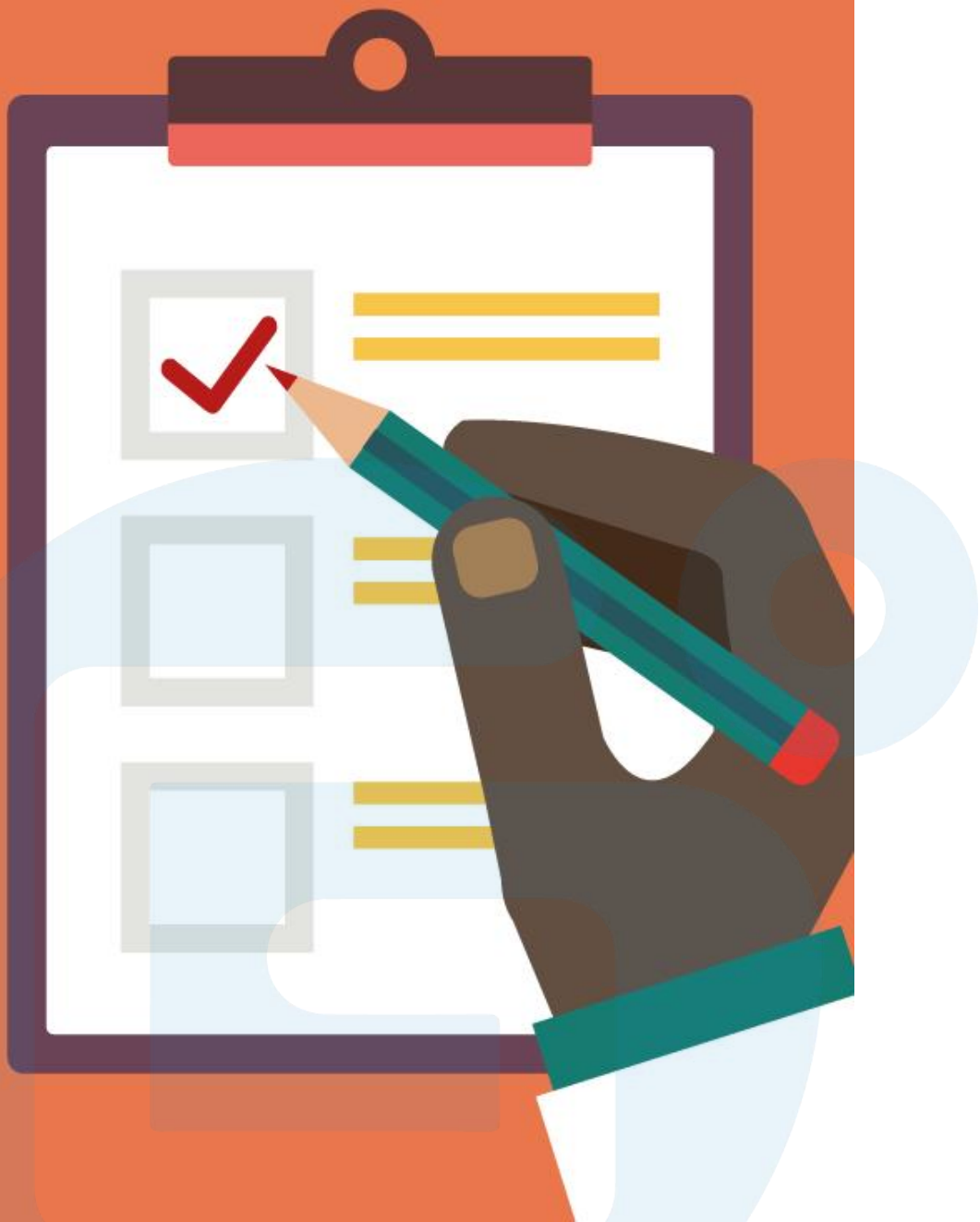


Talk is cheap, show me the code

## 第二课：Python 基础绘图

Python初阶入门课程系列



---

# OUTLINE

## ➤ 实例

### ➤ 程序解析

### ➤ **Turtle**库的基本特点

### ➤ **Turtle**库的绘图流程

### ➤ 完整练习

---

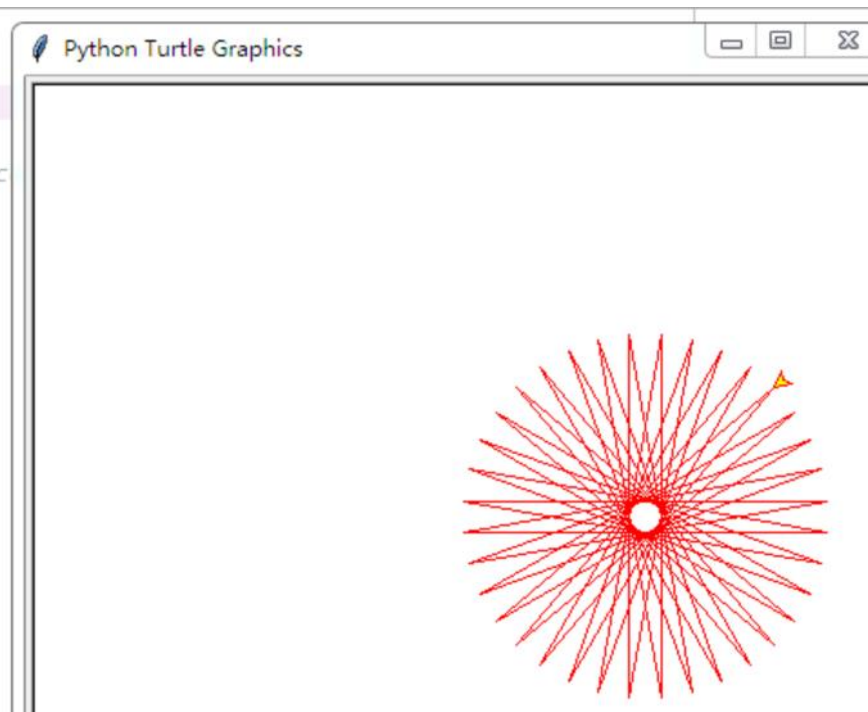


# 一 实例



# 实例

```
1 # coding=utf-8
2 import turtle
3
4
5 # 同时设置pencolor=color1, fillcolor=c
6 turtle.color("red", "yellow")
7
8 turtle.begin_fill()
9 for _ in range(50):
10     turtle.forward(200)
11     turtle.left(170)
12     turtle.end_fill()
13
14 turtle.mainloop()
```

