

Python语言基础与应用

基本扩展模块 / 海龟作图

陈斌 北京大学 gischen@pku.edu.cn



海龟作图

- › 基本概念与方法
- › 简单图形
- › 树图形
- › 分形图形

海龟作图turtle module

- › Python内置，随时可用
- › 其意象为模拟海龟在沙滩上爬行而留下的足迹



海龟作图的基本方法

› turtle模块

内置模块，从LOGO语言借鉴而来

› 属性

位置、方向、画笔（颜色、线条宽度等）

› 指令

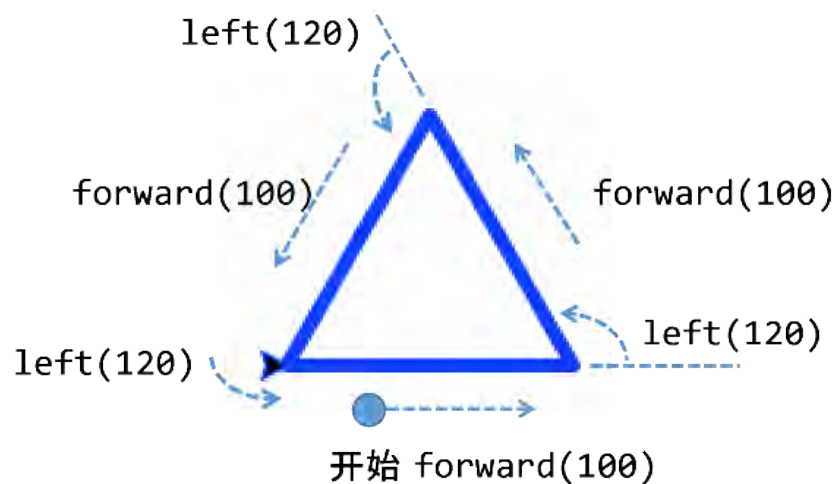
● 画笔运动命令

前/后移动、左/右转动、作画速度等

● 画笔控制命令

抬起/放下、画笔宽度、画笔颜色、填充颜色等

海龟作图的基本方法



```
import turtle # 导入turtle模块
p = turtle.Pen() # 创建一支画笔 (海龟)
p.pencolor('blue') # 设置画笔颜色
p.pensize(5) # 设置画笔的粗细为5
p.forward(100) # 最初画笔 (海龟) 朝向正右方, 向前画长度为100的直线
p.left(120) # 画笔 (海龟) 向左转120度
p.forward(100) # 向前画长度为100的直线
p.left(120) # 画笔 (海龟) 向左转120度
p.forward(100) # 向前画长度为100的直线
p.left(120) # 画笔 (海龟) 向左转120度
```

绘制简单图形

> 画直线

```
import turtle

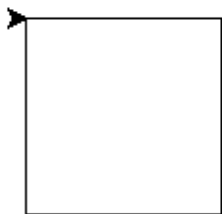
t = turtle.Turtle()
w = turtle.Screen()
t.forward(100)

turtle.done()
```



> 画正方形

```
import turtle
t = turtle.Turtle()
w = turtle.Screen()
for i in range(4):
    t.forward(100)
    t.right(90)
turtle.done()
```

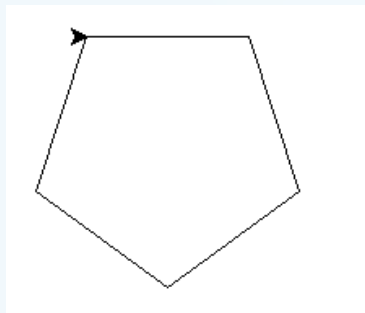


绘制简单图形

画多边形

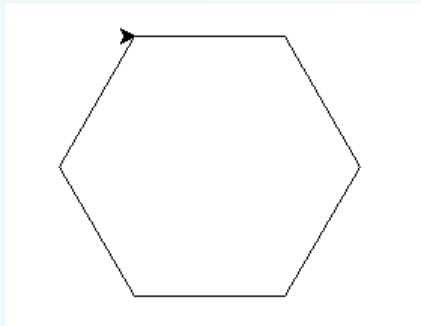
五边形

```
import turtle
t = turtle.Turtle()
w = turtle.Screen()
for i in range(5):
    t.forward(100)
    t.right(72)
turtle.done()
```



