

海龟岛的试炼（一）





1. 探索新知

1.1

地图展开

位置默认在
屏幕正中心



turtle绘图窗体（窗口）生成：

- `turtle.setup(width, height, startx, starty)`
 - `width`—窗口宽度
 - `height`—窗口高度
 - `startx`—窗口左侧与屏幕左侧的像素距离
 - `starty`—窗口顶部与屏幕顶部的像素距离

怎么生成指定大小、
位置的窗口？



1.1

地图展开

生成画布:

- `turtle.screensize(canvwidth, canvheight, bg)`
 - `canvwidth`—正整数, 表示画布的宽度
 - `canvheight`—正整数, 表示画布的高度
 - `bg`—背景颜色



注意:

绘图窗口不改变



练一练

要求:

生成绘图窗口:

宽为800, 高为600, 窗口位于屏幕左上角;

召唤画布:

宽为1000, 高为800, 颜色为“blue”



1.2

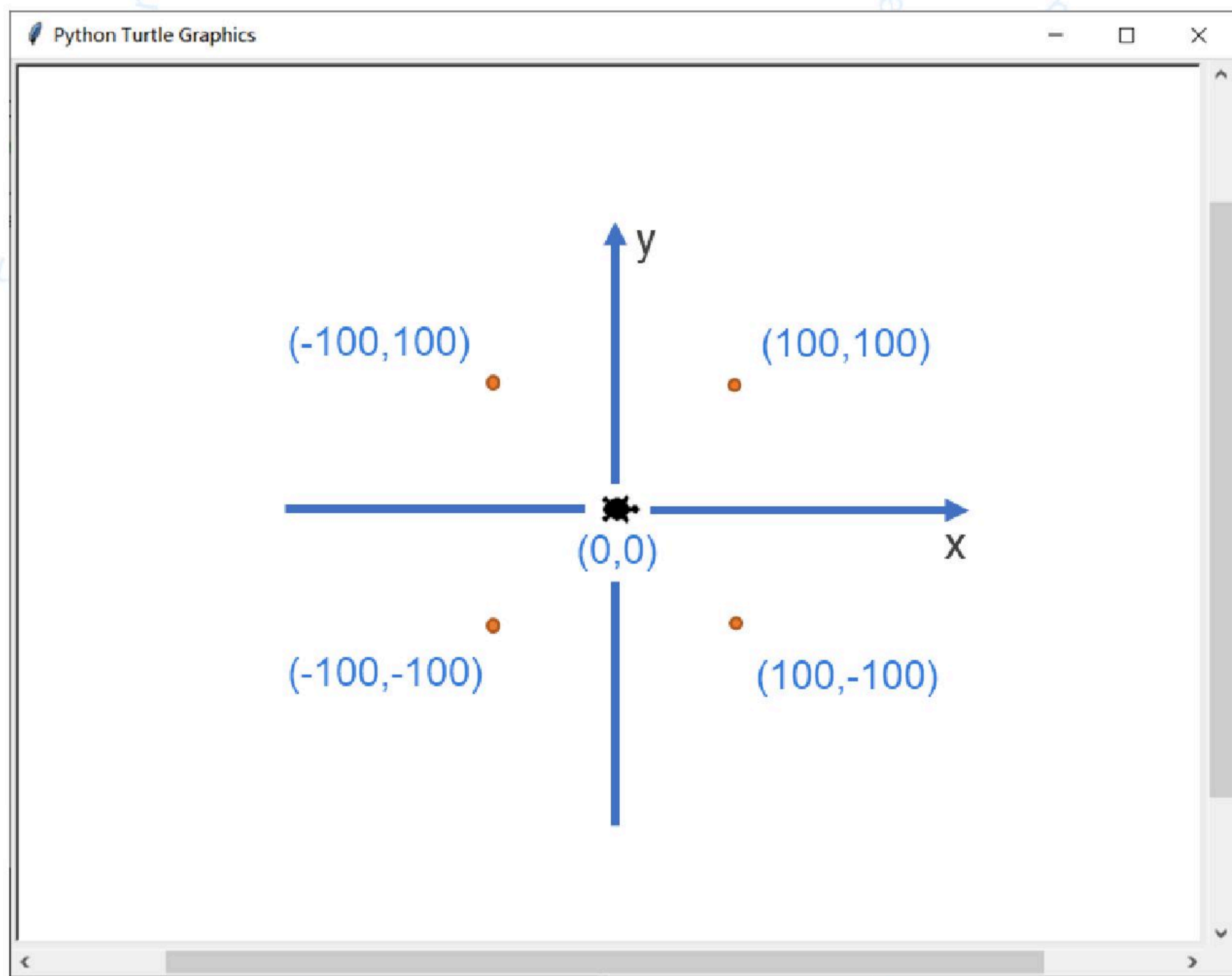
神秘传送阵



海龟岛上的传送阵，一旦被触发，就可以实现定点传输，那么是如何进行定位的呢？

• turtle的空间坐标体系—绝对坐标和海龟坐标

①绝对坐标 中心位置 (0, 0)



