

实验5

turtle 库：图形绘制

【实验目的】

- (1) 掌握 turtle 库的使用方法。
- (2) 了解常用函数的作用。

【相关知识】

1. 画布设置

(1) `turtle.screensize(width, height, bg)`: 参数分别为画布的宽、高、背景颜色。其中宽和高的单位均为像素。

(2) `turtle.setup(width, height, startx, starty)`: 参数 `width` 和 `height` 表示宽和高, 为整数时, 表示像素; 为小数时, 表示占据屏幕的比例。 `startx` 和 `starty` 表示距离屏幕左上角顶点的横向和纵向距离, 如果为空, 则窗口位于屏幕中心。

例如:

```
turtle.screensize(800,600, "green")
```

表示设置宽 800 像素、高 600 像素、背景色为绿色的画布, 位置在屏幕中央。常用的背景色有 `white`(白色)、`black`(黑色)、`grey`(灰色)、`darkgreen`(深绿色)、`gold`(金色)、`violet`(淡紫色)、`purple`(紫色)、`red`(红色)、`blue`(蓝色) 等。

又如:

```
turtle.setup(width=400,height=300, startx=200, starty=100)
```

或

```
turtle.setup(400, 300, 200, 100)
```

表示设置宽 400 像素、高 300 像素的画布, 其左上角距离屏幕左上角顶点的横向和纵向距离分别为 200 像素和 100 像素。

2. 画笔的基本参数

- (1) `turtle.pensize(width)`: 设置画笔的宽度, 单位是像素。

(2) `turtle.pencolor(color)`: 设置画笔颜色。如果没有参数传入,则返回当前画笔颜色。参数可以是字符串如 `green`、`red`,也可以是 RGB 三元组。

(3) `turtle.penup()`: 抬起画笔,之后移动画笔不绘制图形。

(4) `turtle.pendown()`: 落下画笔,之后移动画笔将绘制图形。

(5) `turtle.speed(speed)`: 设置画笔移动速度。画笔移动的速度为 $[0,10]$ 范围内的整数,数字越大则移动速度越快。

3. 画笔运动命令

(1) `turtle.forward(distance)`: 向当前画笔方向移动 `distance` 像素长度。

(2) `turtle.backward(distance)`: 向当前画笔相反方向移动 `distance` 像素长度。

(3) `turtle.right(degree)`: 顺时针移动 `degree` 角度。

(4) `turtle.left(degree)`: 逆时针移动 `degree` 角度。

(5) `turtle.goto(x,y)`: 将画笔移动到坐标为 (x,y) 的位置。

(6) `turtle.circle()`: 画圆,半径为正(负),表示圆心在画笔的左边(右边)画圆。

(7) `turtle.setx()`: 将当前 X 轴移动到指定位置。

(8) `turtle.sety()`: 将当前 Y 轴移动到指定位置。

(9) `turtle.setheading(angle)`: 设置当前朝向为 `angle` 角度。

(10) `turtle.home()`: 设置当前画笔位置为原点,朝向东。

(11) `turtle.dot(r)`: 绘制一个指定直径和颜色的圆点。

4. 画笔控制命令

(1) `turtle.fillcolor(colorstring)`: 绘制图形的填充颜色。

(2) `turtle.color(color1, color2)`: 同时设置 `pencolor=color1`、`fillcolor=color2`。

(3) `turtle.filling()`: 返回当前是否在填充状态。

(4) `turtle.begin_fill()`: 准备开始填充图形。

(5) `turtle.end_fill()`: 填充完成。

(6) `turtle.hideturtle()`: 隐藏画笔的 `turtle` 形状。

(7) `turtle.showturtle()`: 显示画笔的 `turtle` 形状。

5. 其他命令

(1) `turtle.clear()`: 清空 `turtle` 窗口,但是 `turtle` 的位置和状态不会改变。

(2) `turtle.reset()`: 清空窗口,重置 `turtle` 状态为起始状态。

(3) `turtle.undo()`: 撤销上一个 `turtle` 动作。

(4) `turtle.isvisible()`: 返回当前 `turtle` 是否可见。

(5) `stamp()`: 复制当前图形。

(6) `turtle.write(s [,font= ("font-name",font_size,"font_type")])`: 写文本,`s` 为文本内容,`font` 是字体的参数,`font-name`、`font_size` 和 `font_type` 分别为字体名称、大小和类型。`font` 参数为可选项,`font-name`、`font_size` 和 `font_type` 也是可选项。

(7) `turtle.mainloop()`或 `turtle.done()`: 启动事件循环,调用 `turtle` 库的 `mainloop` 函数,必须是图形程序中的最后一个语句。

(8) `turtle.mode(mode=None)`: 设置画笔模式为 `standard`、`logo` 或 `world`, 并执行重置。如果没有给出模式, 则返回当前模式。其中, `standard` 表示向右(东)或逆时针, `logo` 表示向上(北)或顺时针。

