

实例 2：绘制多角星

如果你喜欢作画，一定要尝试一下 Python 的内置模块——turtle 模块，turtle 是一个专门的绘图模块，你可以利用该模块通过程序绘制一些简单图形。

本实例要求编写程序，使用 turtle 模块绘制一个如图 1 所示的多角星。

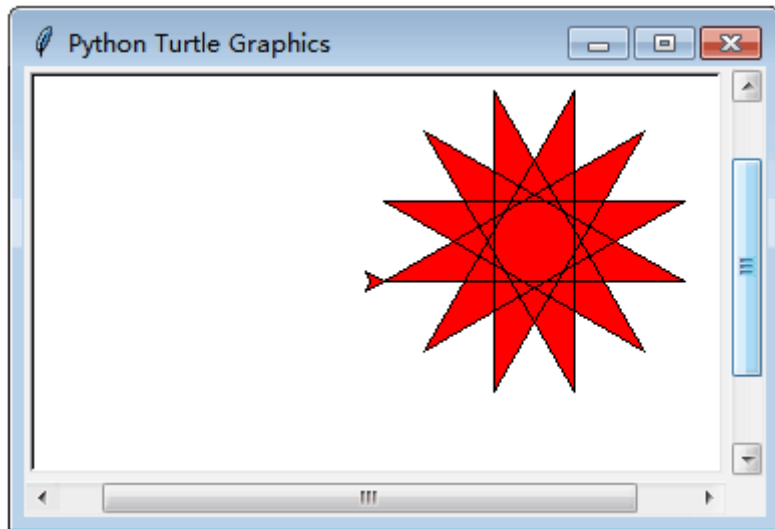


图1 多角星示例

实例目标

- 掌握模块的导入方式
- 掌握 turtle 模块的基本用法

实例分析

本实例的多角星需要借用 turtle 模块，大家可自行从官方文档中 (<https://docs.python.org/3.7/library/turtle.html#turtle.pos>) 了解 turtle 模块的基本用法，了解之后便可以使用 turtle 模块中的相关函数进行绘制。

绘制多角星的功能可以视为将画笔沿逆时针方向旋转固定角度画直线（黑色）的操作，直到当前位置的绝对值小于 1 时就结束绘图，之后在画好的图形上填充颜色（红色）即可。此多角星绘制的过程中涉及到的 turtle 模块中的函数及说明如下：

- (1) color(): 设置画笔颜色和填充颜色。
- (2) begin_fill(): 开始填充。
- (3) end_fill(): 停止填充。
- (4) forward(): 将画笔向前方移动指定的距离。
- (5) left(): 将画笔逆时针旋转指定的角度。

- (6) `pos()`: 获取画笔当前的位置。
- (7) `done()`: 启动事件循环，必须位于末尾位置。

代码实现

完整的代码如下所示：

```
import turtle as t

t.color('black', 'red')  # 设置画笔颜色、填充颜色

t.setup(450, 400)        # 设置主窗口的大小为 450*400

t.begin_fill()           # 在绘制要填充的形状之前调用

while True:

    t.forward(150)        # 将当前画笔移动 150 个像素

    t.left(150)           # 将当前画笔逆时针旋转 150 度

    if abs(t.pos()) < 1:  # 若当前位置的绝对值小于 1 则跳出循环

        break

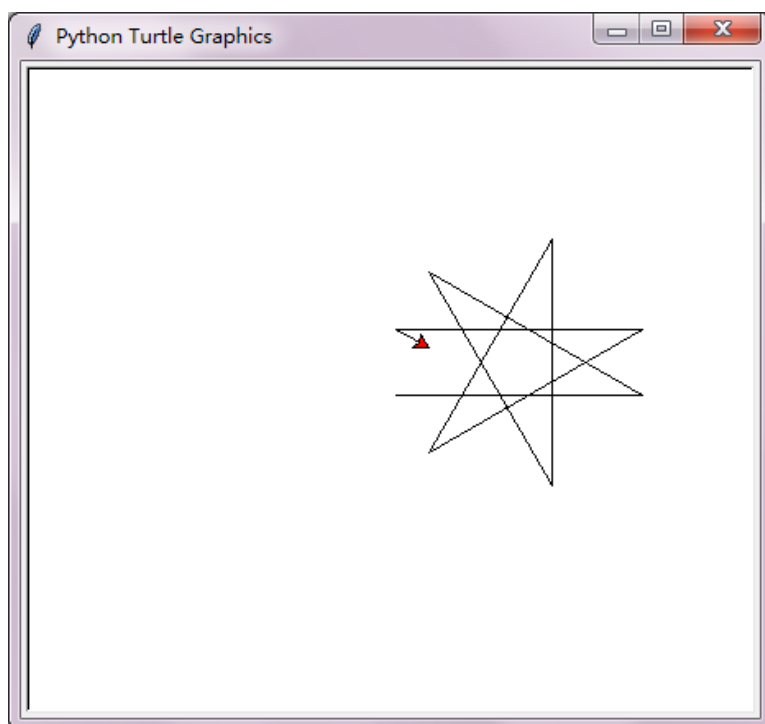
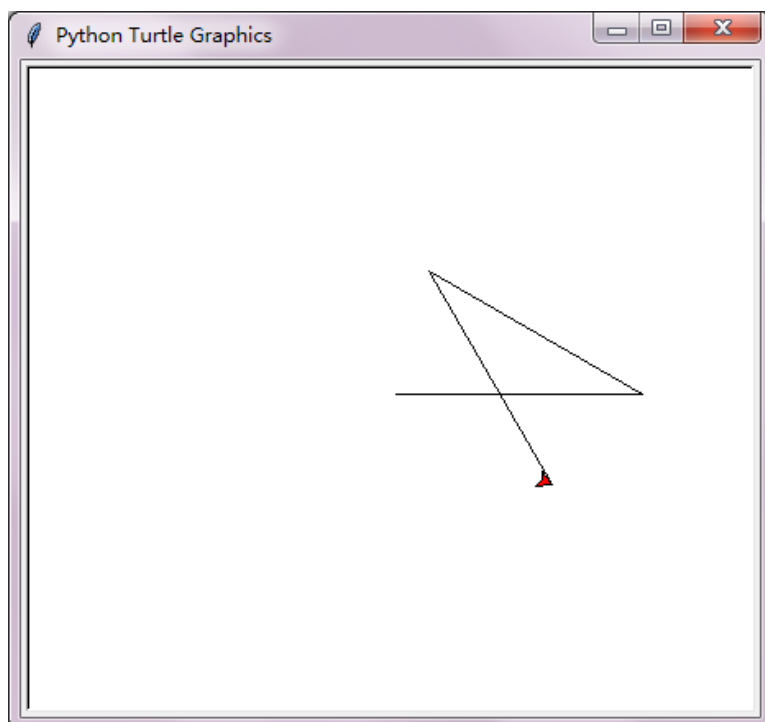
t.end_fill()             # 结束填充

t.done()                 # 停止画笔绘制，窗体不关闭
```

