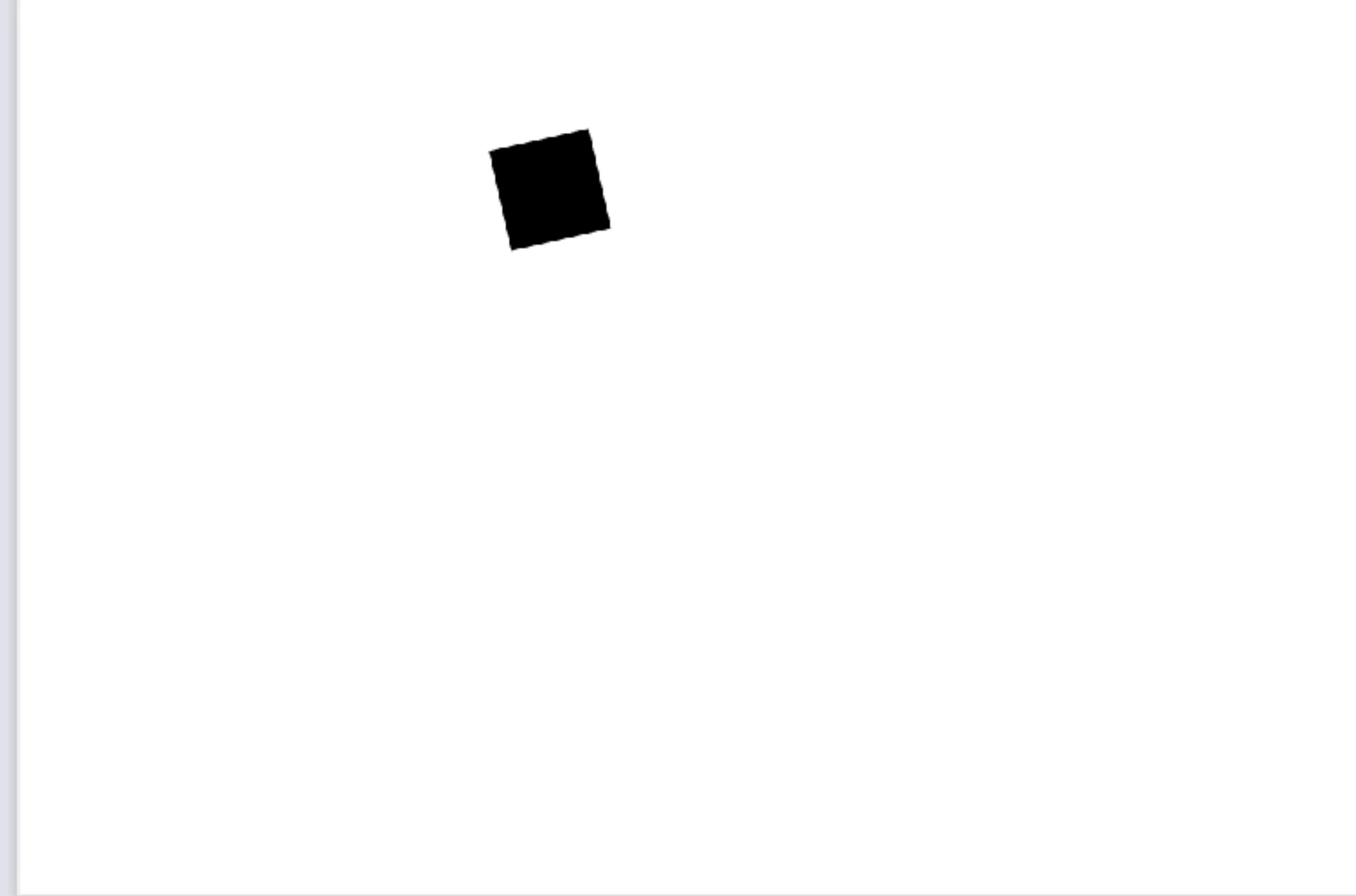


```
1 newPage(600, 400)
2 for a in range(12):
3     with savedState():
4         x = 50 + random() * (width() - 100)
5         y = 50 + random() * (height() - 100)
6         translate(x, y)
7         angle = random() * 45
8         rotate(angle)
9         d = 20 + random() * 60
10        rect(0 - d/2, 0 - d/2, d, d)
```