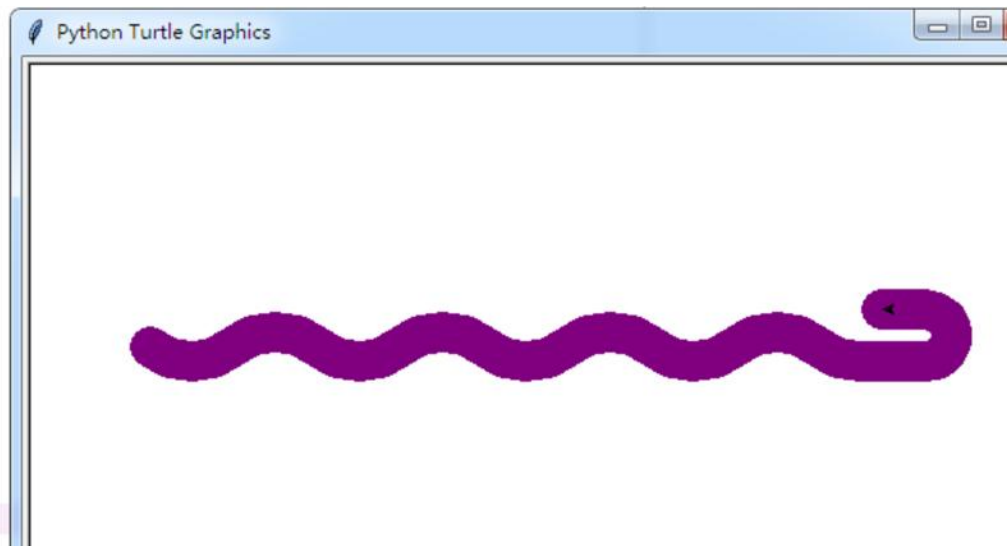


```
import turtle
turtle.color("red", "yellow")
turtle.begin_fill()
turtle.speed(100)
for _ in range(50):
    turtle.forward(200)
    turtle.left(170)
    turtle.end_fill()

turtle.mainloop()
```

# 实例

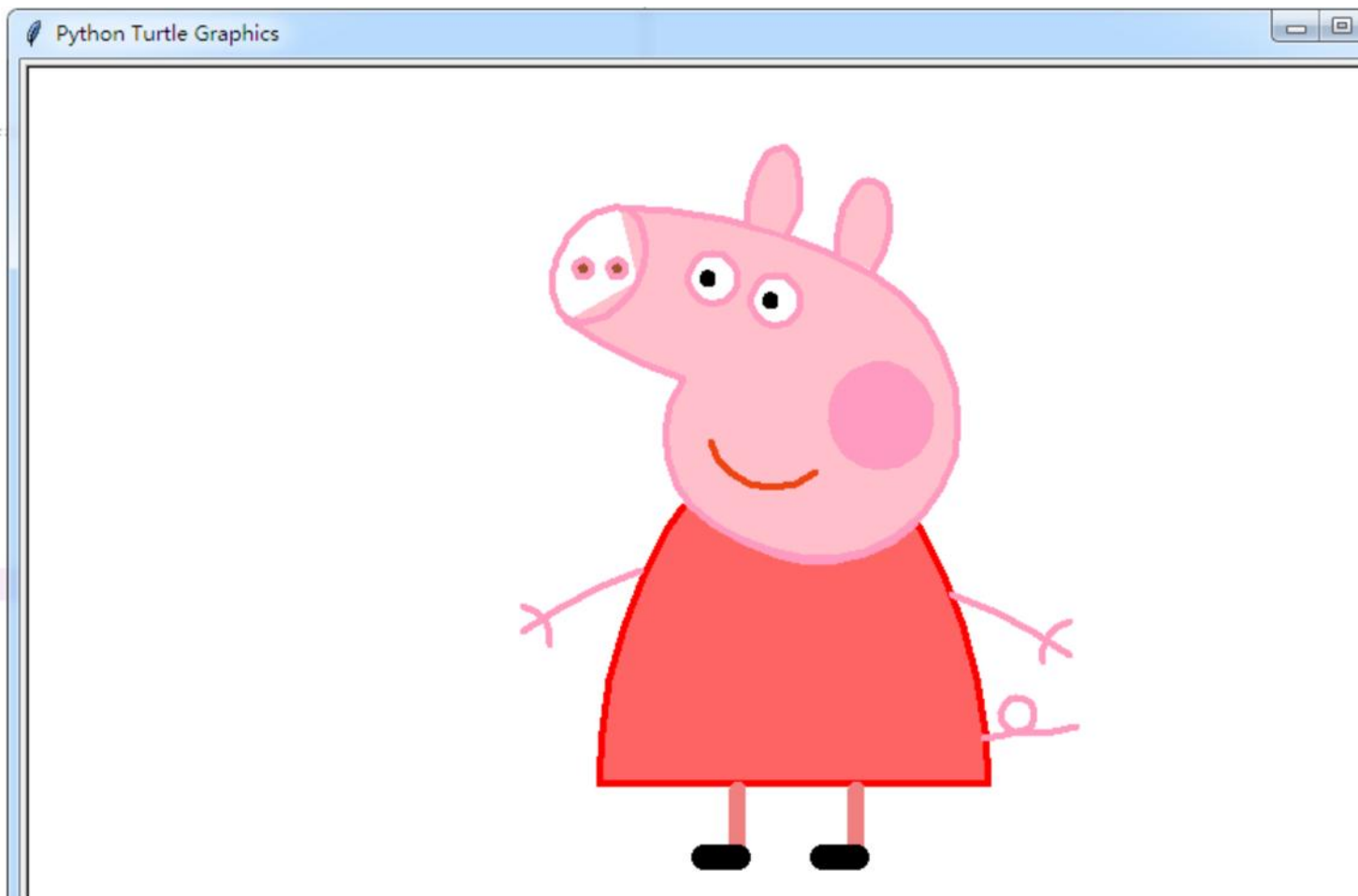
```
1 # coding=utf-8
2 import turtle
3 turtle.setup(650, 350, 200, 200)
4 turtle.penup()
5 turtle.fd(-250)
6 turtle.pendown()
7 turtle.pensize(25)
8 turtle.pencolor("purple")
9 turtle.seth(-40)
10 for i in range(4):
11     turtle.circle(40, 80)
12     turtle.circle(-40, 80)
13 turtle.circle(40, 80/2)
14 turtle.fd(40)
15 turtle.circle(16, 180)
16 turtle.fd(40 * 2/3)
17 turtle.done()
```