setposition
的缩写

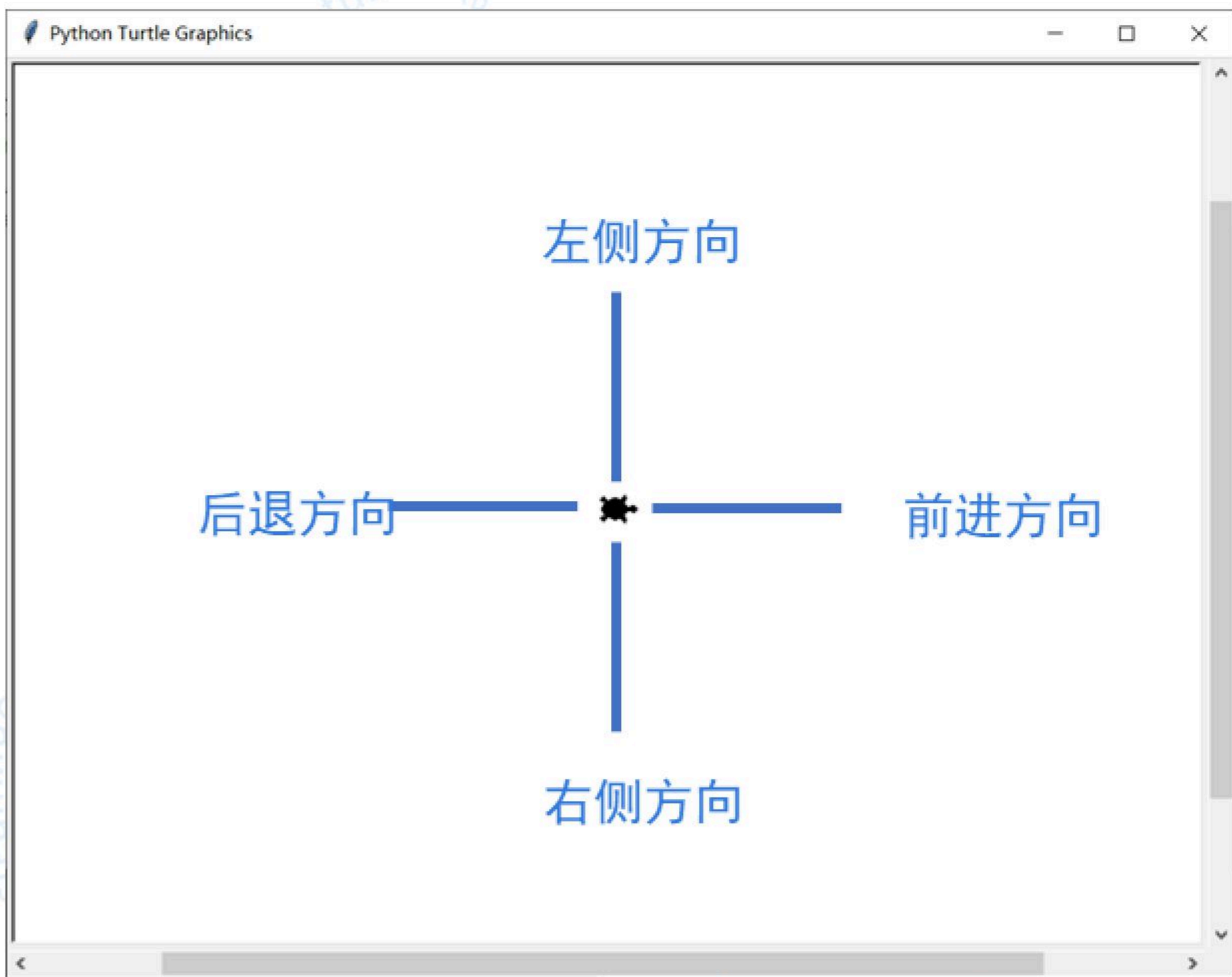
朝向不变

- turtle.goto(x,y) / turtle.setposition(x,y) / turtle.setpos(x,y)
 去往 设定位置 移到坐标(x,y)处
- turtle.setx(x) 设置海龟的横坐标为x，纵坐标保持不变
- turtle.sety(y) 设置海龟的纵坐标为y，横坐标保持不变
- turtle.home() 海龟移至初始位置(0,0)，并设置朝向为初始方向 (右)
 回家

1.2

神秘传送阵

②海龟坐标



思考：
turtle.fd(-10)/
turtle.bk(-10)
分别什么效果？

turtle.forward(d)

别名: turtle.fd(d)

海龟前进d个像素

turtle.backward(d)

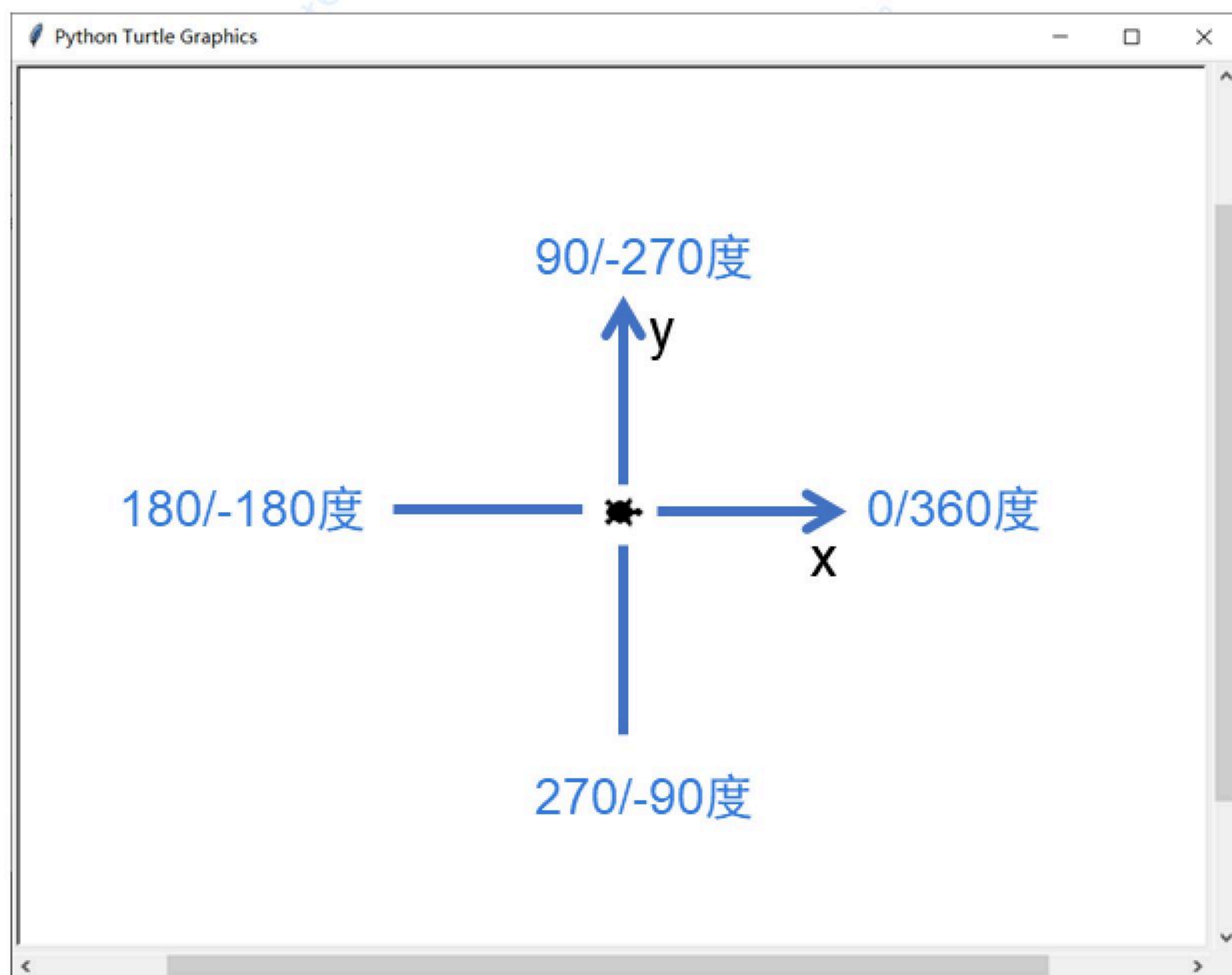
别名: turtle.bk(d)/

turtle.back(d)