mit savedState

Untitled 5

Run Comment Uncomment Indent Dedent

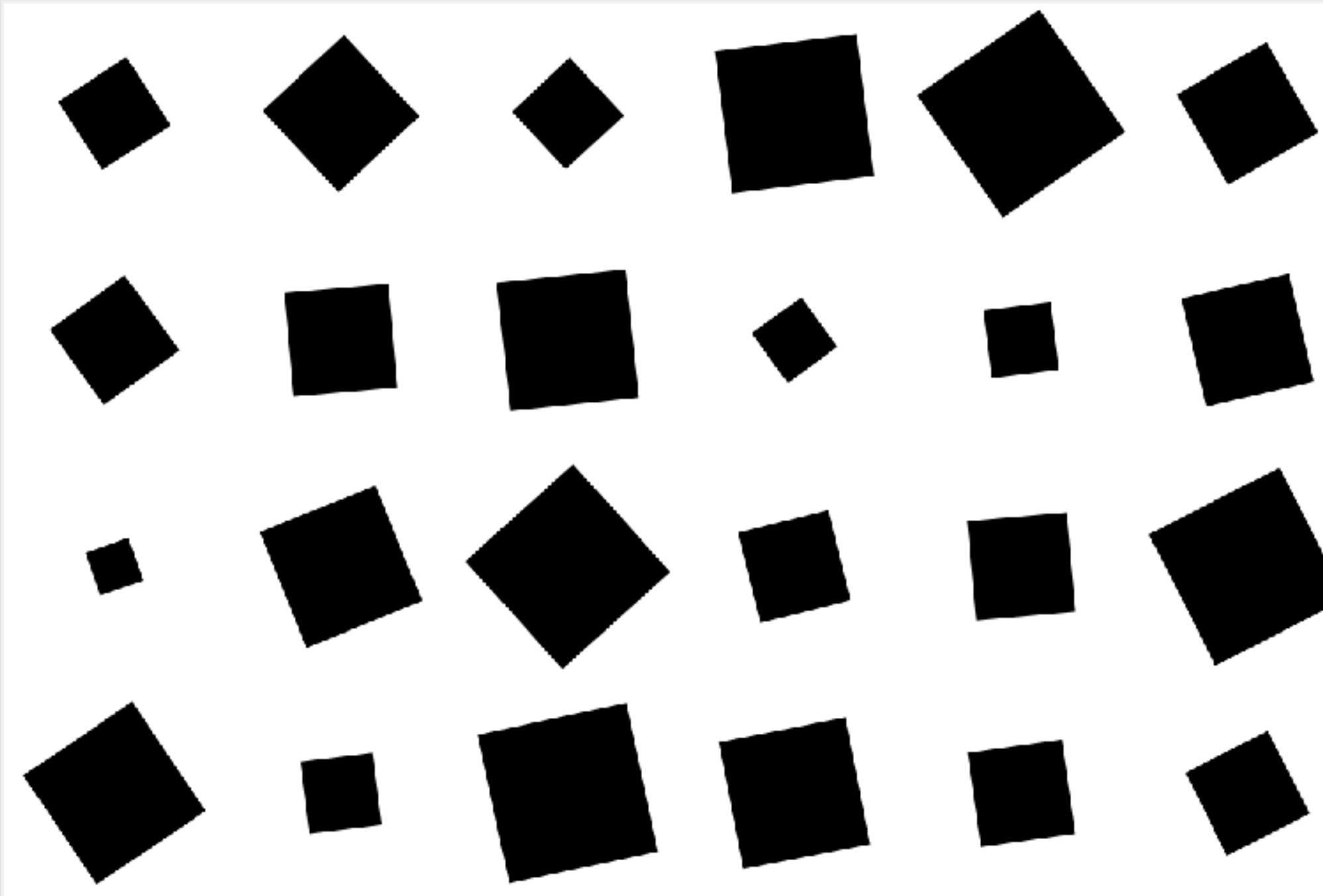


```
1 newPage(600, 400)
2 for a in range(12):
3     x = 50 + random() * (width() - 100)
4     y = 50 + random() * (height() - 100)
5     translate(x, y)
6     angle = random() * 45
7     rotate(angle)
8     d = 20 + random() * 60
9     rect(0 - d/2, 0 - d/2, d, d)
```

ohne savedState

Untitled 5

Run Comment Uncomment Indent Dedent



```
1 newPage(600, 400)
2
3 for row in range(4):
4     for col in range(6):
5         with savedState():
6             translate(col*100 + 50, row*100 + 50)
7             d = randint(20, 70)
8             rotate(randint(5, 45))
9             rect(-d/2, -d/2, d, d)
```