六边形

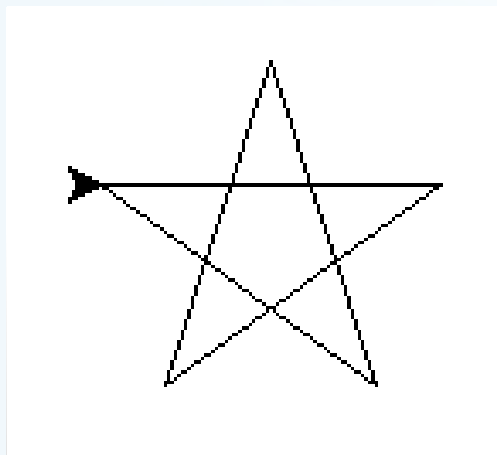
```
import turtle
t = turtle.Turtle()
w = turtle.Screen()
for i in range(6):
    t.forward(100)
    t.right(60)
turtle.done()
```



绘制简单图形

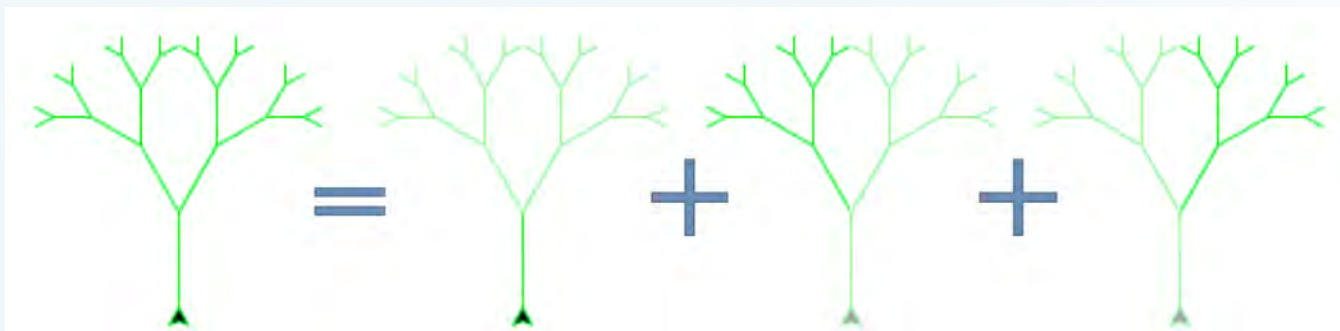
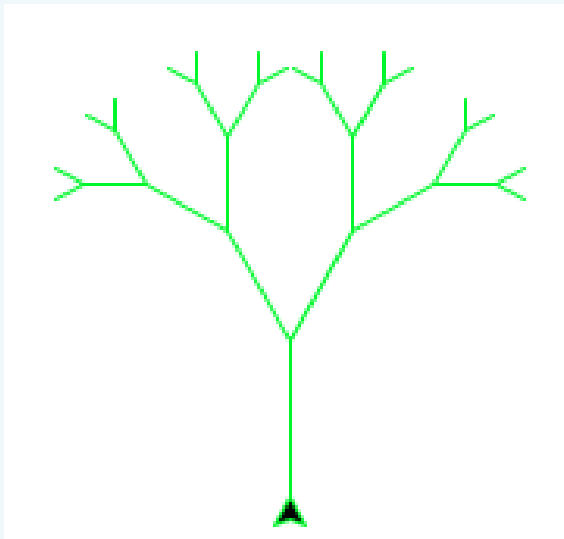
› 画五角星

```
import turtle
t = turtle.Turtle()
w = turtle.Screen()
for i in range(5):
    t.forward(100)
    t.right(144)
turtle.done()
```



绘制树图形

调用递归画树



大树

树干

左小树

右小树

分形树：代码

```
import turtle
```

```
def tree(branchLen,t):
```

```
    if branchLen > 5:
```

```
        t.forward(branchLen)
```

```
        t.right(20)
```

```
        tree(branchLen-15,t)
```

```
        t.left(40)
```

```
        tree(branchLen-15,t)
```

```
        t.right(20)
```

```
        t.backward(branchLen)
```

