

# 课前准备

---

1. 回顾第一讲内容
  1. 计算机组成
  2. 自然语言、编程语言
  3. 启动 python shell, 提示符 >>>
  4. print语句
  5. 文件: 文件名、扩展名、路径、斜杠
2. 打字游戏
3. 关键词汇复习

computer	screen
keyboard	mouse
download	install
input	print
hello world	OS : operation system
code / coding	bug
true / True	false / False

## 第二讲 数学运算与乌龟作图

---

### 2.1 基本数学运算

---

运算符

数学运算符	Python 运算符	描述
+	+	加法
-	-	减法
×	*	乘法
÷	/	除法
	%	求余
> >=	> >=	大于、大于等于
< <=	< <=	小于、小于等于
=	==	等于
≠	!=	不等于
()	()	小括号--改变计算顺序

## 编程计算

```
* 参考计算小超市题目在python shell交互提示符 >>> 输入算式，验证python输出计算结构
* 参考+ - * /混合算式
* 参考带括号的混合算式
* 求余数
* 比较大小，True / False
* 判断是否相等，True / False
```

结论： 计算规则跟数学课堂规则一致

## 计算优先级口诀

先乘除、后加减、小括号里最优先

## 测试计算能力

你能快速的计算下列算式的结果嘛？

```
999*999
999*999*999*999
999*999*999*999*999*999*999*999
999*999*999*999*999*999*999*999*999*999*999*999*999

# 启动Python Shell
# 在交互提示符>>>后输入上述算式
```

```
Python 3.6.8 Shell
File Edit Shell Debug Options Window Help
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 999*999
998001
>>> 999*999*999*999
996005996001
>>> 999*999*999*999*999*999*999*999
992027944069944027992001
>>> 999*999*999*999*999*999*999*999*999*999*999*999
988065780494208923208494780065988001
>>>
```

说说你的感想？

复杂的算式，程序比人是不是运算的要快？

## 比较大小

# 在python shell中，检测下述算式的结果：

```
135 > 50
135 < 50
(135 - 35) > 100
(135 - 35) > 100
(135 - 35) == 100
100 == 90
100 == 100
```

## 简易计算器

```
a = int(input("请输入a=")) #输入a值
b = int(input("请输入a=")) #输入b值
print("a+b=", a+b)
print("a-b=", a-b)
print("a*b=", a*b)
print("a/b=", a/b) # 除法，带有小数
print("a%b=", a%b)
print("a//b=", a//b)
```