海龟后退d个像素

- turtle的角度体系—绝对角度和海龟角度

①绝对角度



turtle.setheading(angle)

别名: turtle.seth(angle)

设置海龟的朝向为angle

eg:

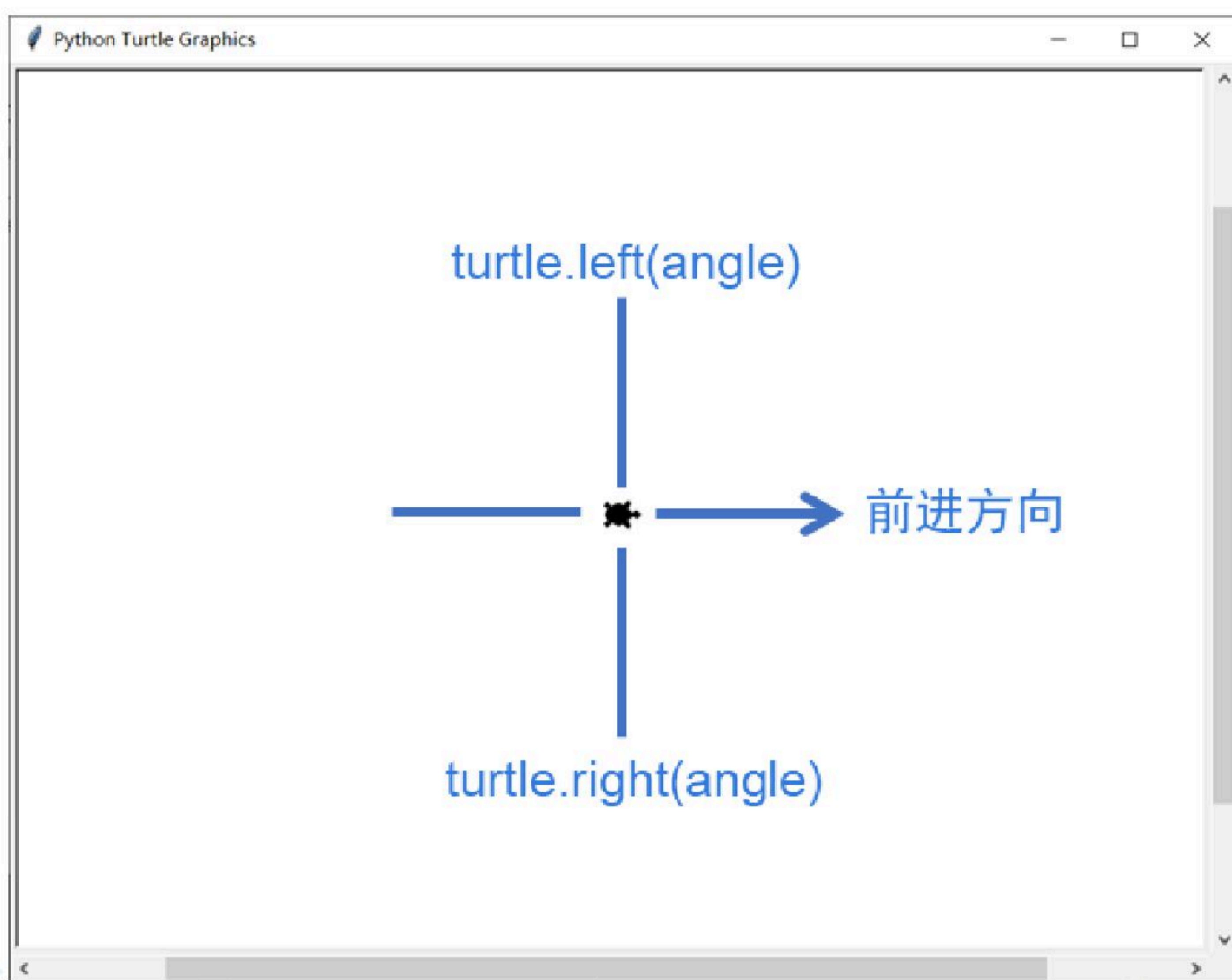
turtle.seth(90)



1.2

神秘传送阵

②海龟角度



turtle.left(angle)

别名: turtle.lt(angle)

在原有前进方向上, 向左转angle度

turtle.right(angle)

别名: turtle.rt(angle)

在原有前进方向上, 向右转angle度

1.3

圈圈圆圆圈圈

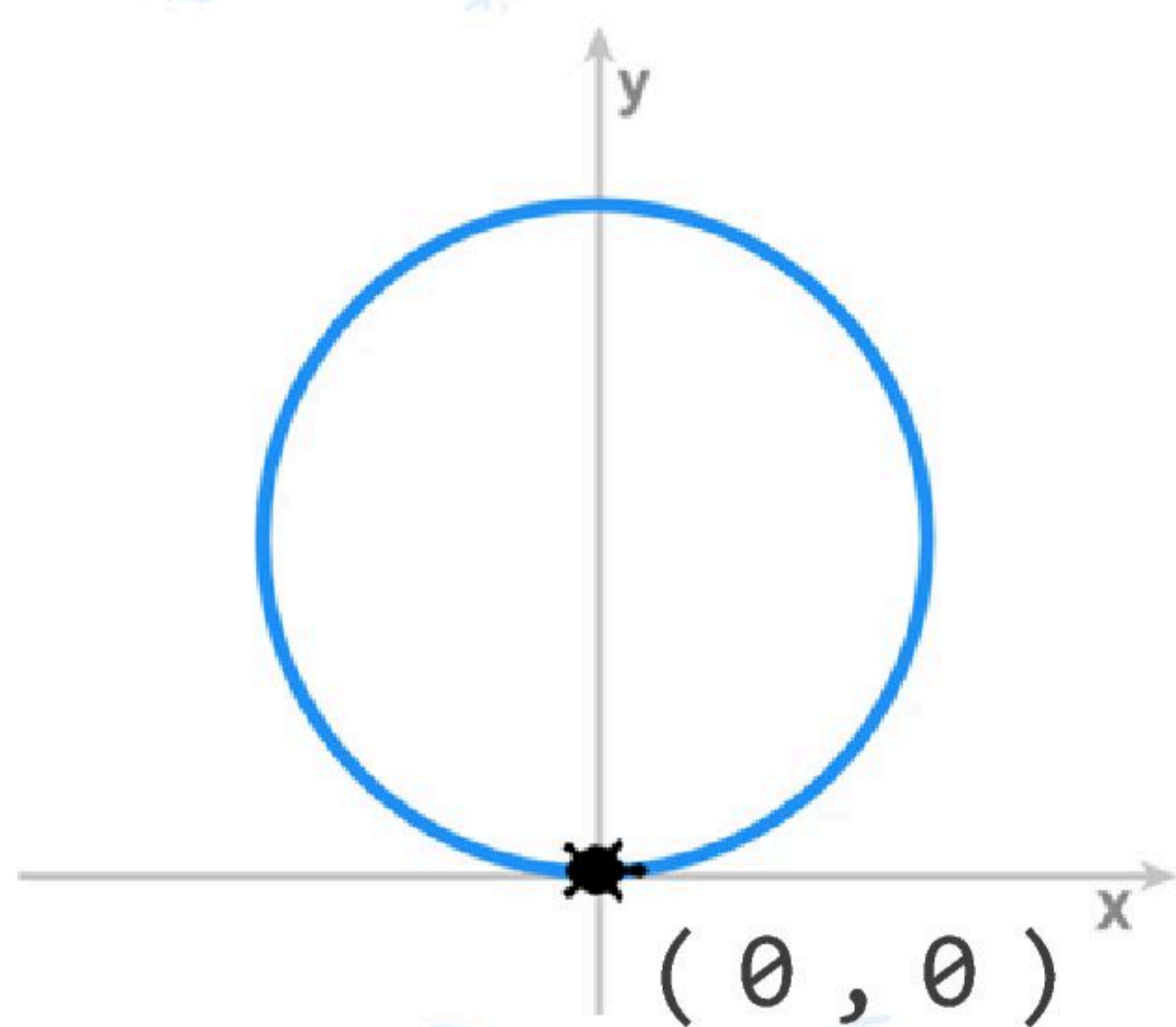
turtle.circle(radius)