最小规模，0直接退出

右倾20度

减小规模，树干减15

回左倾40度，即左20

减小规模，树干减15

回右倾20度，即回正

海龟

画树干

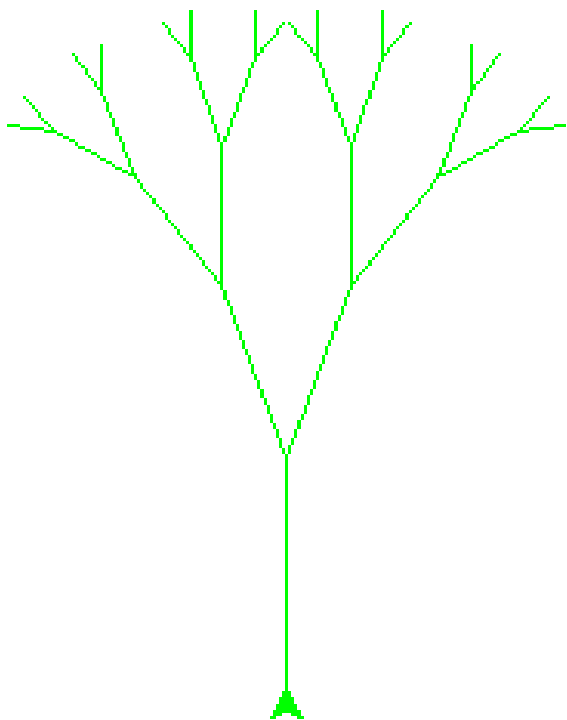
调用自身

海龟回到原位置

分形树：运行

注意海龟作图的次序

先画树干，再画右树枝，最后画左树枝：与递归函数里的流程一致



生成海龟

海龟位置调整

画树，树干长度75

```
def main():  
    t = turtle.Turtle()  
    myWin = turtle.Screen()  
    t.left(90)  
    t.up()  
    t.backward(100)  
    t.down()  
    t.color("green")  
    tree(75,t)  
    myWin.exitonclick()
```

main()

