海龟作图（一）

和孩子一起学习

Python

我们可以使用python模拟尺规作图。

Python自带名为turtle的模块，它的功能和Logo语言非常相似。

Python中的turtle是画布（canvas）上黑色的三角，它向前/向后移动之后，画布上会留下直线或曲线。通过控制海龟移动，可以绘制简单或复杂的图形。

想召唤海龟我们首先要引入turtle模块。

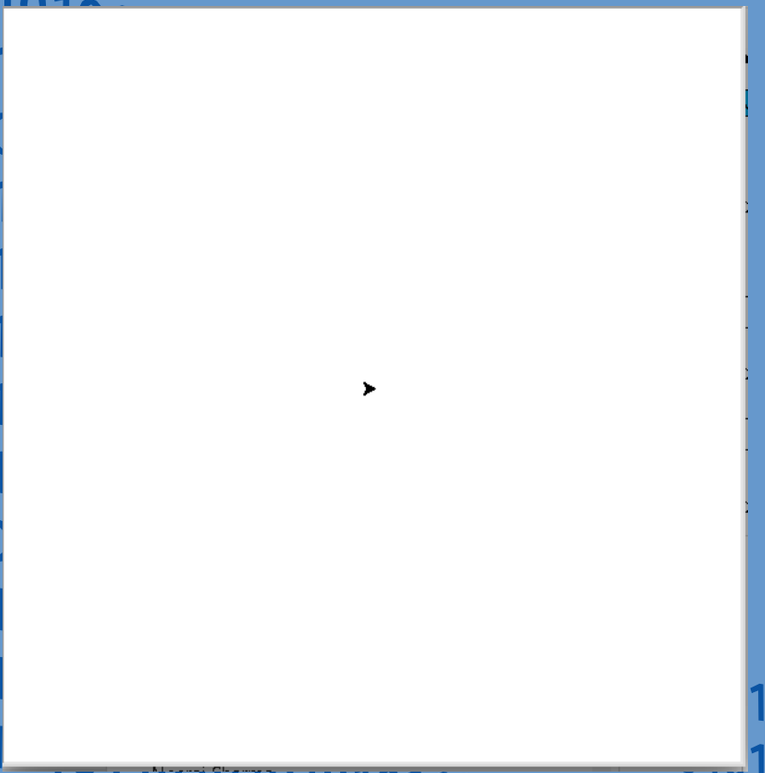
```
>>> import turtle
```

import是python很重要的一个关键字，它为我们的程序加载需要的模块。我们需要的一些功能，已经包含在已经开发好的一些模块中了。我们只要直接去使用它们即可，这样大大简化了我们的开发工作。

譬如，现在我们想要绘图。我们就不必去开发绘图工具，已经有人实现了相应的功能，我们直接拿来用就可以了。

召唤海龟

```
>>> import turtle
>>> t = turtle.Pen()
>>>
```



首先导入turtle模块，接着调用函数turtle.Pen()。注意Pen是从turtle模块导入的函数。将turtle模块给我们的这支笔标记为t，当然你也可使用其他的名称。