绘制一个指定半径的圆

↓
圆

↓
半径

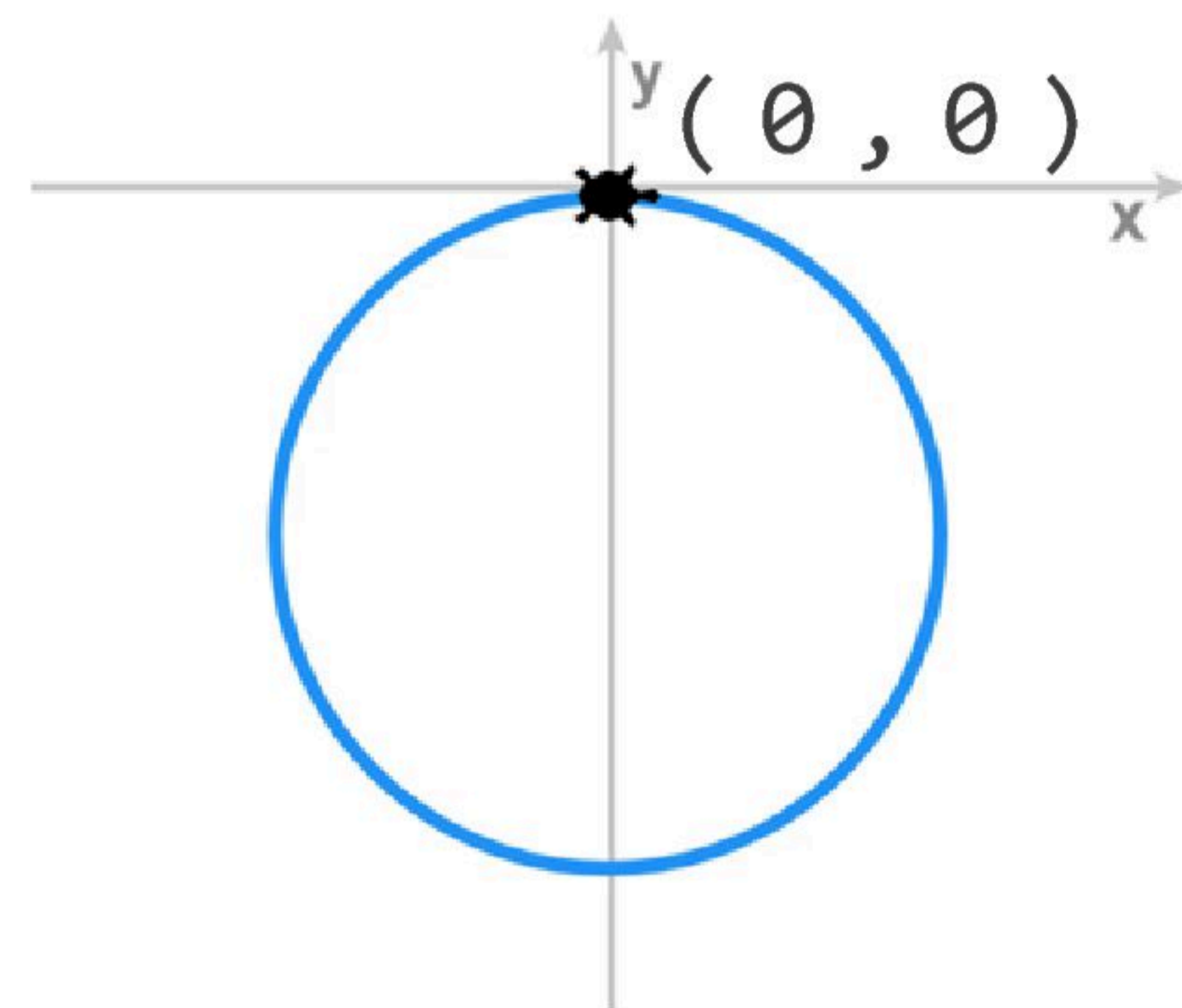
turtle.circle(100)



radius正数:

圆心在当前位置(小海龟)的左侧

turtle.circle(-100)



radius负数:

圆心在当前位置(小海龟)的右侧

1.4

绘制灯泡

要求:

绘制右图简笔画—灯泡

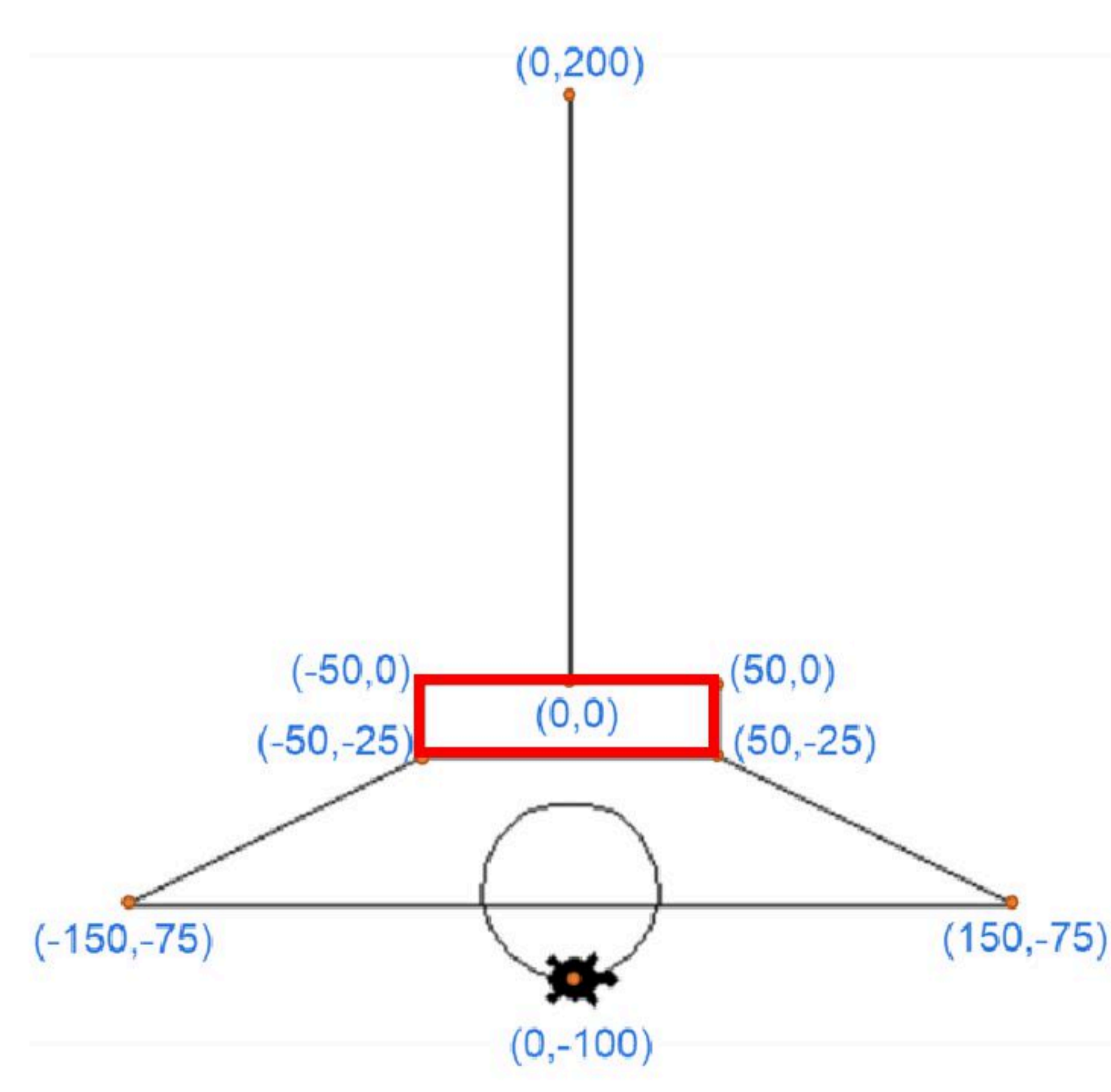