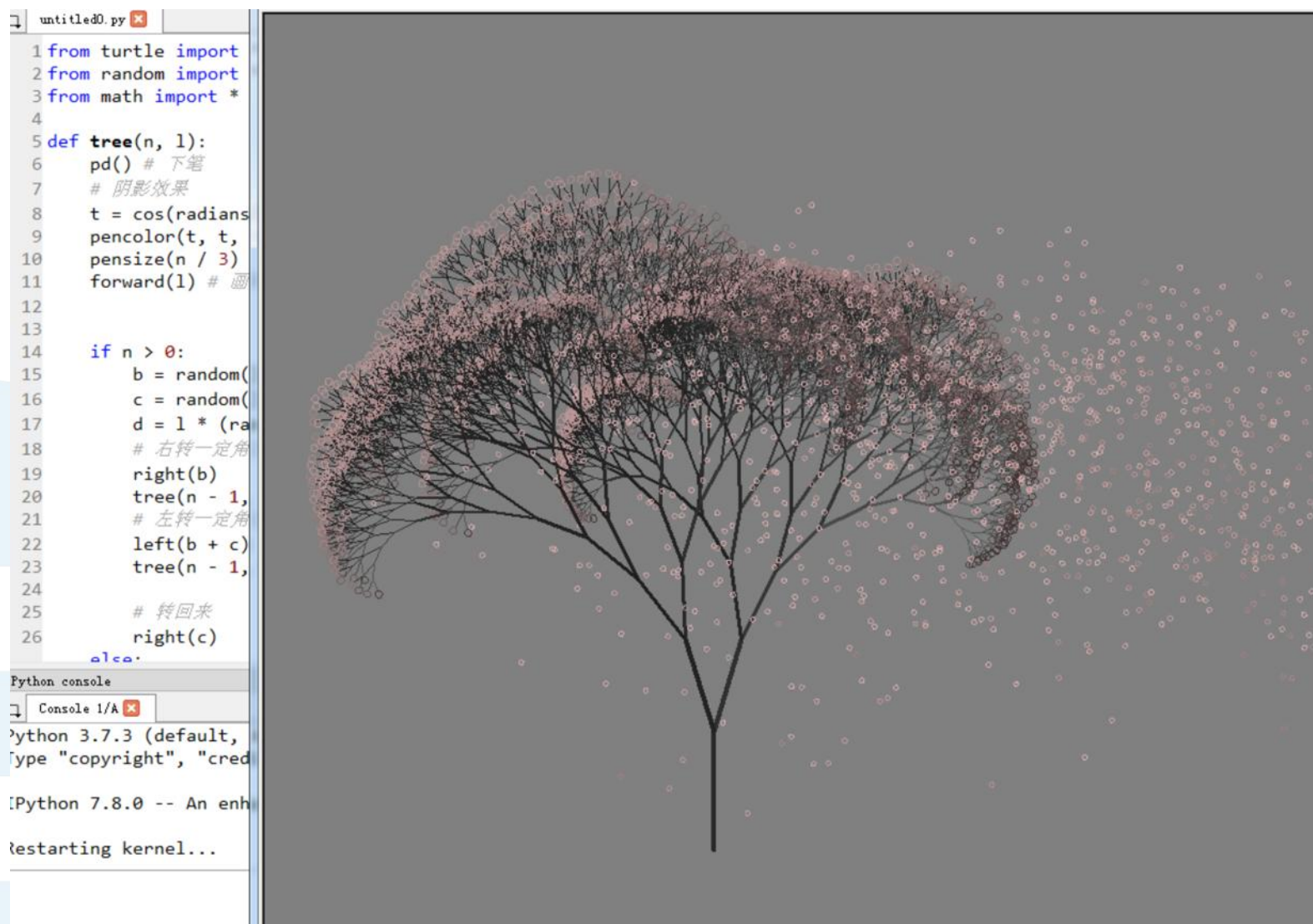
```
import turtle
turtle.setup(650, 350, 200, 200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.pencolor("purple")
turtle.seth(-40)
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
turtle.circle(40, 80/2)
turtle.fd(40)
turtle.circle(16, 180)
turtle.fd(40 * 2/3)
turtle.done()
```