(9) `turtle.delay(delay=None)`: 设置或返回以毫秒为单位的绘图延迟。

(10) `turtle.begin_poly()`: 开始记录多边形的顶点。当前的画笔位置是多边形的第一个顶点。

(11) `turtle.end_poly()`: 停止记录多边形的顶点。当前的画笔位置是多边形的最后一个顶点, 将与第一个顶点相连。

(12) `turtle.get_poly()`: 返回最后记录的多边形。

【实验范例】

例 5.1 绘制正方形, 边长为 100 像素, 画笔宽度为 6 像素, 颜色为红色, 如图 1-5-1 所示。

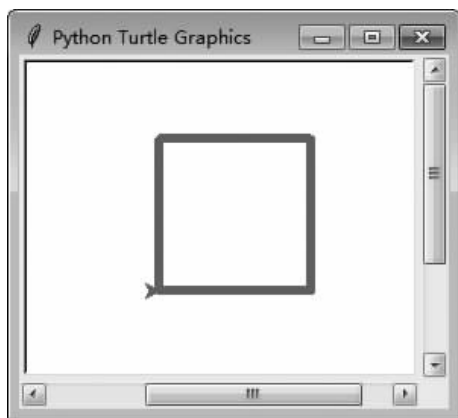


图 1-5-1 正方形

程序代码如下:

```
import turtle
turtle.color("red")
turtle.pensize(6)
for i in range(4):
    turtle.forward(100)
    turtle.left(90)
```

例 5.2 绘制 4 个圆形螺旋, 颜色分别为红、绿、黄、蓝, 如图 1-5-2 所示。

程序代码如下:

```
import turtle
turtle.color("red")
turtle.pensize(2)
turtle.speed(0)
```

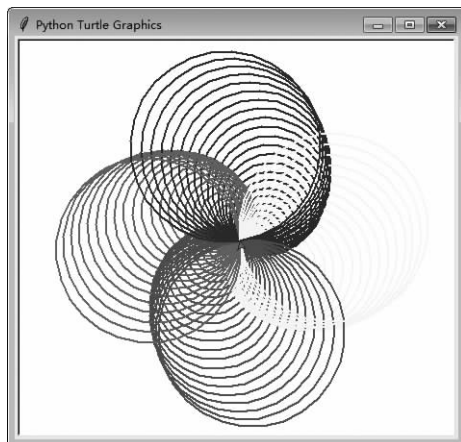


图 1-5-2 4 个圆形螺旋

```
colors=['red','green','yellow','blue']
for i in range(100):
    turtle.pencolor(colors[i%4])
    turtle.circle(i)
    turtle.left(91)
```

例 5.3 同时绘制两个六边形,如图 1-5-3 所示。

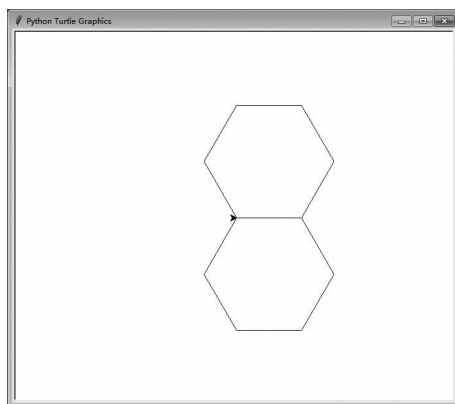


图 1-5-3 两个六边形

程序代码如下:

```
import turtle
t1=turtle.Pen()
t2=turtle.Pen()
for i in range(6):
    t1.forward(100)
    t2.forward(100)
    t1.left(60)
    t2.right(60)
```

例 5.4 绘制组合三角形,如图 1-5-4 所示。

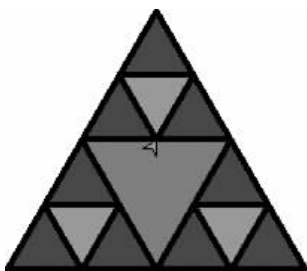


图 1-5-4 组合三角形

程序代码如下：

```
from turtle import *
d=200
d1=d/2
d2=d/4
pensize(4)
color('black','blue')
begin_fill()
for i in range(3):
    forward(d)
    left(120)
end_fill()
up()
seth(0)
forward(d1)
down()
left(60)
color('black','green')
begin_fill()
for i in range(3):
    forward(d1)
    left(120)
end_fill()
up()
seth(0)
forward(d2)
down()
left(60)
color('black','red')
begin_fill()
for i in range(3):
    forward(d2)
    left(120)
end_fill()
up()
seth(0)
```

```
backward(d1)
down()
seth(0)
left(60)
color('black','red')
begin_fill()
for i in range(3):
    forward(d2)
    left(120)
end_fill()
up()
seth(60)
forward(d1)
down()
seth(0)
left(60)
color('black','red')
begin_fill()
for i in range(3):
    forward(d2)
    left(120)
end_fill()
```