## 2.2 乌龟作图

### 巩固数学课上角

角的特点：有1个顶点、2条边

**思考：**角的大小跟2条边的长短有没有关系？

直角：90°C

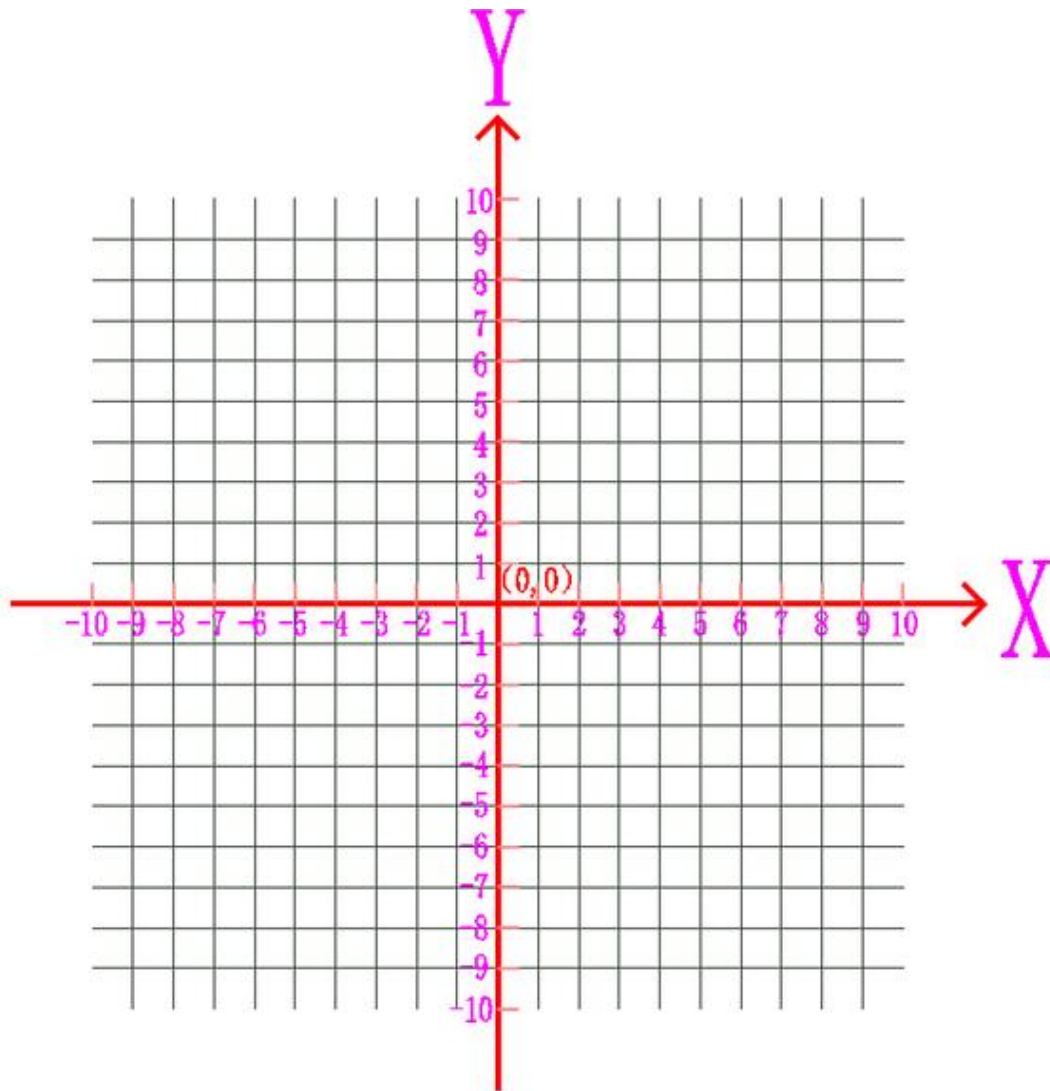
锐角：  $0 < \text{锐角} < 90^\circ$

钝角：  $90^\circ < \text{钝角} < 180^\circ$

角的度数测量：量角器使用--量角器的中心位置和角的顶点重合；角的一边和零刻度线重合，然后看角的另外一边落在量角器的哪个刻度之上，刻度是多少即该角的度数。

## 巩固坐标知识

### 直角坐标系



两条线（X、Y）相互垂直，交叉点称为原点（0，0），即X值为0，Y值为0