第一次调用这个函数时，一块画布（canvas）会被创建出来，海龟（图中黑色的三角，也就是我们用t标注的的第一支画笔）会趴在画布的正中心。

turtle模块采用直角坐标系。画布的中心点为坐标原点（0，0），海龟的初始方向指向x轴的方向。

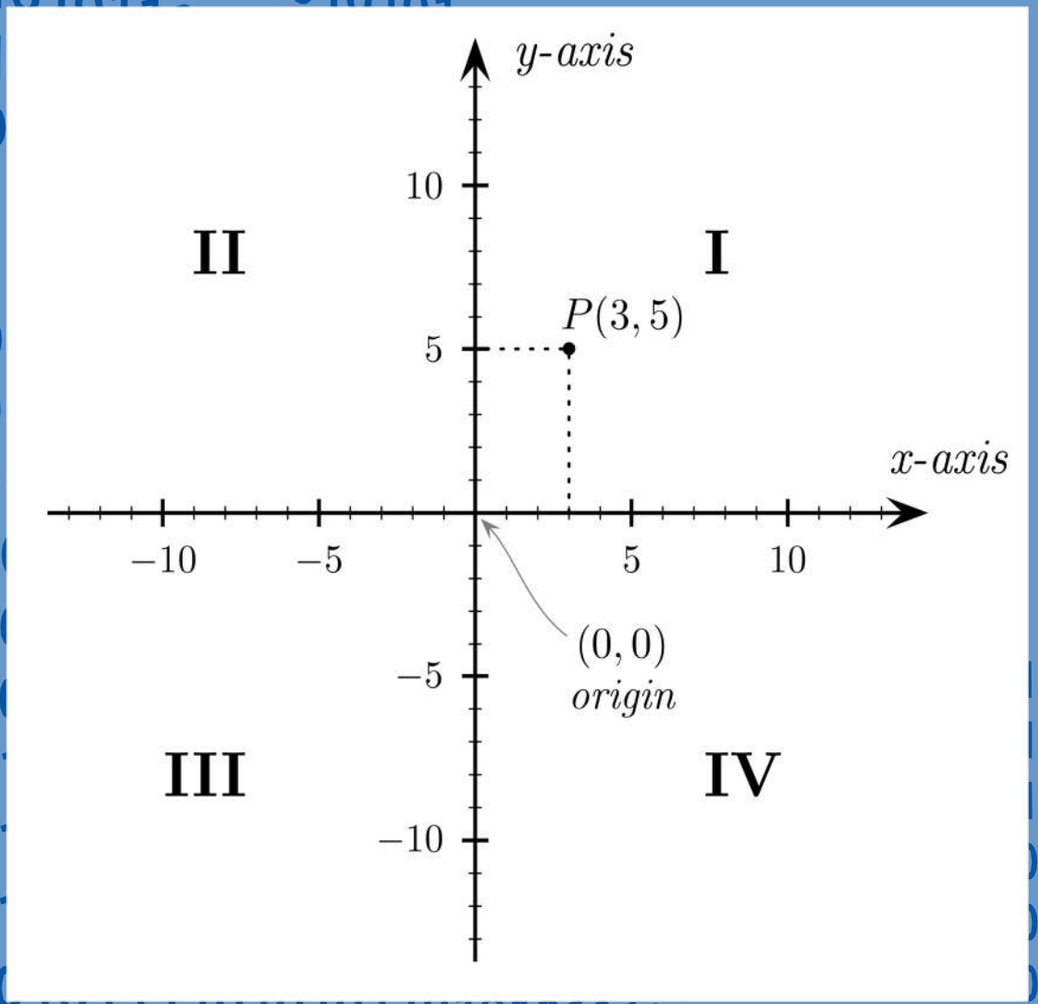
画板与直角坐标系

右图是平面直角坐标系。它由相互垂直的两条直线构成。水平方向是x轴，垂直方向是y轴。图中的每一个点都分别在x、y轴上有对应的数值。坐标系中一个特殊的点是x=0、y=0的点，即原点（Origin）。图中示例的P点x=3、y=5。