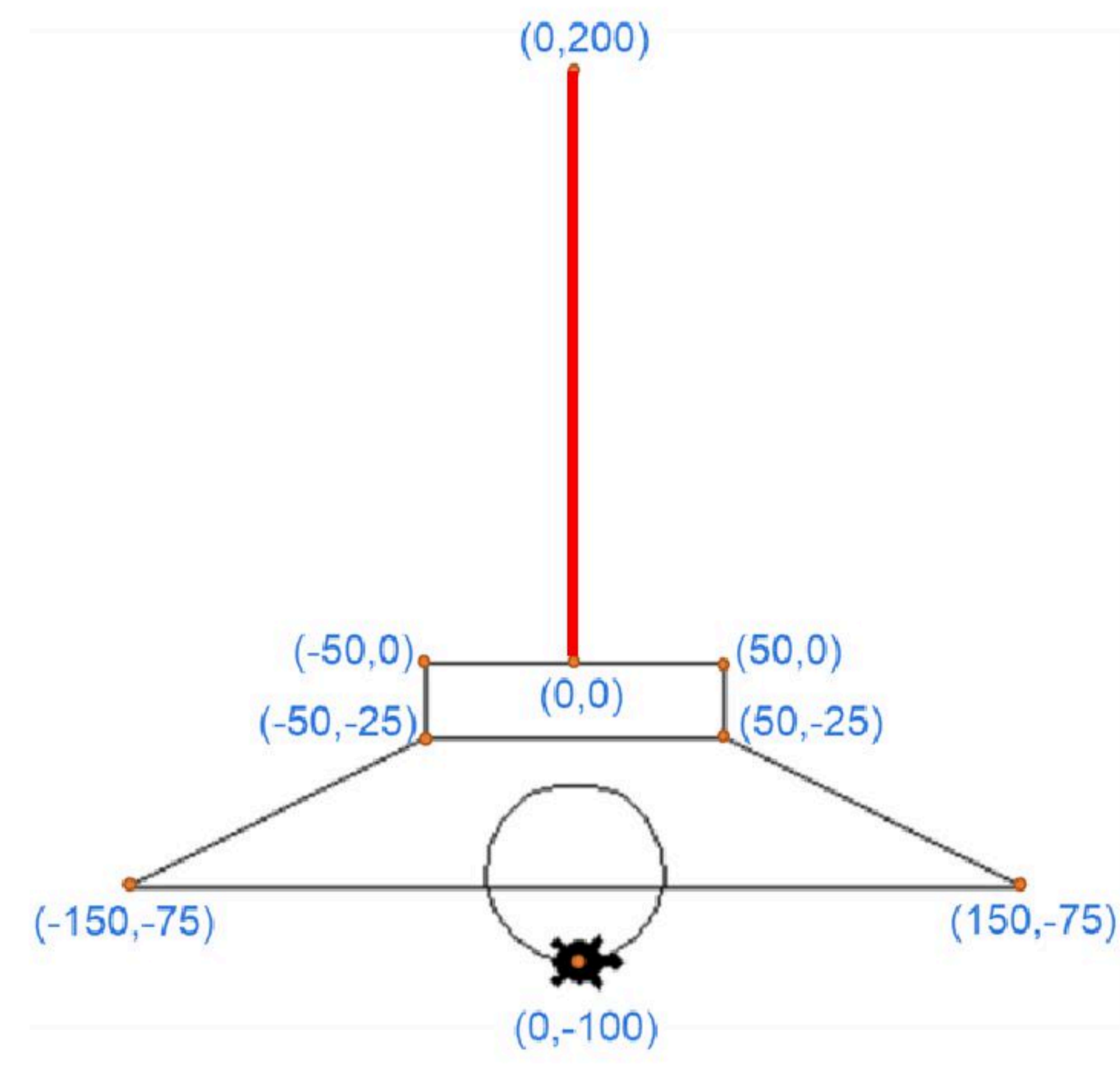
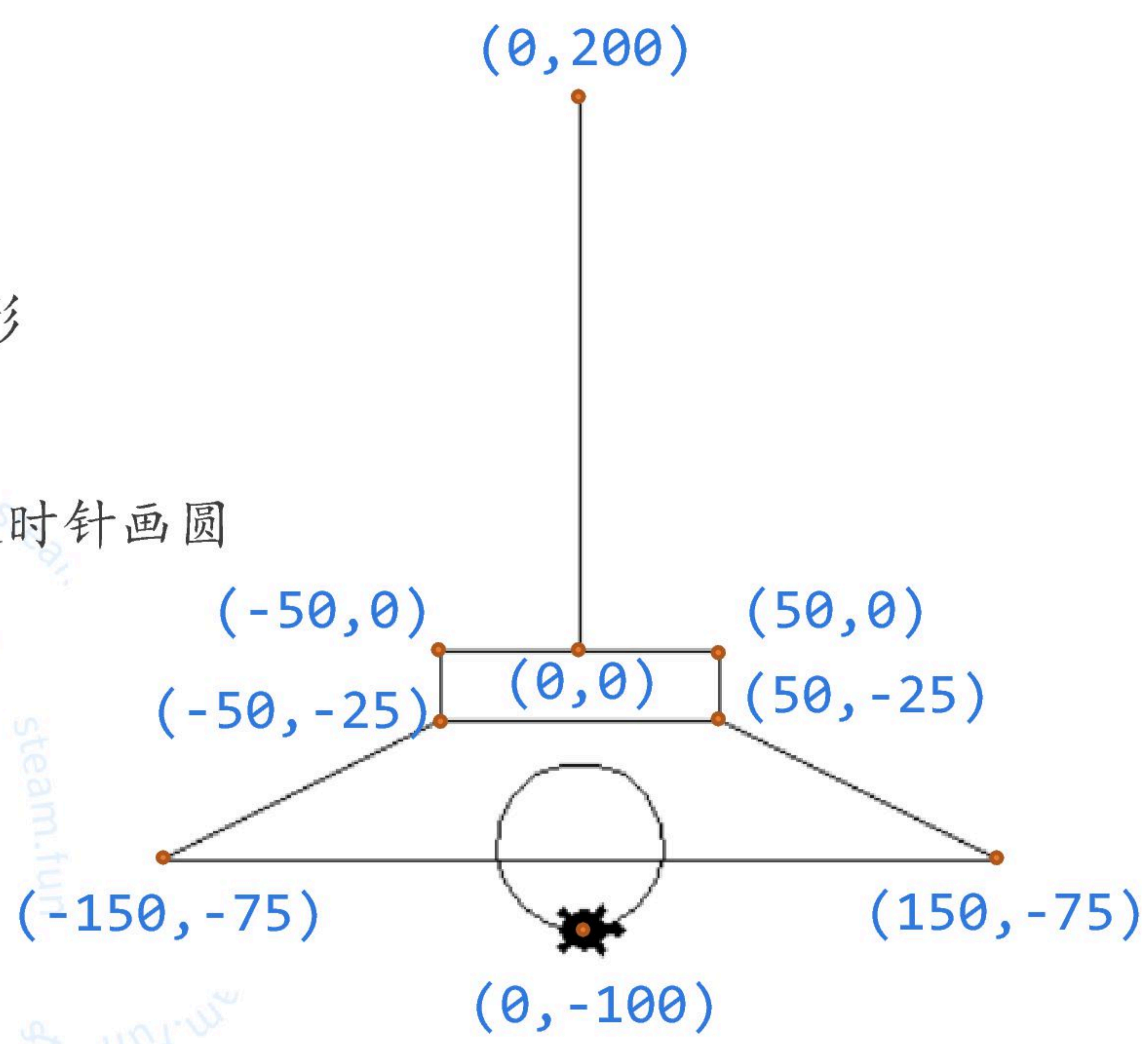
提示:

1. 组成部分: 竖线+长方形+梯形+圆形
2. 合理使用penup/pendown
3. turtle.circle()参数为正时, 逆时针画圆

```
# 导入turtle模块
import turtle as t
# 使用“turtle”形状
t.shape('turtle')
```

```
# 不画线, 移到(0,200)
t.penup()
t.goto(0,200)
t.pendown()
# 画线(0,200) -> (0,0)
t.goto(0,0)
```

```
# 画长方形(0,0) -> (50,0)
-> (50,-25) -> (-50,-25)
-> (-50,0) -> (0,0)
t.setx(50)
t.sety(-25)
t.setx(-50)
t.sety(0)
t.setx(0)
```



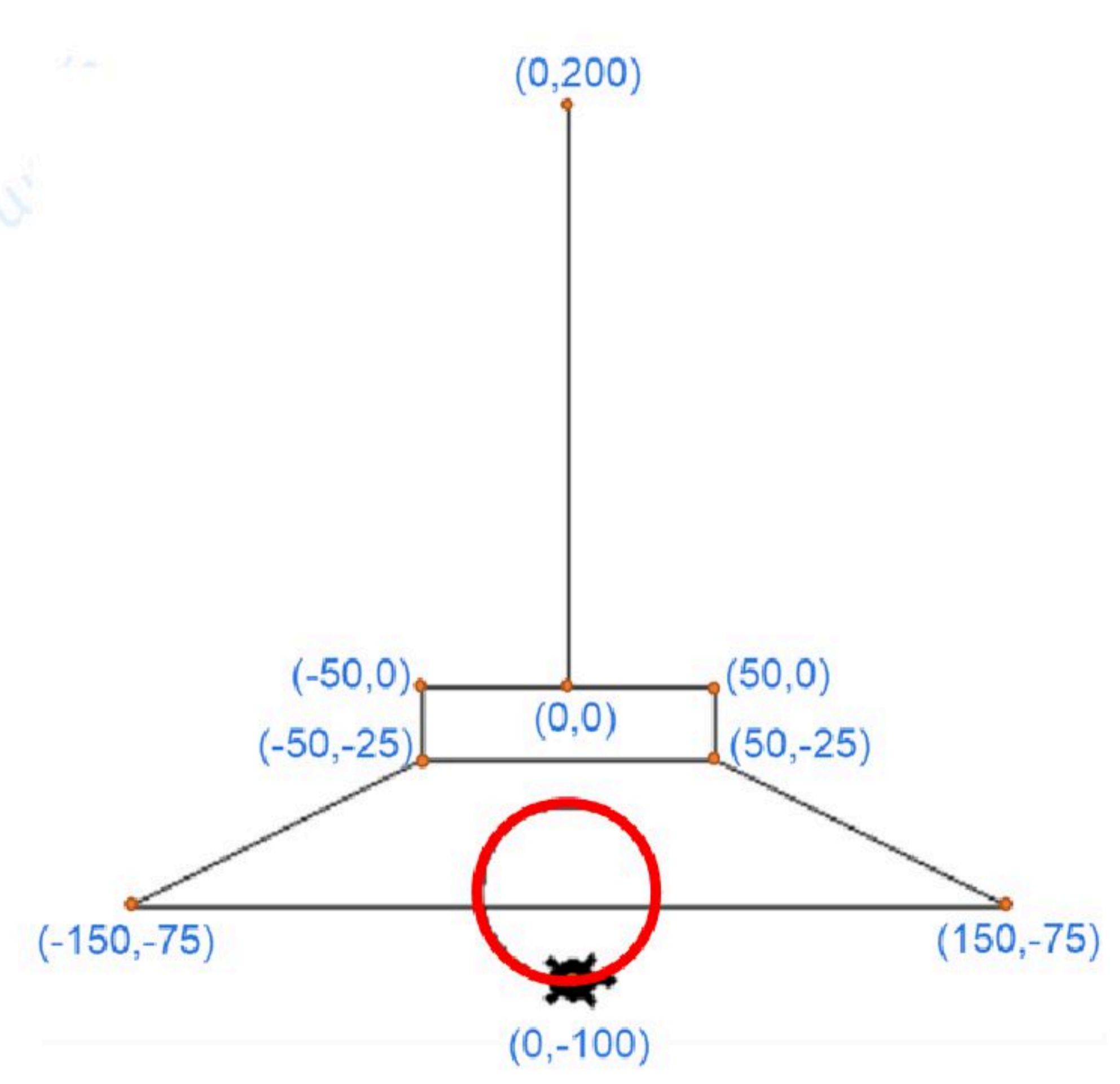
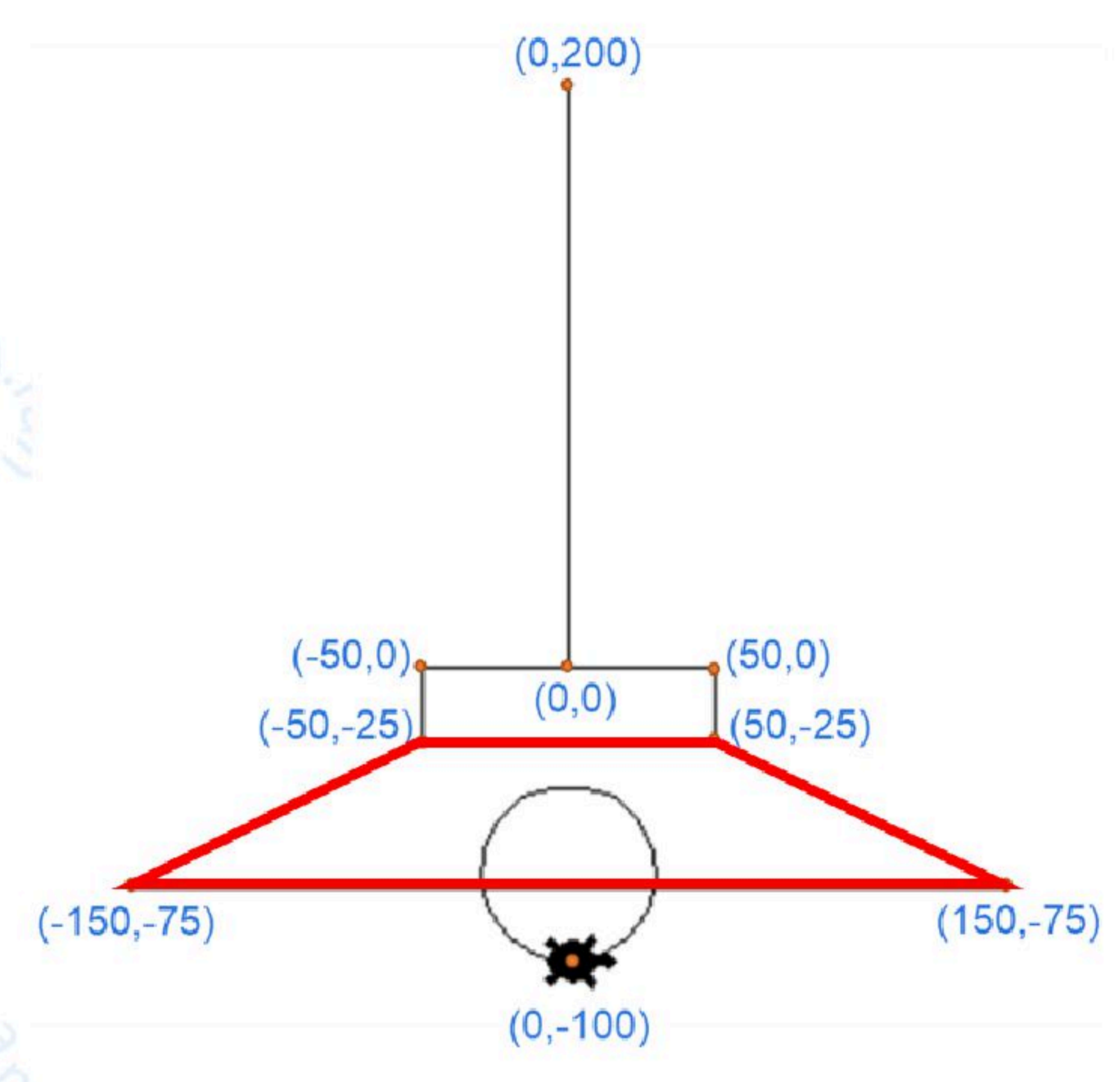
1.4

绘制灯泡

```
# 不画线, 移到(-50, -25)
t.penup()
t.goto(-50, -25)
t.pendown()
# 画梯形(-50, -25) → (50, -25)
→ (150, -75) → (-150, -75)
→ (-50, -25)
t.setx(50)
t.goto(150, -75)
t.setx(-150)
t.goto(-50, -25)
```

```
# 不画线, 移到(0, -100)
t.penup()
t.goto(0, -100)
t.pendown()
# 画圆
t.circle(30)
```

```
t.done()
```