mit savedState

Untitled 5

Run Comment Uncomment Indent Dedent

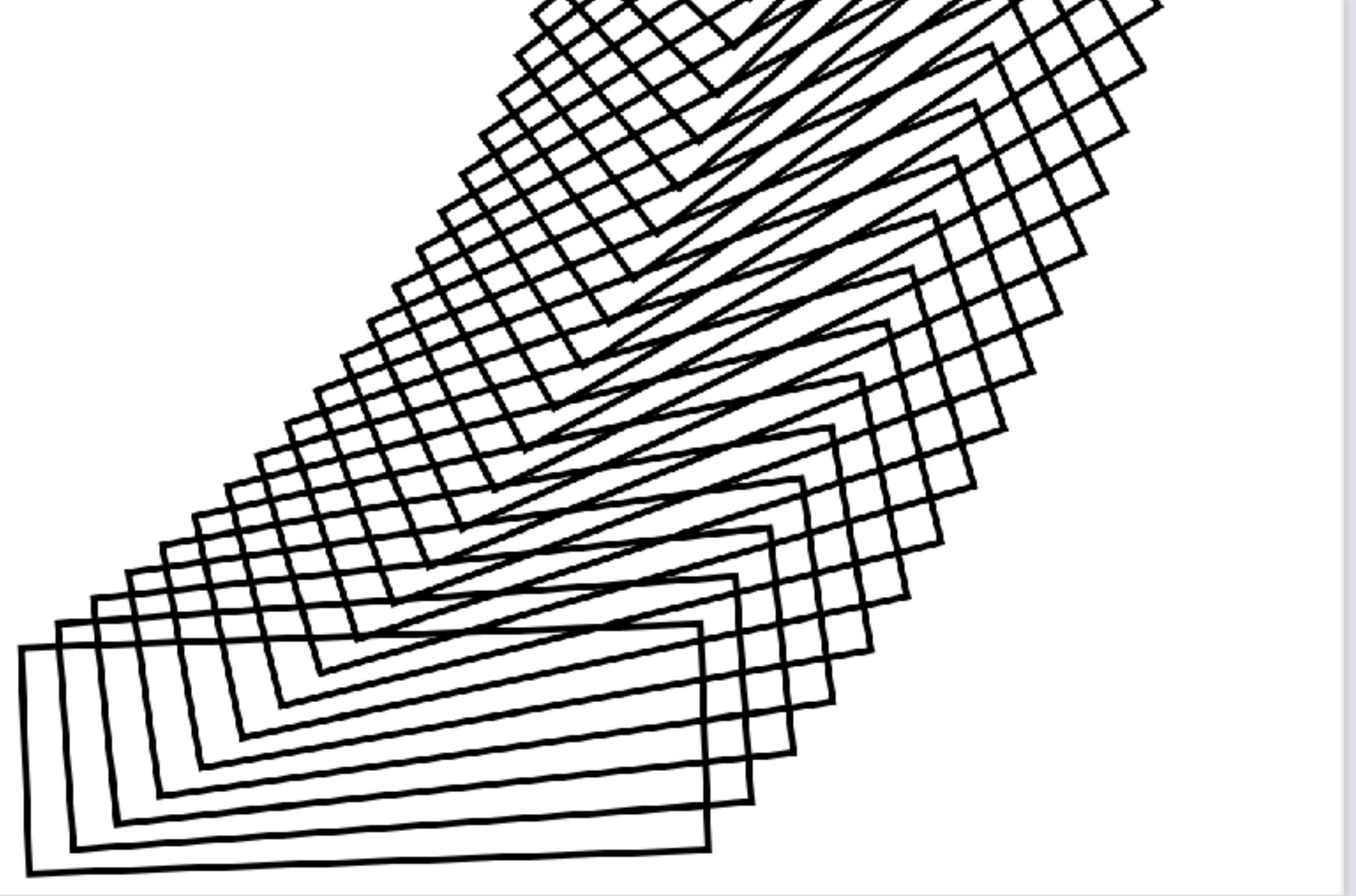
The code generates a 4x6 grid of black rectangles. Each rectangle has a width and height of d , and is rotated by a random angle between 5 and 45 degrees. The rectangles are positioned at coordinates $(col*100 + 50, row*100 + 50)$.

```
1 newPage(600, 400)
2
3 for row in range(4):
4     for col in range(6):
5         translate(col*100 + 50, row*100 + 50)
6         d = randint(20, 70)
7         translate(50, 50)
8         rotate(randint(5, 45))
9         rect(-d/2, -d/2, d, d)
```

ohne savedState

Untitled 5

Run Comment Uncomment Indent Dedent



```
1 newPage(600, 400)
2 fill(None)
3 stroke(0)
4 strokeWidth(3)
5 for a in range(30):
6     translate(20, 10)
7     rotate(2)
8     rect(0, 0, 300, 100)
```