坐标轴将平面分为了4个部分，分别命名为第一象限、第二象限、第三象限和第四象限。一般用罗马数字标识。

海龟作图时，海龟初始的位置为图中的原点。海龟的位值信息包括：

- 海龟在坐标系中的坐标。
- 海龟前进方向和x轴的夹角。



常用函数

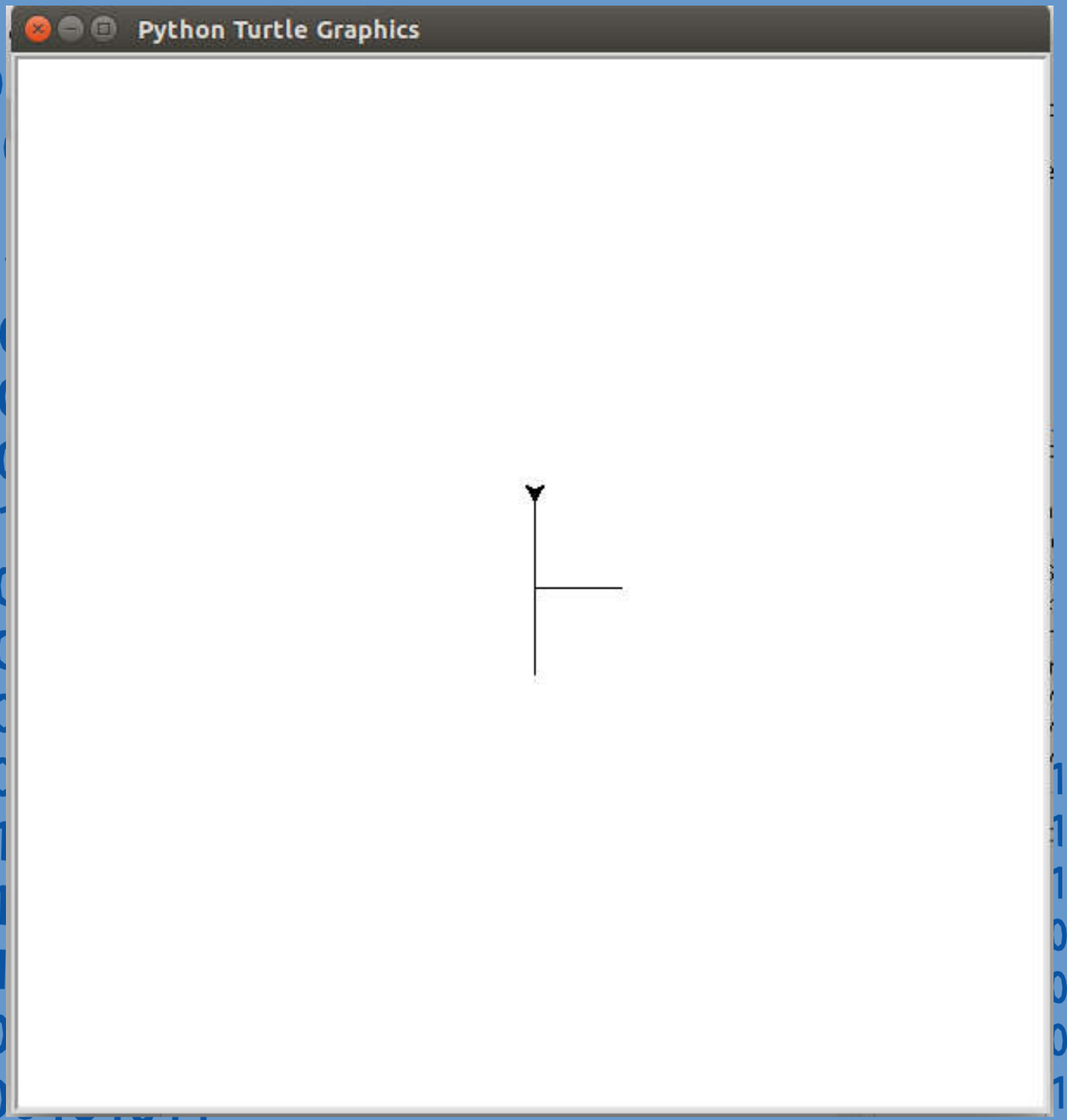
函数	功能
forward() | fd()	海龟前进
backward() | bk() | back()	海龟后退
right() | rt()	海龟右转
left() | lt()	海龟左转
home()	海龟回家
reset()	清空画板，海龟回家 home+clear
circle()	海龟画圆
position() | pos()	设置海龟当前坐标位置
penup() | pu() | up()	抬起画笔，此后海龟移动，将不会在画布上留下痕迹
pendown() | pd() | down()	放下画笔，此后海龟将在画布上留下移动痕迹
pencolor()	设置画笔颜色
fillcolor()	设置填充颜色
begin_fill()	开始填充颜色
end_fill()	停止填充颜色
showturtle() | st()	显示海龟
hideturtle() | ht()	隐藏海龟