例 5.5 绘制方形螺旋,如图 1-5-5 所示。

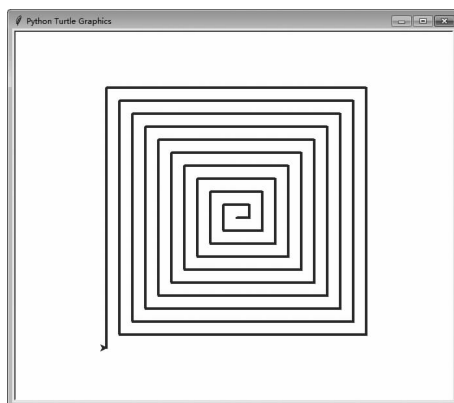


图 1-5-5 方形螺旋

程序代码如下:

```
from turtle import *
d=20
w=20
pensize(4)
color('blue')
for i in range(20):
    forward(d)
    left(90)
    forward(d)
```

```
left(90)  
d=d+w
```

例 5.6 绘制圆形螺旋,如图 1-5-6 所示。

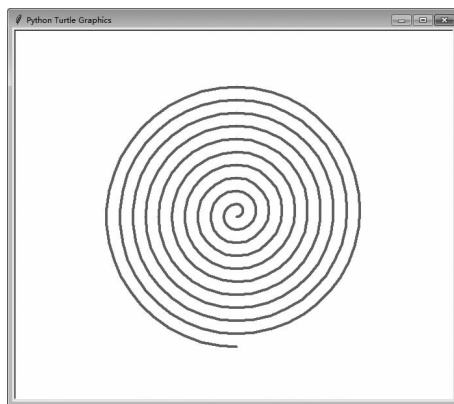


图 1-5-6 圆形螺旋

程序代码如下：

```
from turtle import *  
color('red')  
R=10  
pensize(4)  
for i in range(20):  
    circle(R,180)  
    R=R+10  
hideturtle()
```

例 5.7 绘制正弦曲线,如图 1-5-7 所示。

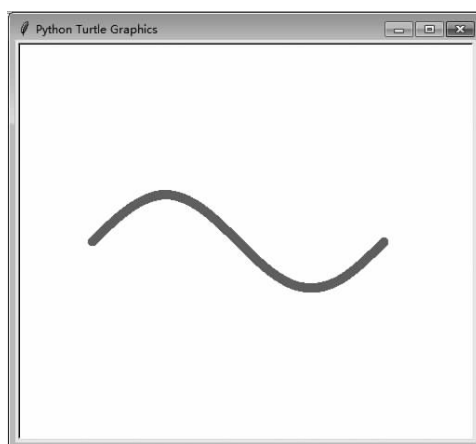


图 1-5-7 正弦曲线

程序代码如下：

```
from math import *
setup(800,600)
penup()
goto(-300,0)
pendown()
i=0
pensize(10)
pencolor('red')
while i<=6.28:
    f=sin(i)
    i+=0.01
    goto(i*50-300,f*50)
hideturtle()
```

例 5.8 书写文字,如图 1-5-8 所示。

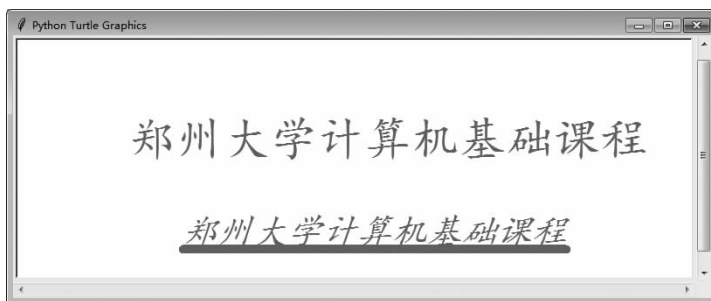


图 1-5-8 书写文字

程序代码如下:

```
from turtle import *
pensize(4)
setup(800,300)
penup()
goto(-200,-100)
pendown()
pensize(10)
pencolor('red')
write('郑州大学计算机基础课程',move=True,font=('楷体',30,'italic'))
penup()
goto(-260,0)
write('郑州大学计算机基础课程',font=('楷体',40))
hideturtle()
```

【实验任务】

- (1) 绘制如图 1-5-9 所示图形,最小的三角形边长 20 像素。
- (2) 绘制如图 1-5-10 所示图形,小三角形边长 20 像素。

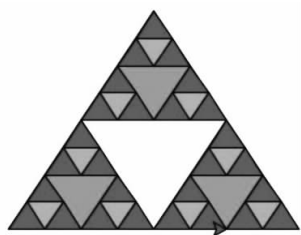


图 1-5-9 实验任务 1



图 1-5-10 实验任务 2

【拓展训练】

- (1) 绘制国际象棋棋盘, 每个格子边长 40 像素。
- (2) 绘制运动场及跑道, 尺寸自定。
- (3) 绘制星空图, 要求有圆月, 星星为 20 个小五角星, 随机分布。