# 实例

```
1 # coding=utf-8
2 import turtle as t
3 # 绘制小猪佩奇
4 # =====
5
6 t.pensize(4)
7 t.hideturtle()
8 t.colormode(255)
9 t.color((255, 155, 192), "pink")
10 t.setup(840, 500)
11 t.speed(10)
12
13 # 鼻子
14 t.pu()
15 t.goto(-100, 100)
16 t.pd()
17 t.seth(-30)
18 t.begin_fill()
19 a = 0.4
20 for i in range(120):
21     if 0 <= i < 30 or 60 <= i < 90:
22         a = a + 0.08
23         t.lt(3) # 向左转3度
24         t.fd(a) # 向前走a的步长
25     else:
26         a = a - 0.08
27         t.lt(3)
```



# 实例





## 二 程序解析





turtle(海龟)库是turtle绘图体系的Python实现

- turtle绘图体系： 1969年诞生，主要用于程序设计入门
  - Python语言的标准库之一
- 入门级的图形绘制函数库

库，或者包，模块，实际上都是一段或几段现成的代码，免除程序员们重复造轮子之苦。

库分为标准库和第三方库，第三方库通常需要另外安装，类似于手机上的自带系统程序和APP的区别。

---

## 三种导库方法

- `import turtle`
- `import turtle as t`
- `from turtle import *`

使用库之前必须先进行导入

`import <库名>`

`<库名>.<函数名>(<函数参数>)`

```
import turtle
turtle.setup(650, 350, 200, 200)
```

每次调用库内的函数都需要重复库名，显得繁琐，因此可以采用别名。

`import <库名> as <库别名>`

`<库别名>.<函数名>(<函数参数>)`

给调用的外部库关联一个更短、更适合自己的名字

```
2 import turtle as t
3 # 绘制小猪佩奇
4 # =====
5
6 t.pensize(4)
```

## 三种导库方法

- `import turtle`
- `import turtle as t`
- `from turtle import *`

还有一种更加简单的方式：

使用`from`和`import`保留字共同完成

`from <库名> import <函数名>`

`from <库名> import *`

`<函数名>(<函数参数>)`

这样就把库内所有函数都放到主程序的命名空间里了，调用的时候无需再提库名，直接函数名即可。

但是，这样有可能造成命名空间的污染，导入多个库的时候，不易发现重名问题。



# 三 Turtle库的基本特点



# Turtle库的基本特点

导库完成后，就可以利用**turtle**库进行绘图了。

**turtle**库是一个自带的标准库，无需额外安装。

**turtle**绘图原理：有一只海龟，其实在窗体正中心，在画布上游，走过的轨迹形成了绘制的图形，海龟由程序控制，可以变换颜色、改变宽度等。

从一个窗体开始

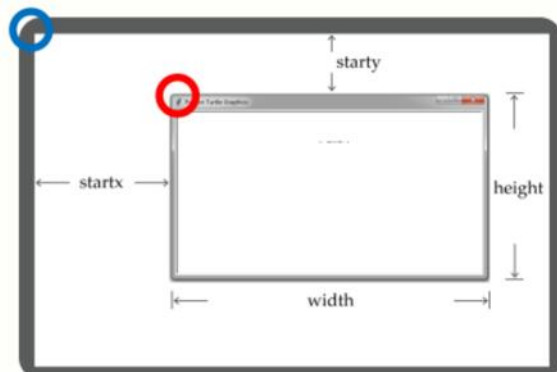


- turtle绘图窗体布局：最小单位是像素
- `setup()`设置窗体大小及位置
- 格式：`turtle.setup(width,height,startx,starty)`。  
4个参数中后两个可选。
- `setup`函数不是必须的，只有当需要控制绘图窗体大小的时候才调用。

# Turtle库的基本特点



`turtle.setup(width, height, startx, starty)`



- `setup()` 设置窗体大小及位置
- 4个参数中后两个可选
- `setup()` 不是必须的

`turtle.setup(800, 400, 0, 0)`



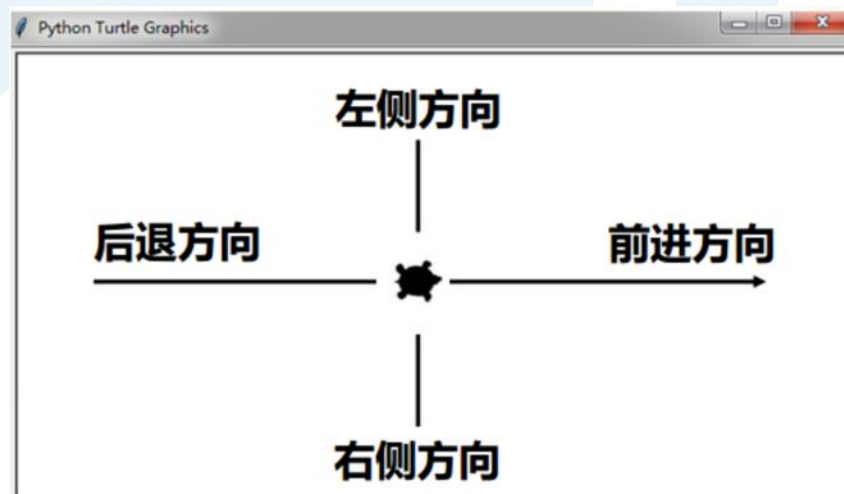
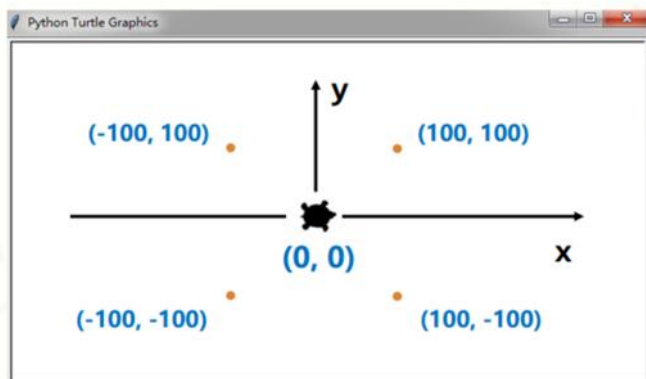
`turtle.setup(800, 400)`