通常取向右、向上（箭头方向）为正方向，即X方向，Y方向；往箭头的方向，逐渐增加，相反的方向逐渐减少

### 角度

一个直角坐标系总共有几个直角？

这四个直角加起来正好是一圈，一个直角是90度，那么一圈是多少度？

所以说，面向一个方向，转一圈是360度

**规定：**X箭头方向为0，反方向180/-180；Y箭头方向为90，反方向270/-90

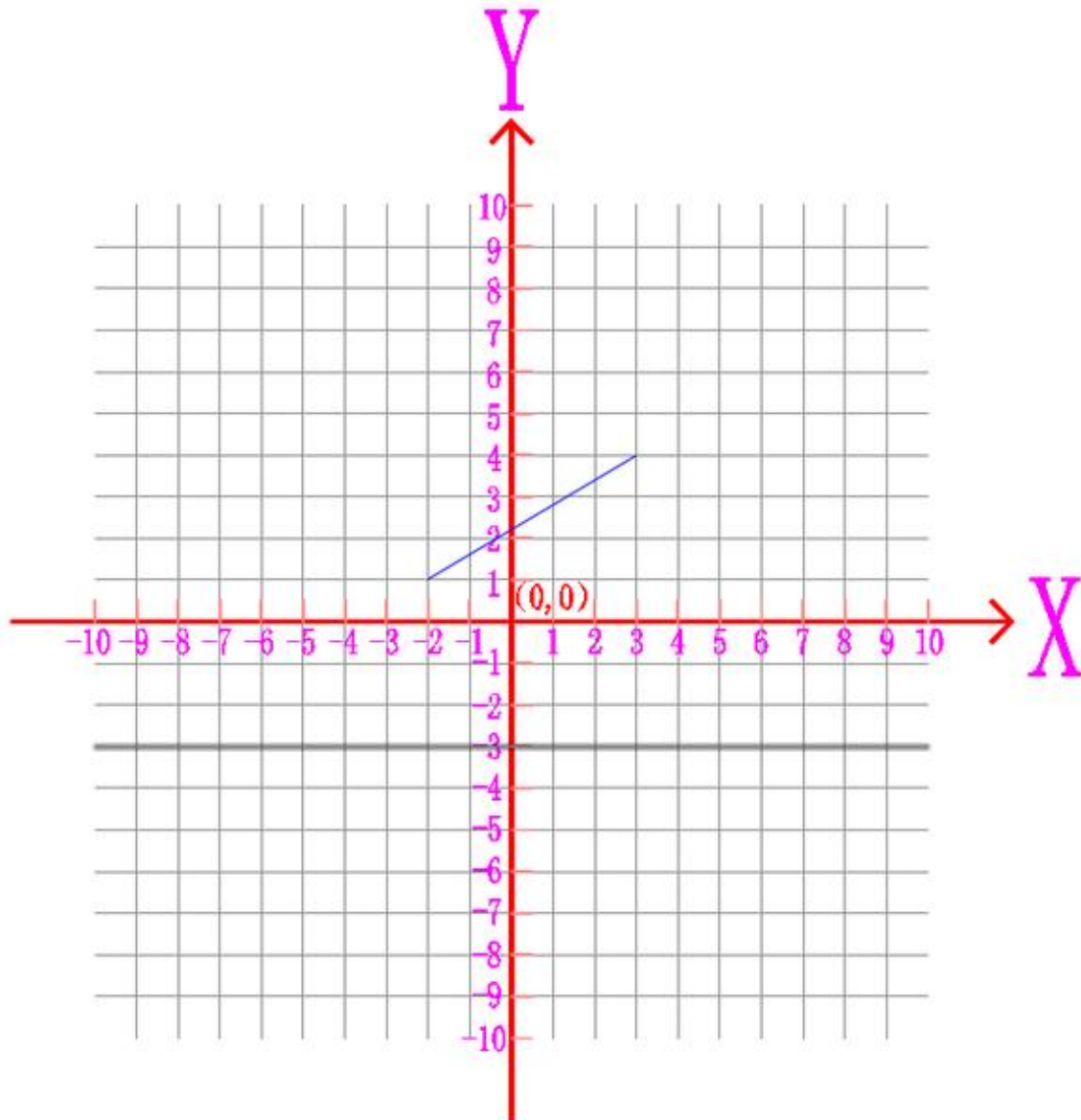
## 点的坐标

在直角坐标系中，有一个点，那么就有X, Y值，那么用  $(X, Y)$  就可以表示它的位置，举例说明

## 线段

线段有两个端点，点在坐标系中X, Y值，那么  $(X_1, Y_1)$  就可以表示一个点， $(X_2, Y_2)$  表示另一个点