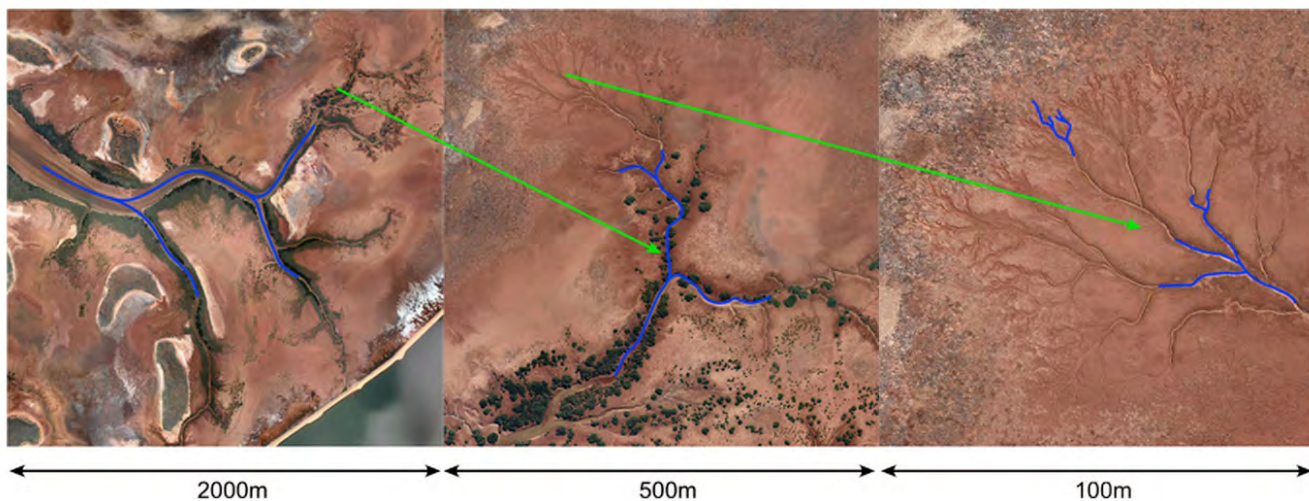
绘制分形图

› 分形(fractal)的基本概念

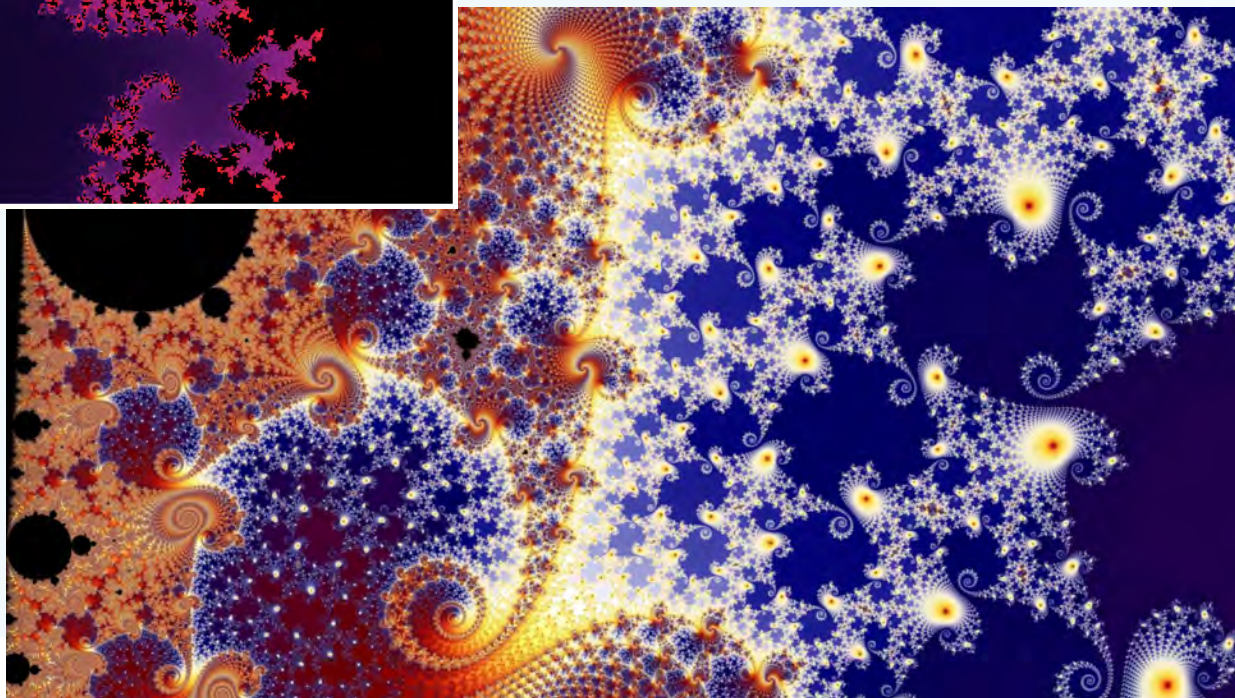
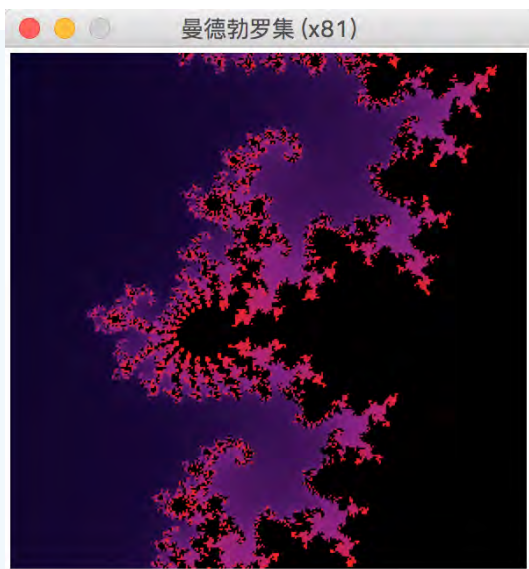
部分与整体以某种方式相似的形体

› “分形学之父”

数学家曼德勃罗 (Benoit B. Mandelbrot)