# Turtle库的基本特点

## turtle空间坐标体系

绝对坐标



```
import turtle
```

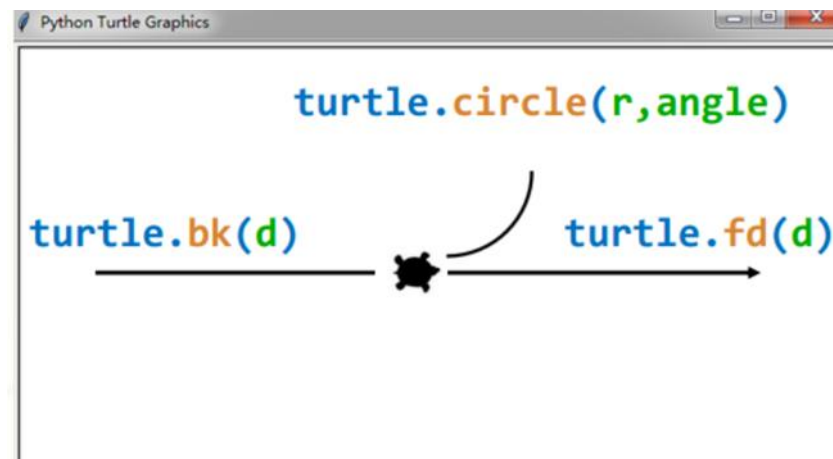
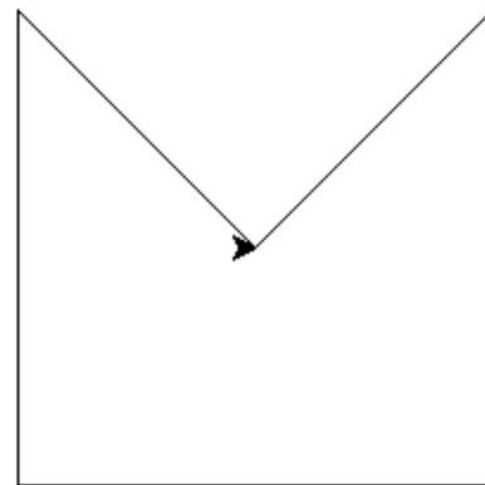
```
turtle.goto( 100, 100)
```

```
turtle.goto( 100,-100)
```

```
turtle.goto(-100,-100)
```

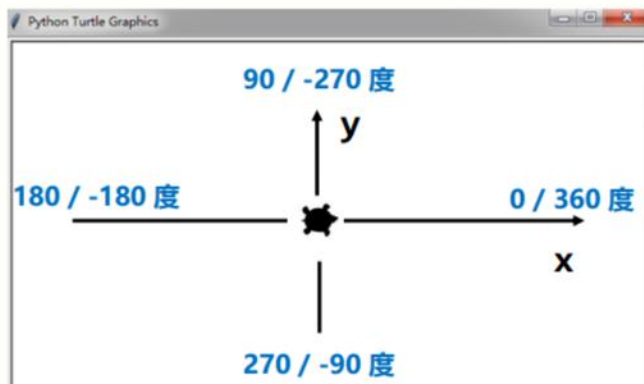
```
turtle.goto(-100, 100)
```

```
turtle.goto(0,0)
```