移动海龟

```
>>> t.forward(50)
>>> t.back(50)
>>> t.right(90)
>>> t.forward(50)
>>> t.back(50)
>>> t.back(50)
```

右图为执行完以上命令序列后，画笔上的图形。

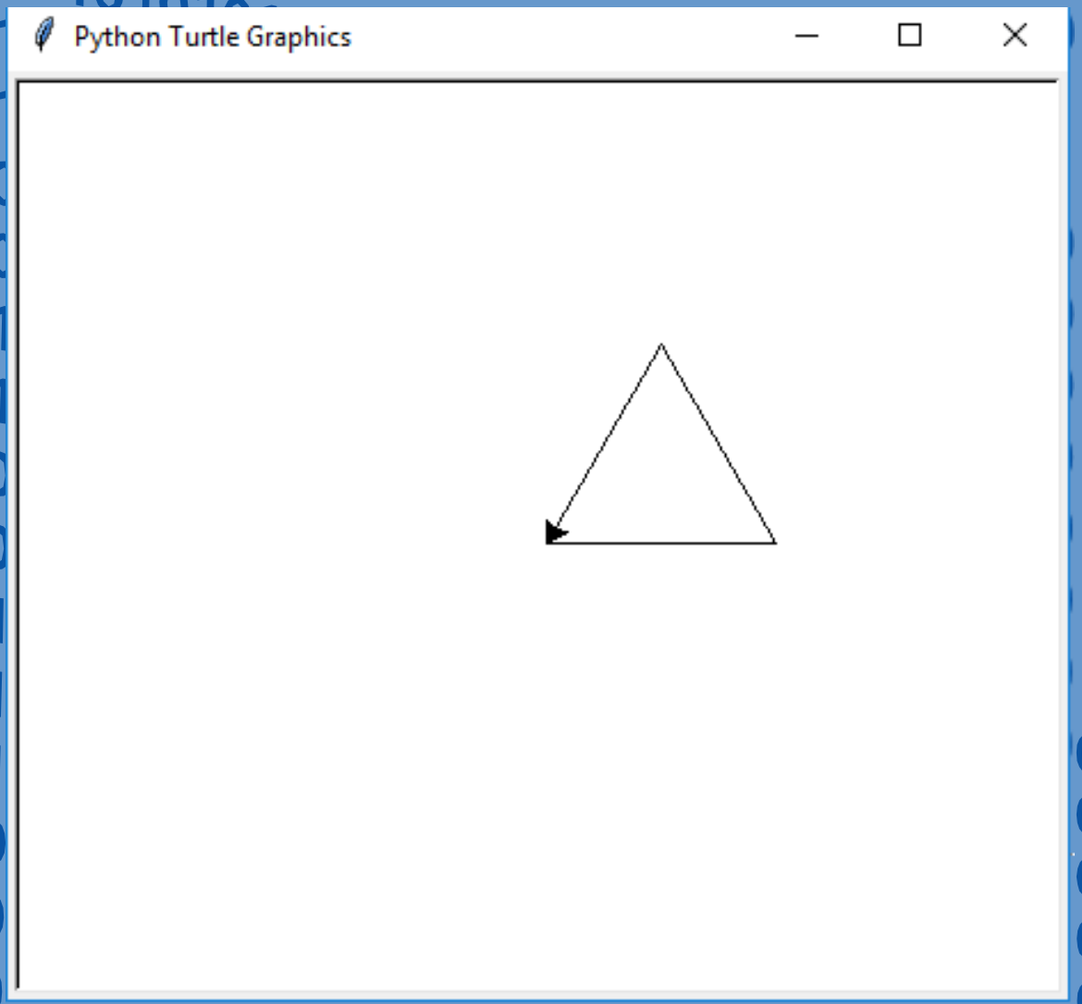
对于画笔移动函数，其输入值的单位是像素。第一条语句让海龟前进50个像素点；第二条语句让海龟后退50个像素点；第三条语句让海龟右转90°；第四条语句让海龟前进50个像素点；后两条语句让海龟总共后退100个像素点。



