

# python

The Python logo, consisting of two interlocking snakes, one blue and one yellow, is positioned below the word "python".

```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")

for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```

Python语言程序设计

# 实例11：自动轨迹绘制

---



嵩 天  
北京理工大学





# "自动轨迹绘制"问题分析

# 问题分析

## 自动轨迹绘制

- **需求：根据脚本来绘制图形？**
- **不通过写代码而通过写数据绘制轨迹**
- **数据脚本是自动化最重要的第一步**

# 问题分析

## 自动轨迹绘制

300,0,144,1,0,0

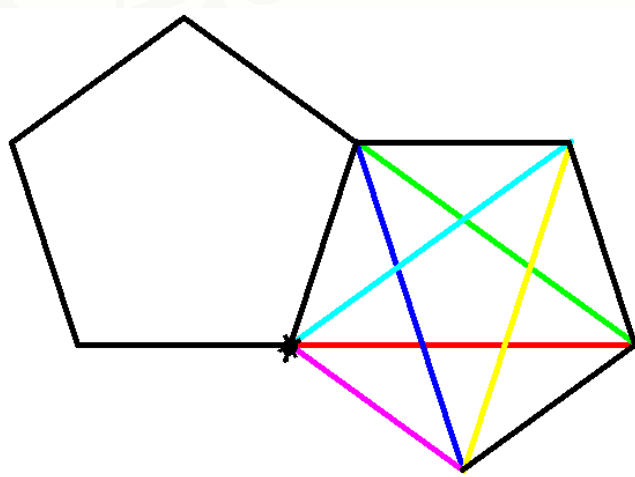
300,0,144,0,1,0

300,0,144,0,0,1

300,0,144,1,1,0

300,0,108,0,1,1

184,0,72,1,0,1





# "自动轨迹绘制"实例讲解

# 自动轨迹绘制

## 基本思路

- 步骤1：定义数据文件格式（接口）
- 步骤2：编写程序，根据文件接口解析参数绘制图形
- 步骤3：编制数据文件

# 数据接口定义

非常具有个性色彩

300,0,144,1,0,0

300,1,144,0,1,0

行进距离

转向判断

0: 左转 1:右转

转向角度

RGB三个通道颜色

0-1之间浮点数



```
#AutoTraceDraw.py
```

```
import turtle as t
```

```
t.title('自动轨迹绘制')
```

```
t.setup(800, 600, 0, 0)
```

```
t.pencolor("red")
```

```
t.pensize(5)
```

```
#数据读取
```

```
datals = []
```

```
f = open("data.txt")
```

```
for line in f:
```

```
    line = line.replace("\n", "")
```

```
    datals.append(list(map(eval, line.split(","))))
```

```
f.close()
```

```
#自动绘制
```

```
for i in range(len(datals)):
```

```
    t.pencolor(datals[i][3], datals[i][4], datals[i][5])
```

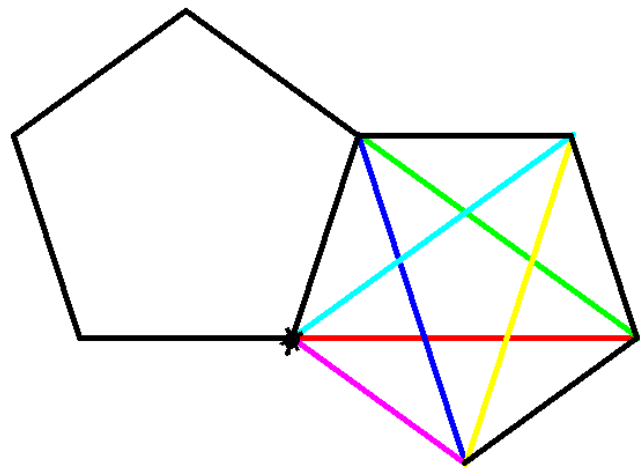
```
    t.fd(datals[i][0])
```

```
    if datals[i][1]:
```

```
        t.right(datals[i][2])
```

```
    else:
```

```
        t.left(datals[i][2])
```



# 数据文件

300,0,144,1,0,0

300,0,144,0,1,0

300,0,144,0,0,1

300,0,144,1,1,0

300,0,108,0,1,1

184,0,72,1,0,1

184,0,72,0,0,0

184,0,72,0,0,0

184,0,72,0,0,0

184,1,72,1,0,1

184,1,72,0,0,0

184,1,72,0,0,0

184,1,72,0,0,0

184,1,72,0,0,0

184,1,720,0,0,0

data.txt

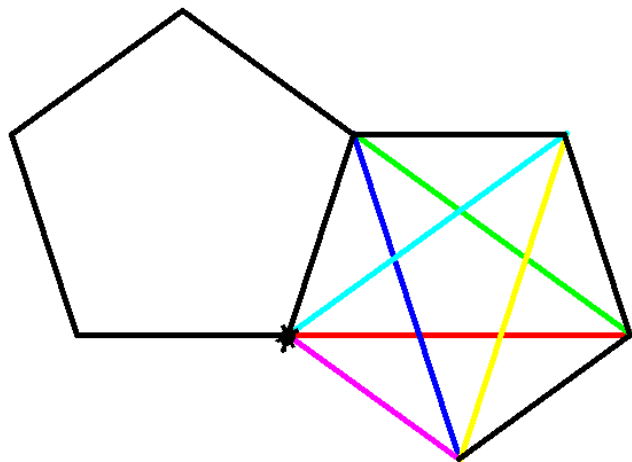


**准备好电脑，与老师一起编码吧！**



# "自动轨迹绘制"举一反三

```
import turtle as t
t.title('自动轨迹绘制')
t.setup(800, 600, 0, 0)
t.pencolor("red")
t.pensize(5)
datals = []
f = open("data.txt")
for line in f:
    line = line.replace("\n", "")
    datals.append(list(map(eval, line.split(","))))
f.close()
for i in range(len(datals)):
    t.pencolor(datals[i][3], datals[i][4], datals[i][5])
    t.fd(datals[i][0])
    if datals[i][1]:
        t.right(datals[i][2])
    else:
        t.left(datals[i][2])
```



# 举一反三

## 理解方法思维

- 自动化思维：数据和功能分离，数据驱动自动运行
- 接口化设计：格式化设计接口，清晰明了
- 二维数据应用：应用维度组织数据，二维数据最常用

# 举一反三

## 应用问题的扩展

- 扩展接口设计，增加更多控制接口
- 扩展功能设计，增加弧形等更多功能
- 扩展应用需求，发展自动轨迹绘制到动画绘制



# 小花絮



# 如何进一步提高Python编程能力？

三个步骤，请参考：<https://python123.io/python>

- **第一步：**学好Python语法，即掌握非库功能，练好内功
- **第二步：**学好Python领域，数据分析、Web开发、人工智能，找准了深入学
- **第三步：**学好计算机专业知识，构建“系统”是本领，需要专业计算机知识

数据结构、算法、计算机网络、组成原理、操作系统、网络安全、体系结构、软件工程...