# Turtle库的基本特点

`turtle.seth(angle)`



- `seth()`改变海龟行进方向
- `angle`为绝对度数
- `seth()`只改变方向但不行进

角度坐标体系

`turtle.seth(45)`



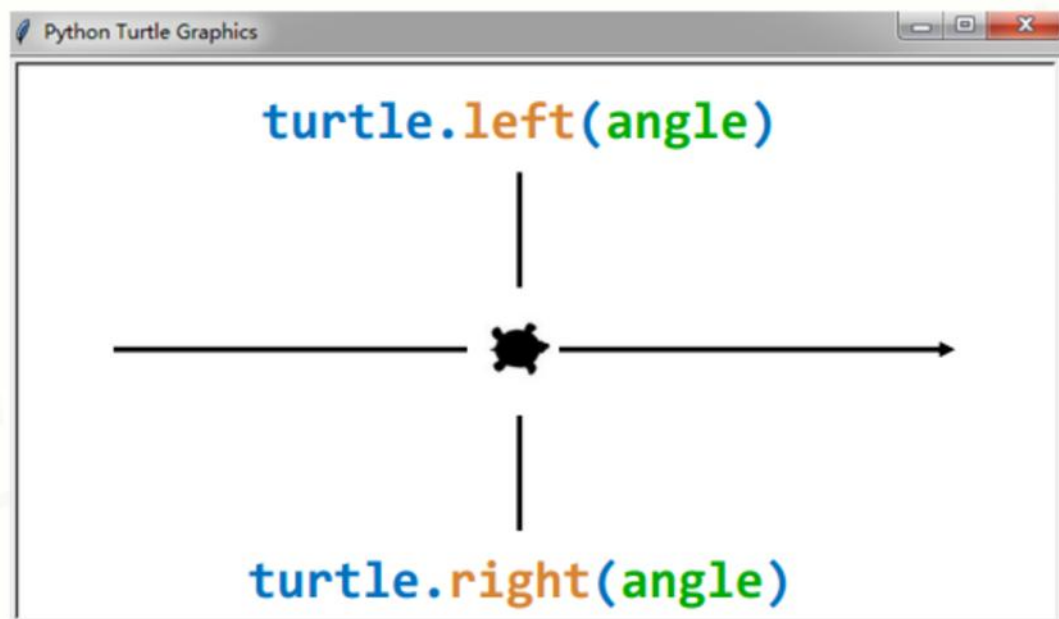
`turtle.seth(-135)`



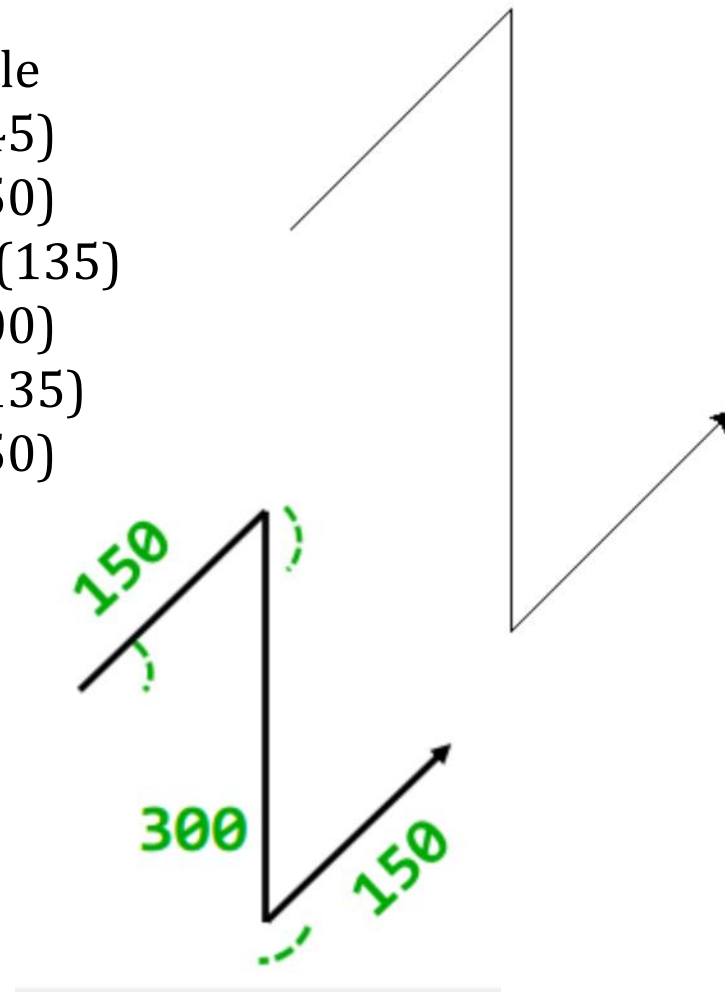


# Turtle库的基本特点

## 角度坐标体系



```
import turtle
turtle.left(45)
turtle.fd(150)
turtle.right(135)
turtle.fd(300)
turtle.left(135)
turtle.fd(150)
```



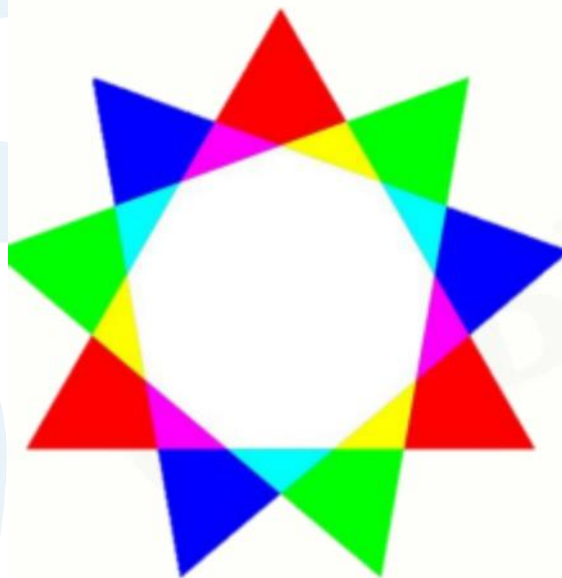
# Turtle库的基本特点

---

## 色彩体系

利用RGB体系给出具体色彩。

RGB指红绿蓝三个通道的颜色组合，可覆盖视力所能感知的所有颜色，RGB每色取值范围0-255整数或0-1小数。



# Turtle库的基本特点

常用RGB色彩

英文名称	RGB整数值	RGB小数值	中文名称
white	255 , 255 , 255	1 , 1 , 1	白色
yellow	255 , 255 , 0	1 , 1 , 0	黄色
magenta	255 , 0 , 255	1 , 0 , 1	洋红
cyan	0 , 255 , 255	0 , 1 , 1	青色
blue	0 , 0 , 255	0 , 0 , 1	蓝色
black	0 , 0 , 0	0 , 0 , 0	黑色

seashell	255 , 245 , 238	1 , 0.96 , 0.93	海贝色
gold	255 , 215 , 0	1 , 0.84 , 0	金色
pink	255 , 192 , 203	1 , 0.75 , 0.80	粉红色
brown	165 , 42 , 42	0.65 , 0.16 , 0.16	棕色
purple	160 , 32 , 240	0.63 , 0.13 , 0.94	紫色
tomato	255 , 99 , 71	1 , 0.39 , 0.28	番茄色

`turtle.colormode(mode)`

- **1.0**: RGB小数值模式

- **255**: RGB整数值模式



## 四 Turtle库的绘图流程



# Turtle库的绘图流程

## 画布的属性

画布就是turtle为我们展开用于绘图区域，我们可以设置它的大小和初始位置。

### ➤ 设置画布大小

`turtle.screensize(canvwidth=None, canvheight=None, bg=None)`，参数分别为画布的宽(单位像素)，高，背景颜色。

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

`turtle.screensize()` #返回默认大小(400, 300)

`turtle.setup(width=0.5, height=0.75, startx=None, starty=None)`，参数：`width, height`: 输入宽和高为整数时，表示像素；为小数时，表示占据电脑屏幕的比例，`(startx, starty)`: 这一坐标表示矩形窗口左上角顶点的位置，如果为空，则窗口位于屏幕中心。