比如从  $(-2, 1)$  到  $(3, 4)$  的线段，下图蓝色的线段



## 关键词汇

left	right
forward	back
home	width
up	down
turtle	pen
shape	color

## 乌龟作图

- 在python交互提示符 >>> 逐句执行，观察小乌龟的变化

### 画线

```
import turtle # 导入小乌龟  
  
turtle.Pen() # 创建画笔
```

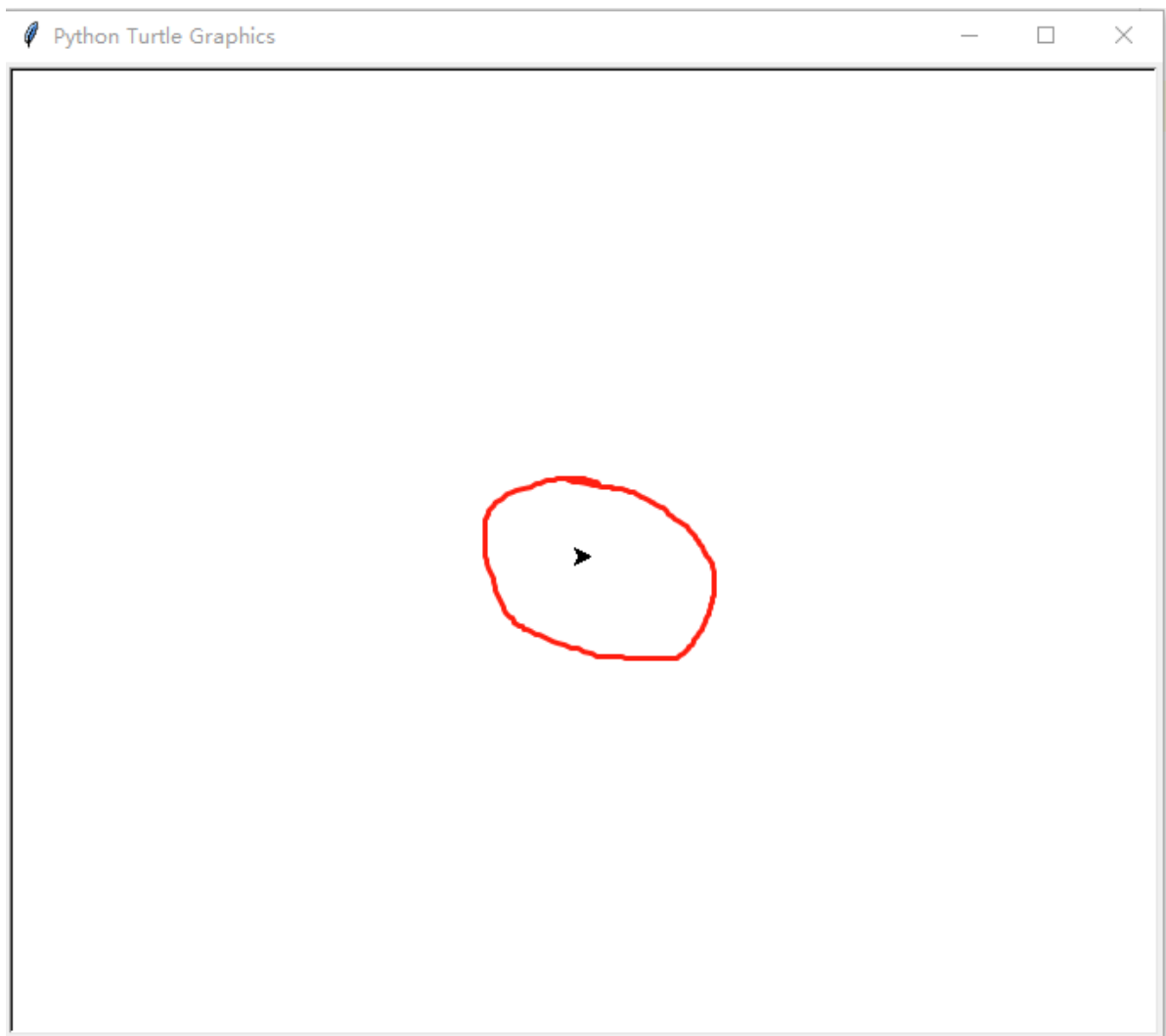
turtle 是python的一个库，里面包含了小乌龟各种操作（函数），比如移动，旋转，设置颜色，画笔粗细等等