绘制三角形

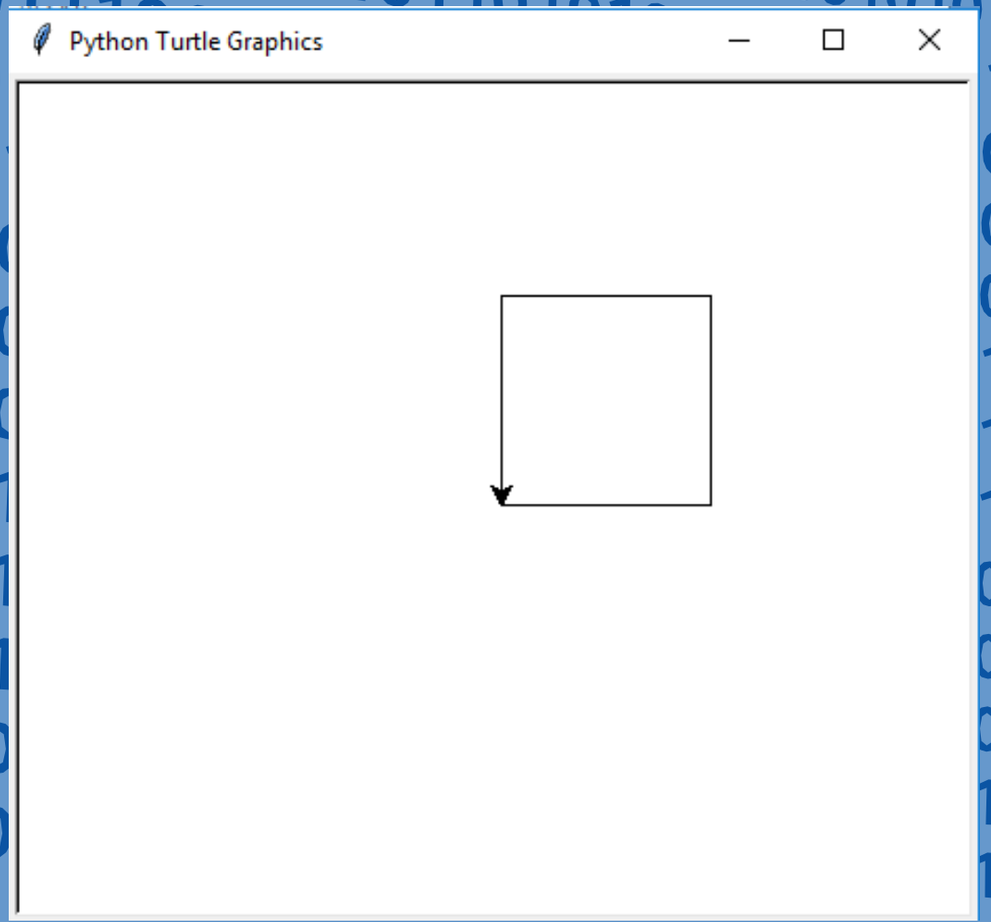
```
>>> import turtle
>>> t = turtle.Pen()
>>> t.forward(100)
>>> t.left(120)
>>> t.forward(100)
>>> t.left(120)
>>> t.forward(100)
```