如：`turtle.setup(width=0.6,height=0.6)`

`turtle.setup(width=800,height=800, startx=100, starty=100)`

# Turtle库的绘图流程

## 画笔的属性

画笔(画笔的属性, 颜色、画线的宽度等)

1、turtle.penup() 别名turtle.pu()

画笔抬起, 不留下痕迹

2、turtle.pendown() 别名turtle.pd()

画笔落下, 留下痕迹

3、turtle.pensize(): 设置画笔的宽度;

4、turtle.pencolor(): 没有参数传入, 返回当前画笔颜色, 传入参数设置画笔颜色, 可以是字符串如“green”, “red”, 也可以是RGB 3元组。

例如: turtle.pencolor("purple")颜色字符串

turtle.pencolor(0.63,0.13,0.94)RGB的小数值

turtle.pencolor((0.63,0.13,0.94))RGB的元组值

5、turtle.speed(speed): 设置画笔移动速度, 画笔绘制的速度范围[0,10]整数, 数字越大越快。

# Turtle库的绘图流程

## 绘图命令

命令可以划分为3种：一种为运动命令，一种为画笔控制命令，还有一种是全局控制命令。

(1) 画笔运动命令

命令	说明
<code>turtle.forward(distance)</code>	向当前画笔方向移动distance像素长度
<code>turtle.backward(distance)</code>	向当前画笔相反方向移动distance像素长度
<code>turtle.right(degree)</code>	顺时针移动degree°
<code>turtle.left(degree)</code>	逆时针移动degree°
<code>turtle.pendown()</code>	移动时绘制图形，缺省时也为绘制
<code>turtle.goto(x,y)</code>	将画笔移动到坐标为x,y的位置
<code>turtle.penup()</code>	提起笔移动，不绘制图形，用于另起一个地方绘制
<code>turtle.circle()</code>	画圆，半径为正(负)，表示圆心在画笔的左边(右边)画圆
<code>setx( )</code>	将当前x轴移动到指定位置
<code>sety( )</code>	将当前y轴移动到指定位置
<code>setheading(angle)</code>	设置当前朝向为angle角度
<code>home()</code>	设置当前画笔位置为原点，朝向东。
<code>dot(r)</code>	绘制一个指定直径和颜色的圆点

# Turtle库的绘图流程

## 绘图命令

命令可以划分为3种：一种为运动命令，一种为画笔控制命令，还有一种是全局控制命令。

### (2) 画笔控制命令

命令	说明
<code>turtle.fillcolor(colorstring)</code>	绘制图形的填充颜色
<code>turtle.color(color1, color2)</code>	同时设置pencolor=color1, fillcolor=color2
<code>turtle.filling()</code>	返回当前是否在填充状态
<code>turtle.begin_fill()</code>	准备开始填充图形
<code>turtle.end_fill()</code>	填充完成
<code>turtle.hideturtle()</code>	隐藏画笔的turtle形状
<code>turtle.showturtle()</code>	显示画笔的turtle形状



# Turtle库的绘图流程

## 绘图命令

命令可以划分为3种：一种为运动命令，一种为画笔控制命令，还有一种是全局控制命令。

### (3) 全局控制命令

命令	说明
<code>turtle.clear()</code>	清空turtle窗口，但是turtle的位置和状态不会改变
<code>turtle.reset()</code>	清空窗口，重置turtle状态为起始状态
<code>turtle.undo()</code>	撤销上一个turtle动作
<code>turtle.isvisible()</code>	返回当前turtle是否可见
<code>stamp()</code>	复制当前图形
<code>turtle.write(s [,font=("font-name",font_size,"font_type")])</code>	写文本，s为文本内容，font是字体的参数，分别为字体名称，大小和类型；font为可选项，font参数也是可选项

# Turtle库的绘图流程

## 绘图命令

命令可以划分为3种：一种为运动命令，一种为画笔控制命令，还有一种是全局控制命令。

(4) 其他命令

命令	说明		
turtle.mainloop()或turtle.done()	启动事件循环 -调用Tkinter的mainloop函数。 必须是乌龟图形程序中的最后一个语句。		
turtle.mode(mode=None)	设置乌龟模式（"standard"，"logo"或"world"）并执行重置。如果没有给出模式，则返回当前模式。		
	模式	初始龟标题	正角度
	standard	向右（东）	逆时针
	logo	向上（北）	顺时针
turtle.delay(delay=None)	设置或返回以毫秒为单位的绘图延迟。		
turtle.begin_poly()	开始记录多边形的顶点。当前的乌龟位置是多边形的第一个顶点。		
turtle.end_poly()	停止记录多边形的顶点。当前的乌龟位置是多边形的最后一个顶点。将与第一个顶点相连。		
turtle.get_poly()	返回最后记录的多边形。		

## turtle程序语法元素分析

- 库引用: `import`、`from...import`、`import...as...`
- `penup()`、`pendown()`、`pensize()`、`pencolor()`
- `fd()`、`circle()`、`seth()`
- 循环语句: `for`和`in`、`range()`函数