A screenshot of a Python 3.6.5 Shell window. The window has a title bar with the text "Python 3.6.5 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains the following text:

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import turtle
>>> turtle.Pen()
<turtle.Turtle object at 0x000002174B7B10B8>
>>>
```

The text is color-coded: "import" is orange, "turtle.Pen()" is blue, and the object representation is blue. The window has a vertical scrollbar on the right side. At the bottom right of the window, the status bar shows "Ln: 6 Col: 4".

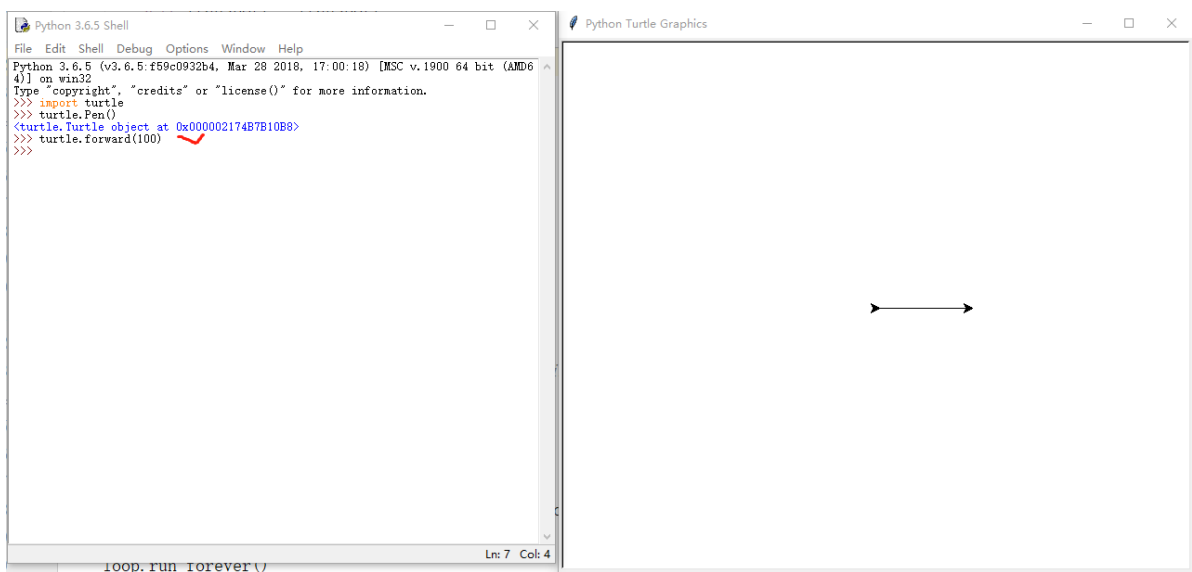
此时乌龟画图界面启动



请观察上图黑色箭头，位置位于界面正中心，即  $(0, 0)$ ；方向朝着直角坐标系X方向，角度为0度

从中心位置  $(0, 0)$ ，朝着X方向（0度方向），前进100