以上代码首先使用 `import` 语句导入了 `turtle` 模块，并起别名为 `t`，其次调用 `color()` 函数分别设置了画笔颜色与填充颜色，调用 `setup()` 函数设置主窗口的大小，然后使用 `while` 语句绘制多角星：将当前画笔移动 150 像素画直线，逆时针旋转 150 度，判断当前画笔所处位置的绝对值是否小于 1，小于 1 则结束循环，不小于 1 则重复上述步骤，最后调用 `done()` 函数停止绘制，保持主窗口不关闭。

代码测试

运行程序，绘制过程如图 2 所示。



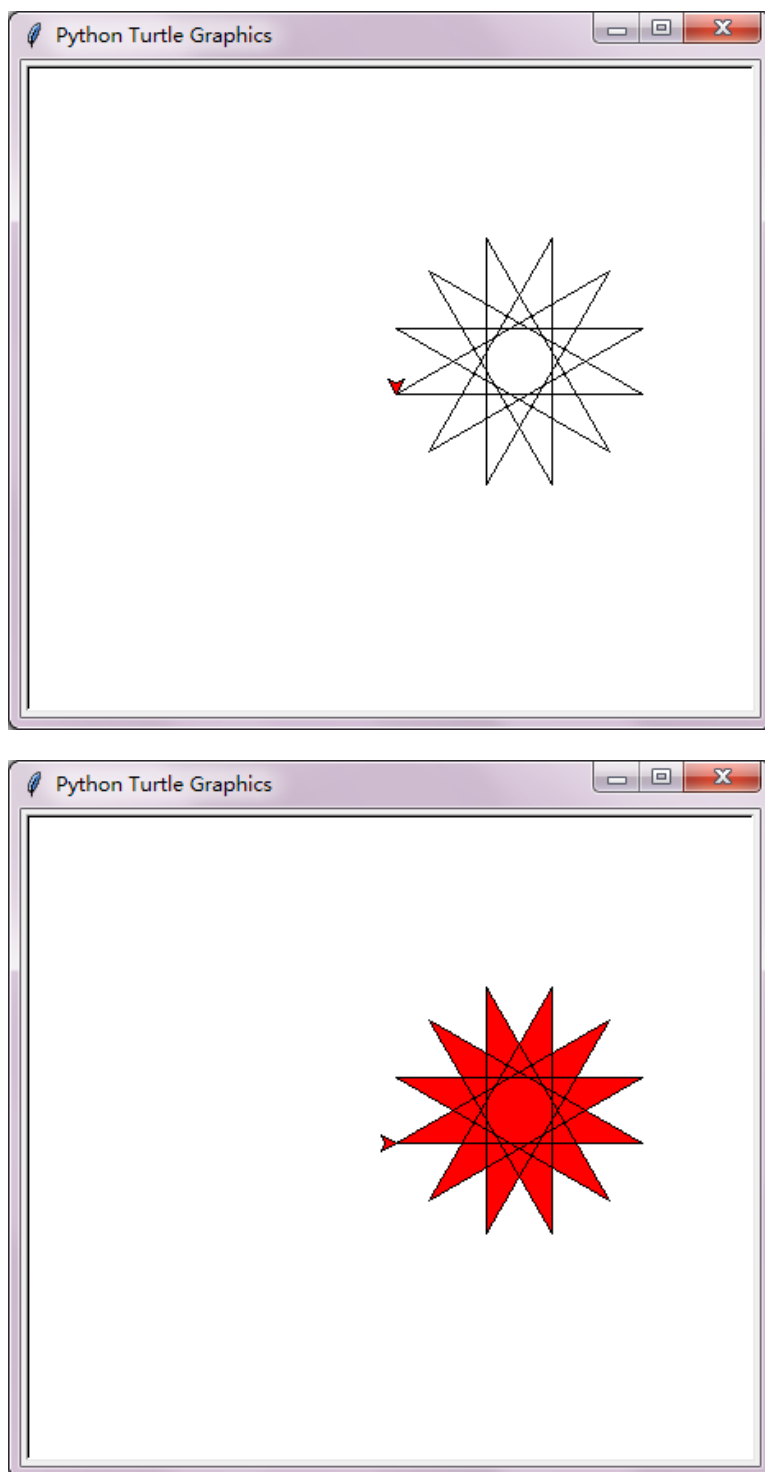


图2 多角星绘制过程