---

# 五 完整练习



---



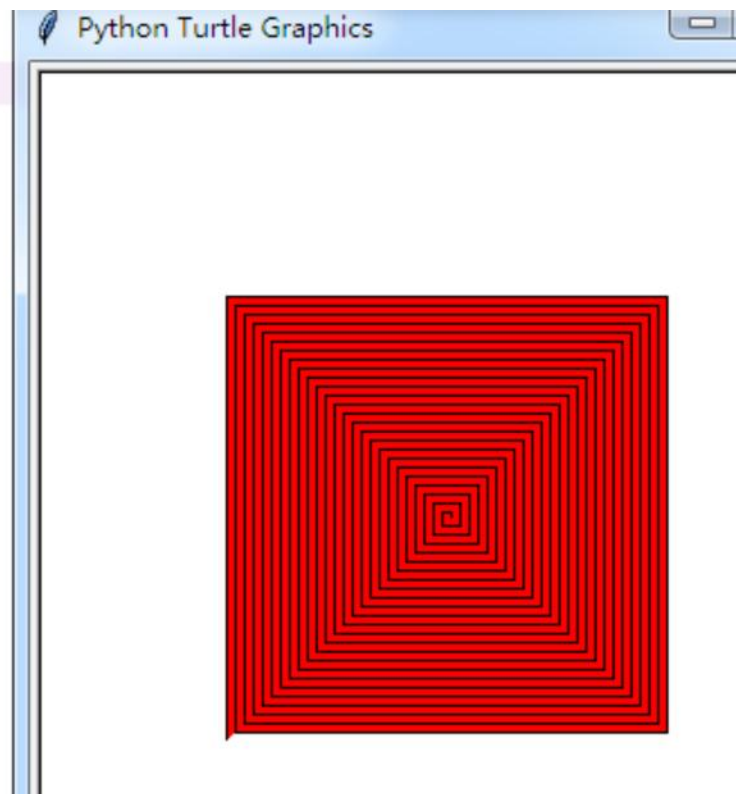
# 完整练习

<https://docs.python.org/3.3/library/turtle.html?highlight=turtle>

<https://docs.python.org/zh-cn/3/library/turtle.html>

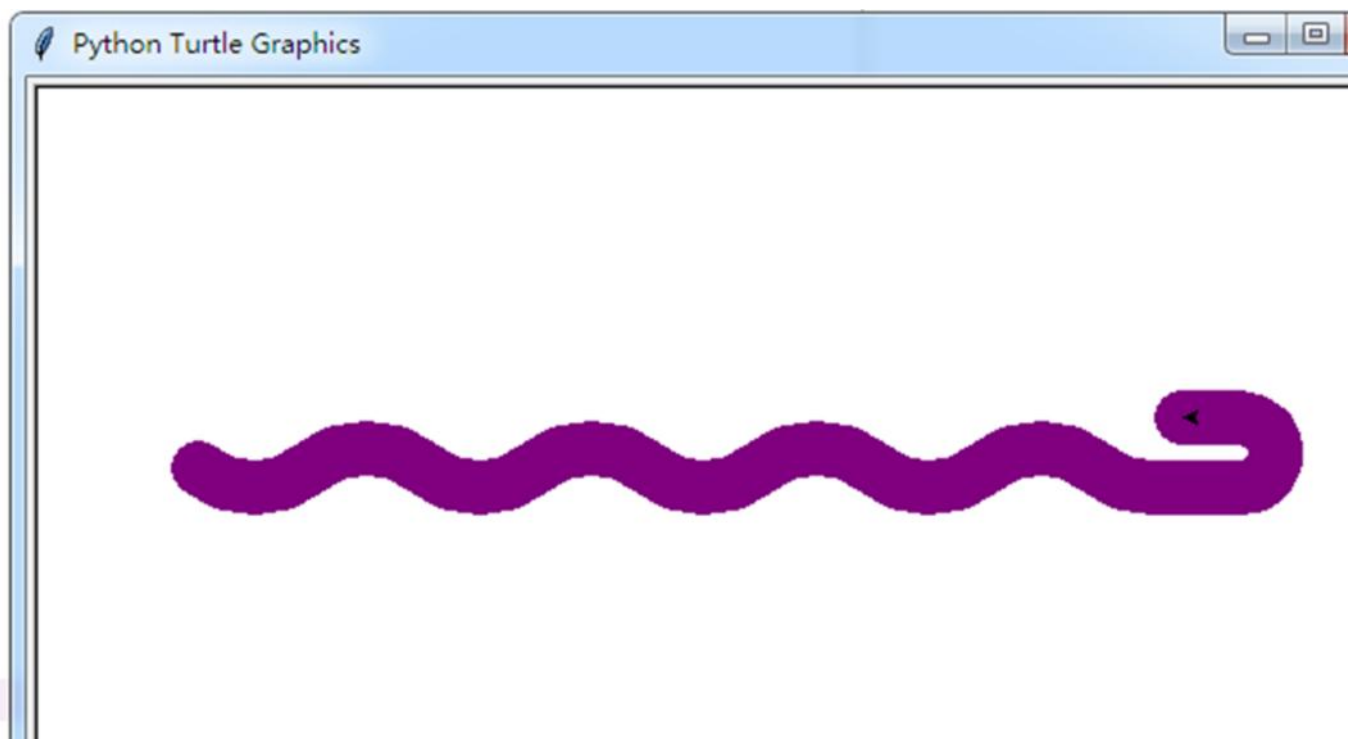
官网的文档总是最好的教材

```
1 import turtle
2 turtle.hideturtle()
3 turtle.color('black', 'red')
4 turtle.speed("fastest")
5 turtle.begin_fill()
6 for x in range(100):
7     turtle.forward(2*x)
8     turtle.left(90)
9 turtle.end_fill()
10 turtle.done()
```



# 完整练习

```
1 # coding=utf-8
2 import turtle
3 turtle.setup(650, 350, 200, 200)
4 turtle.penup()
5 turtle.fd(-250)
6 turtle.pendown()
7 turtle.pensize(25)
8 turtle.pencolor("purple")
9 turtle.seth(-40)
10 for i in range(4):
11     turtle.circle(40, 80)
12     turtle.circle(-40, 80)
13 turtle.circle(40, 80/2)
14 turtle.fd(40)
15 turtle.circle(16, 180)
16 turtle.fd(40 * 2/3)
17 turtle.done()
```





感谢参与 下堂课见