```
turtle.forward(100) # 沿着0度方向，向前走了100
```



逐句执行如下代码，观察乌龟画图界面的变化



```
turtle.left(90) # 左转90度

turtle.forward(100) # 沿着90度方向，向前走了100

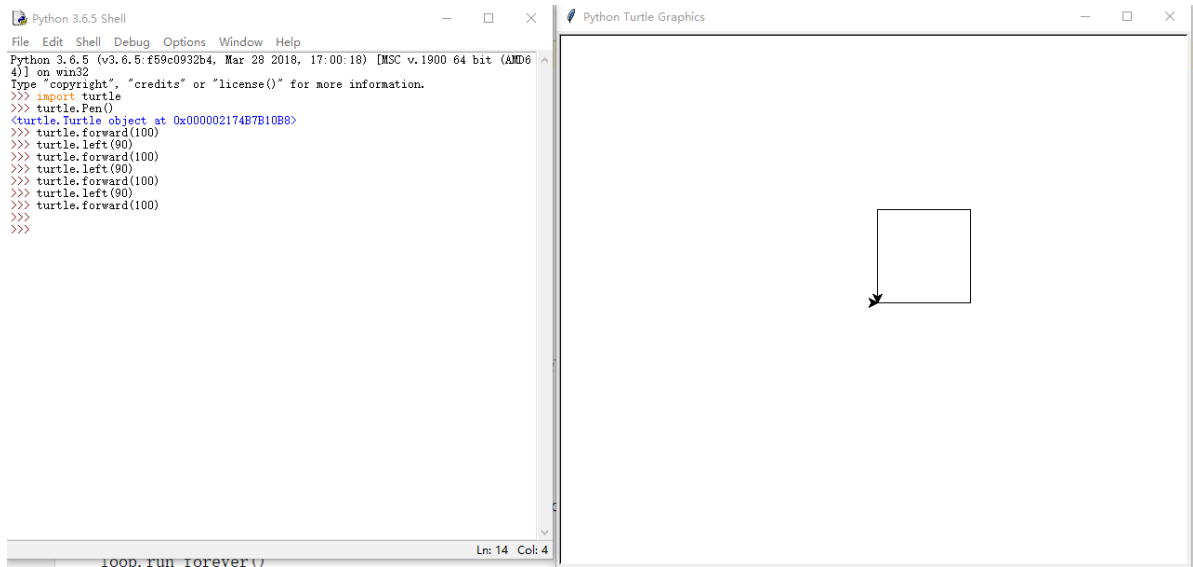
turtle.left(90) # 再左转90度

turtle.forward(100) # 沿着180度方向，向前走了100

turtle.left(90) # 再左转90度

turtle.forward(100) # 沿着270度方向，向前走了100
```

执行完上述python代码，运行效果如下



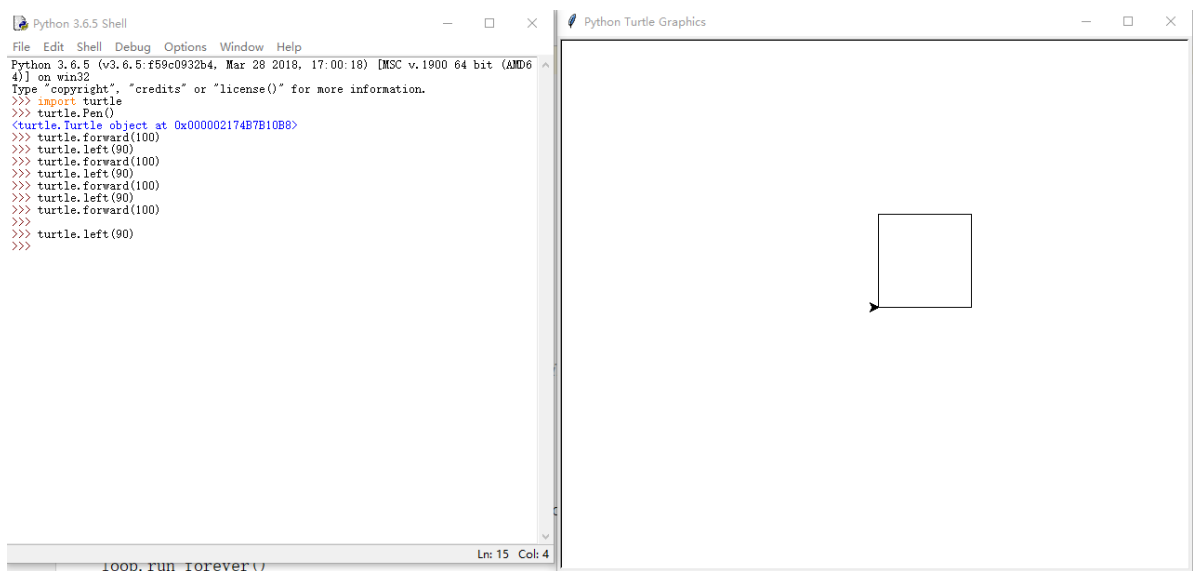
绘制出了一个正方形，箭头回到了中心位置，但是箭头的方向是指向270度，而不是0度

通过代码看一看，总共左转了几个90度，算一算，目前箭头的方向是否正确？

思考，再转多少度，箭头的反向正好是0度？

再左转90度