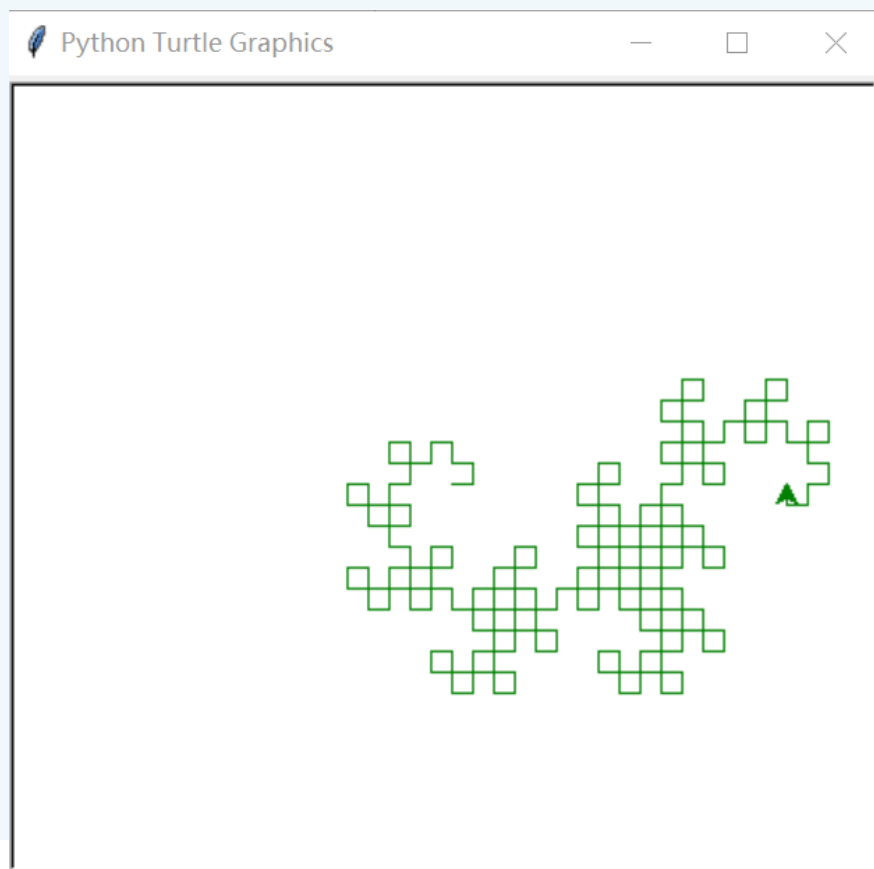


分形图：曼德勃罗集



绘制分形曲线：龙形曲线

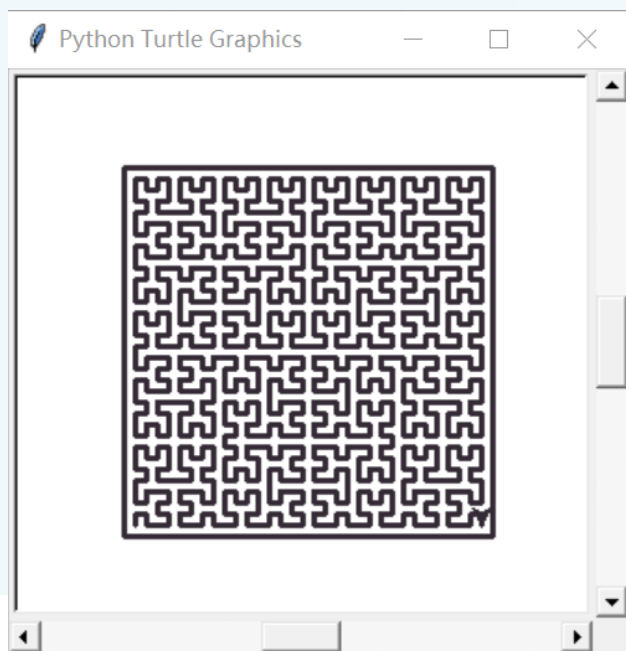
- 自相似分形曲线



绘制分形图：Hilbert曲线

› Hilbert曲线

- 空间填充曲线，可以线性地贯穿二维或者更高维度的每个离散单元并进行排序和编码
- 可用于将高维空间中沒有良好顺序的数据映射到一维空间



绘制分形图：谢尔宾斯基三角形

谢尔宾斯基三角形

- 由3个相同的谢尔宾斯基三角形按照品字形拼叠而成
- 面积为0，周长为无穷大
- 我们只能画出维数(degree)有限的近似图形