ohne savedState

Untitled 5

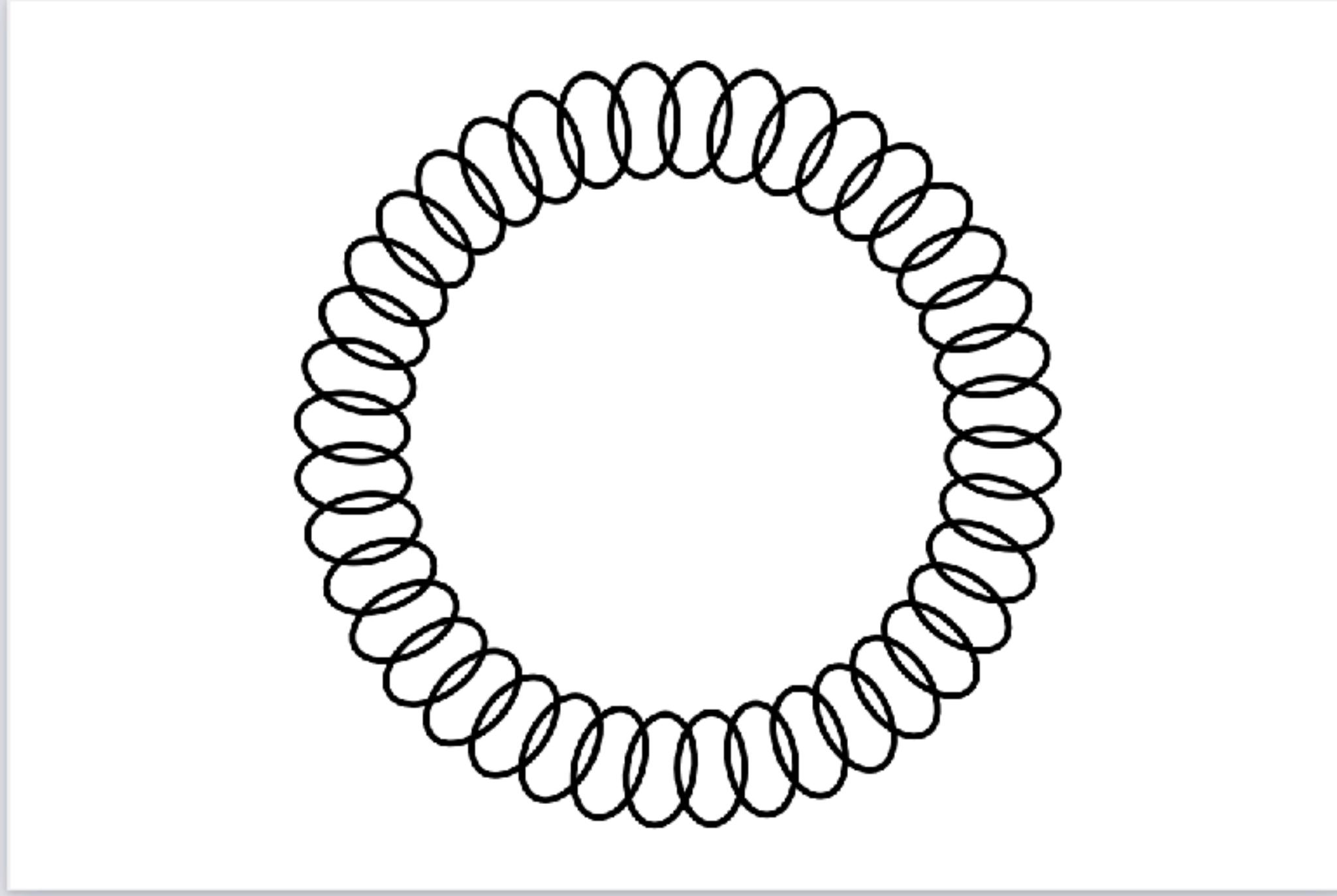
Run Comment Uncomment Indent Dedent

```
1 newPage(600, 400)
2 fill(None)
3 stroke(0)
4 strokeWidth(3)
5 for a in range(30):
6     with savedState():
7         translate(20, 10)
8         rotate(2)
9         rect(0, 0, 300, 100)
```

mit savedState

Untitled 5

Run Comment Uncomment Indent Dedent



```
1 newPage(600, 400)
2 fill(None)
3 stroke(0)
4 strokeWidth(3)
5 translate(width()/2, height()/2)
6 for a in range(40):
7     rotate(9)
8     oval(120, 0, 50, 30)
```

ohne savedState

Untitled 5

Run Comment Uncomment Indent Dedent

The code editor displays a Python script with the following code:

```
1 newPage(600, 400)
2 fill(None)
3 stroke(0)
4 strokeWidth(3)
5 translate(width()/2, height()/2)
6 for a in range(40):
7     with savedState():
8         rotate(9)
9         oval(120, 0, 50, 30)
```

The output window shows a single black oval centered on the canvas.

mit savedState