```
turtle.left(90)
```



此时，转了4次，刚好转了一圈（360度），箭头又指向了X的方向（0度）

改变线的粗细

```
turtle.pensize(5)

turtle.width(5)
```

改变线的颜色

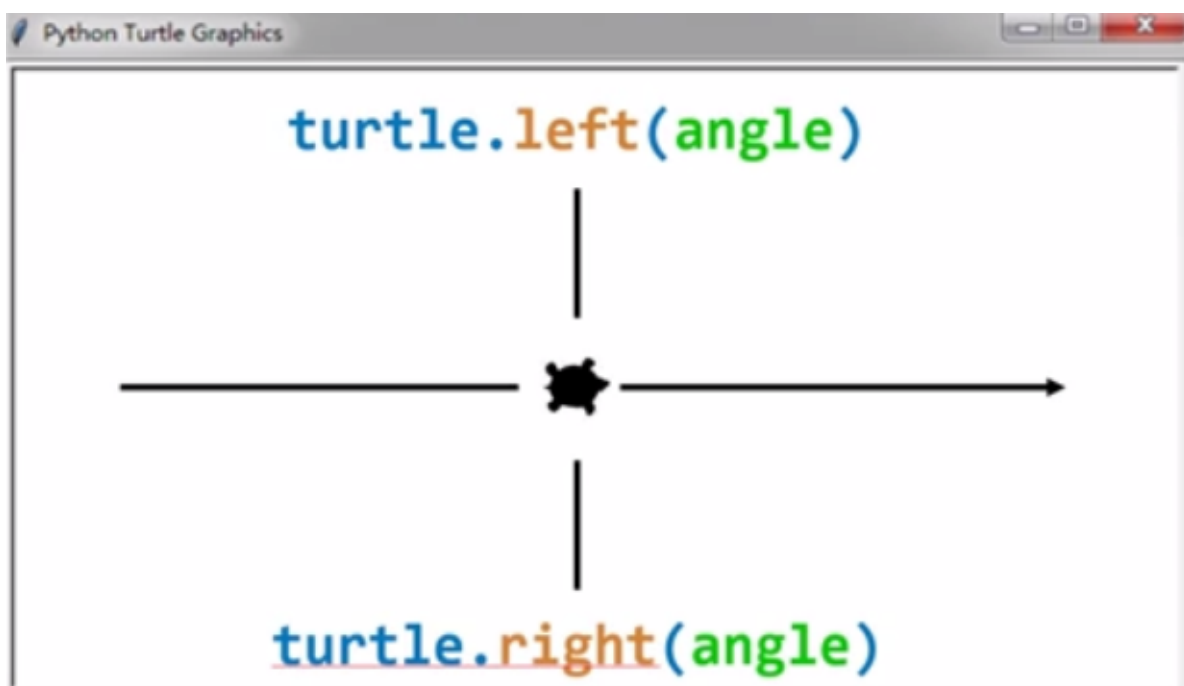
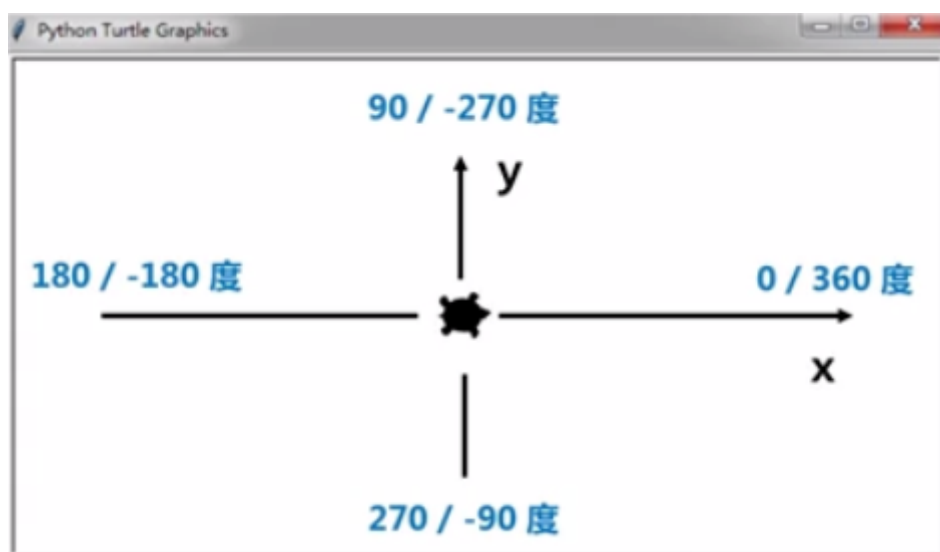
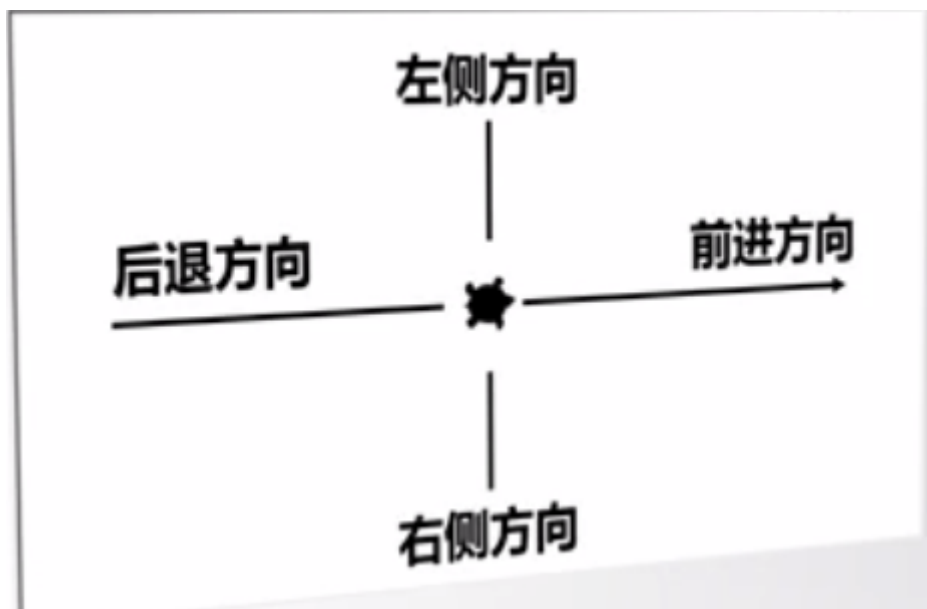
```
turtle.pencolor("red")

red
violet
```

常用颜色名称

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

方向、左转、右转



```
turtle.shape("turtle") # 设置画图小乌龟，默认为箭头
turtle.left(45)
turtle.right(45)
turtle.left(90)
turtle.left(90)
turtle.right(45)
turtle.right(45)
```

## 2.3 编程实践

---

### 画线

- 画线
- 改变线的颜色
- 改变线的粗细

### 画四边形

分析：

- 与正方形区别
  - 正方形四条边一样长，四个角都是直角
  - 长方形四条边，对边的长度一样，四个角都是直角
- 关键代码
  - forward (200) , forward (50) , forward (200) , forward (50)
  - left(90) 或者 right(90)

### 画等边三角形

分析：

- 有三条边，三条边一样长
- 边与边的夹角是60度，需要转 $180-60=120$ 度
- 设置颜色pencolor("red")
- forward(200)
- left(120) 或者 right(120)

### 打字练习

在金山打字通里自定义今天关键词汇，进行指法训练

## 2.4 你学到了什么？

---