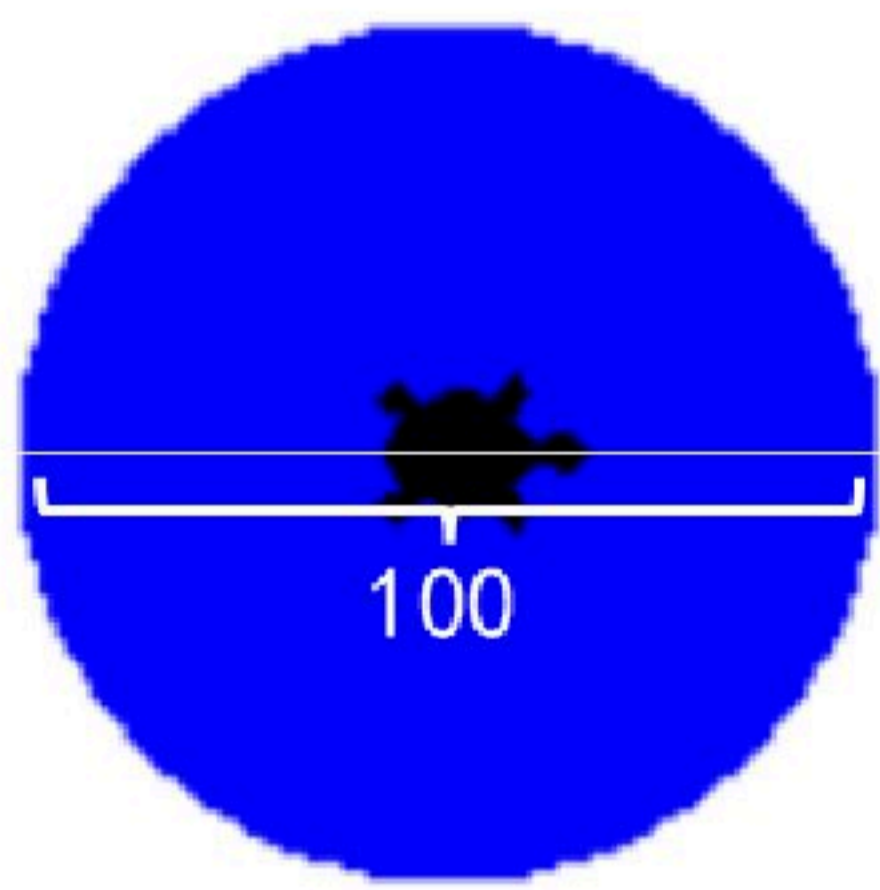


小彩蛋

- `turtle.dot(直径, 颜色)`

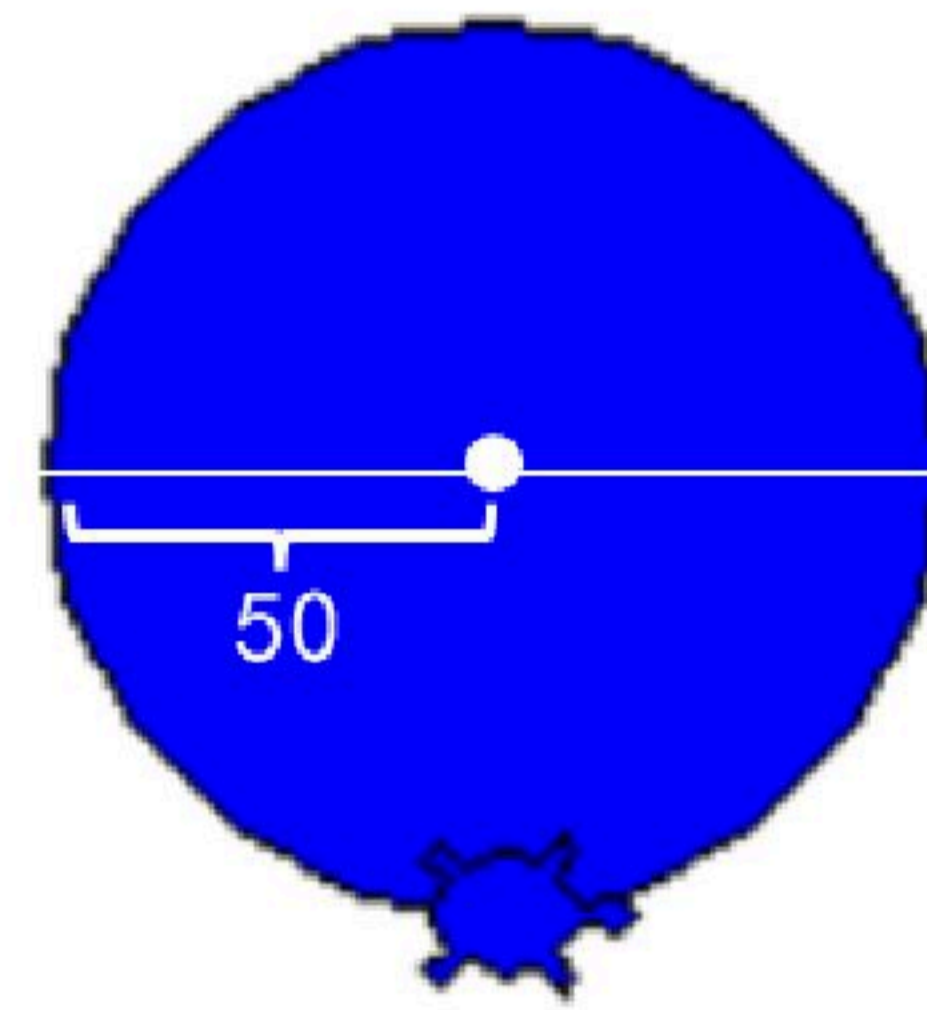
举例:

```
turtle.dot(100, 'blue')
```



对比理解:

```
turtle.circle(50)
```



3. 术语箱

setposition 设定位置

goto 去往, 前往

setheading 设定朝向

home 回家

screensize 屏幕尺寸

angle 角度

pensize 画笔尺寸

dot 点

width 宽度

height 高度

4. 课后挑战

绘制右侧的机器人图案

要求如下:

(1)正方形脸: 起始坐标 $(0,0)$, 边长100

(2)三角形嘴: 起始坐标 $(30,20)$, 边长40

(3)圆形点或圈: `circle()`或者`dot()`方法都可以, 直径自由设置, 建议为20

提示: 灵活运用`penup()/pendown()`, 控制移动过程是否留下痕迹