以上代码绘制一个正三角形（等边三角形）

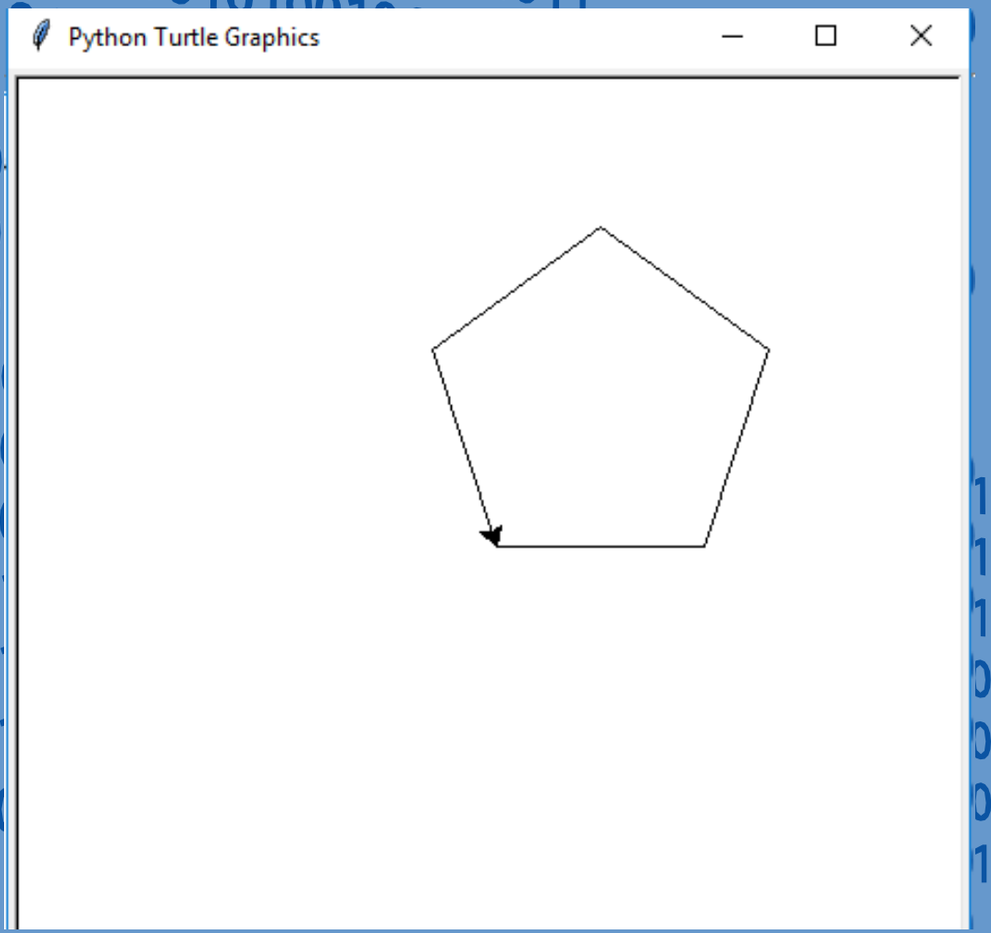


绘制正多边形

正四边形



正五边形



请同学们思考如何绘制以上的正多边形？

使用循环

第一辑中，我们介绍了循环语句。绘制多边形的过程，前进以及转弯的需要多次重复。重复的次数由多边形边数决定。重复的操作均可以有循环语句实现。

```
import turtle
t = turtle.Pen()

for i in range(3):
    t.forward(50)
    t.left(120)
```

以上便是由循环语句实现的三角形。

比较异同

```
import turtle
t = turtle.Pen()
for i in range(4):
    t.forward(50)
    t.left(360/4)
```

```
import turtle
t = turtle.Pen()
for i in range(5):
    t.forward(50)
    t.left(360/5)
```

上面分别是绘制正四边形和正五边形的代码。left()输入的角度是正多边形外角的度数（360/sides）。和前面正三边型一起做比较，我们发现这几个程序唯一的不同就是多边形的边数。那么，我们能否做一个可以绘制任意边数多边形的程序呢？

绘制多边形

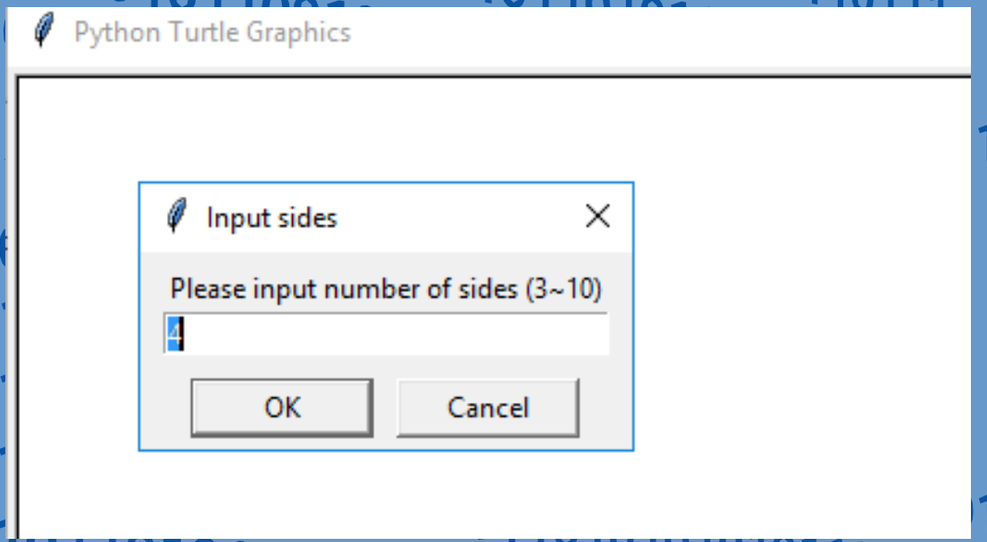
```
import turtle

t = turtle.Pen()
sides = int(turtle.numinput("Input sides",
                "Please input number of sides (3~10)",
                4, 3, 10))
for i in range(sides):
    t.forward(50)
    t.left(360/sides)
```

新的程序中，多边形的边数被提取出来由sides表示，这样程序对所有的正多边形都一样了。但是这里有了新问题：sides的值如何确定？

前面的课程，我介绍了input()函数可以被用来接收用户的输入。这里，我使用了一个新的函数turtle.numinput()。这个函数弹出一个窗口，让用户输入sides的数值。函数有5个入参：第一个为弹出窗口的名称；第二个为提示语句，这里提示了用户程序所接受的多边形边数（3~10），当用户输入不在这个范围内，程序会提示输入错误。第三个参数是边数默认值；第四个是边数允许的最小值；第五个是边数允许的最大值。

弹出窗口

函数turtle.numinput()产生的窗口如右图。请对照代码，看看前三个参数在该窗口出现的位置。并请输入不同的数值，看看你的输入是否被程序接受。



保存文件

Python程序可以直接在shell中运行。但是这样做，每次运行都要把程序重新录入一次，而且还不利于程序的分享。

如果我们将文件保存在文件中，不仅可以方便我们自己再次使用，而且还有利于分享。

IDLE自带的文本编辑器很适合初学者，因为它具有适合Python的语法高亮和自动缩进功能。

点击shell窗口【file】->【New File】即可打开新的文本编辑窗口。我们可以在文本编辑器中录入代码，然后运行。运行的方式可以是【Run】->【Run Module】，也可以直接按键盘上的功能键F5。对于改动后未曾的文件，系统会提示你保存文件。保存文件时，请仔细选择文件存放的目录，以便下次使用。

小结

本讲，我们介绍了以下内容：

1. 如何使用turtle模块简单作图
2. 如何使用循环语句
3. 如何提取不同对象的异同，建立一个共同的模型
4. 如何保存文件

如果您有任何问题或者建议，请发邮件至如下地址

samuelzhang77@yahoo.com