

# Python艺术编程04

## ——基本数据类型和数学曲线

北京大学 陈斌

2018.09.20

# Python语言的几个要件

## 数据对象和组织

- 对现实世界实体和概念的抽象
- 分为简单类型和容器类型
- 简单类型用来表示值
  - 整数int、浮点数float、复数complex、逻辑值bool、字符串str
- 容器类型用来组织这些值
  - 列表list、元组tuple、集合set、字典dict
- 数据类型之间几乎都可以转换

## 赋值和控制流

- 对现实世界处理和过程的抽象
- 分为运算语句和控制流语句
- 运算语句用来实现处理与暂存
  - 表达式计算、函数调用、赋值
- 控制流语句用来组织语句描述过程
  - 顺序、条件分支、循环
- 定义语句也用来组织语句，描述一个包含一系列处理过程的计算单元
  - 函数定义、类定义

# Python数据类型：整数int、浮点数float

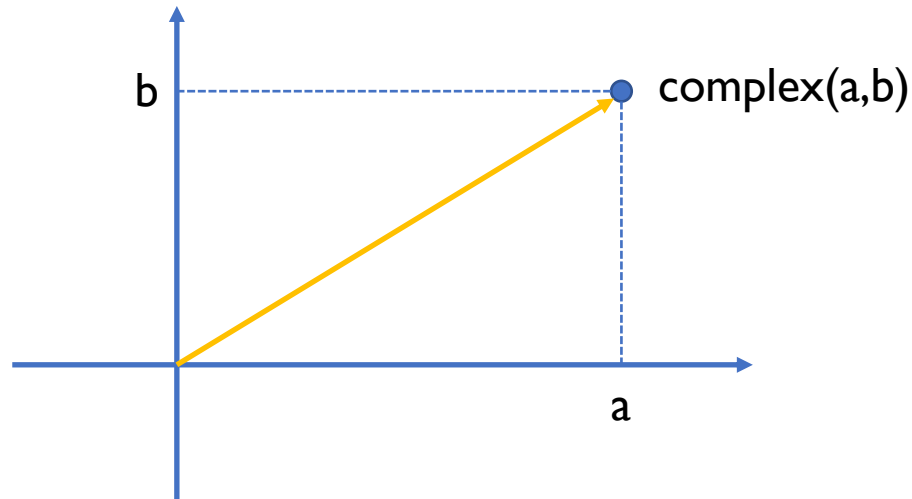
- 最大的特点是不限制大小
- 浮点数受到17位有效数字的限制
- 常见的运算包括加、减、乘、除、整除、求余、幂指数等
- 浮点数的操作也差不多
- 一些常用的数学函数如sqrt/sin/cos等都在math模块中
  - import math
  - math.sqrt(2)

```
>>> 5
5
>>> -100
-100
>>> 5 + 8
13
>>> 90 - 10
80
>>> 4 * 7
28
>>> 7 / 2
3.5
>>> 7 // 2
3
>>> 7 % 3
1
>>> 3 ** 4
81
>>> 2 ** 100
1267650600228229401496703205376
>>> divmod(9, 5)
(1, 4)
>>> |
```



# Python数据类型：复数

- Python内置对复数的支持
  - `complex(a,b)`
- 支持所有常见的复数计算
  - 求距离：`abs`函数
- 对复数处理的数学函数在模块`cmath`中
  - `import cmath`
  - `cmath.sqrt(1+2j)`



```
>>> 1+3j
(1+3j)
>>> (1+2j)*(2+3j)
(-4+7j)
>>> (1+2j)/(2+3j)
(0.6153846153846154+0.07692307692307691j)
>>> (1+2j)**2
(-3+4j)
>>> (1+2j).imag
2.0
>>> (1+2j).real
1.0
>>>
```

# Python数据类型：逻辑值

- 逻辑值仅包括True/False两个
- 用来配合if/while等语句做条件判断
- 其它数据类型可以转换为逻辑值：
  - 例如数值：0与非0等

```
>>> True
True
>>> False
False
>>> 1>2
False
>>> 23<=34
True
>>> bool(0)
False
>>> bool(999)
True
>>> if (2>1):
        print ("OK")

OK
>>> |
```

# Python数据类型：字符串

- 用双引号或者单引号都可以表示字符串
- 多行字符串用三个连续单引号表示
- 特殊字符用转义符号“\”表示
  - 制表符\t，换行符号\n
- 字符串操作：
  - +连接、\*复制、len长度
  - [start:end:step]用来提取一部分

```
>>> 'abc'
'abc'
>>> "abc"
'abc'
>>> '''abc def
ghi jk'''
'abc def\nghi jk'
>>> "Hello\nWorld!"
'Hello\nWorld!'
>>> print ("Hello\nWorld!")
Hello
World!
>>> 'abc' + 'def'
'abcdef'|
>>> 'abc' * 4
'ab cab cab cab'
>>> len('abc')
3
>>> 'abcd'[0:2]
'ab'
>>> 'abcd'[0::2]
'ac'
```

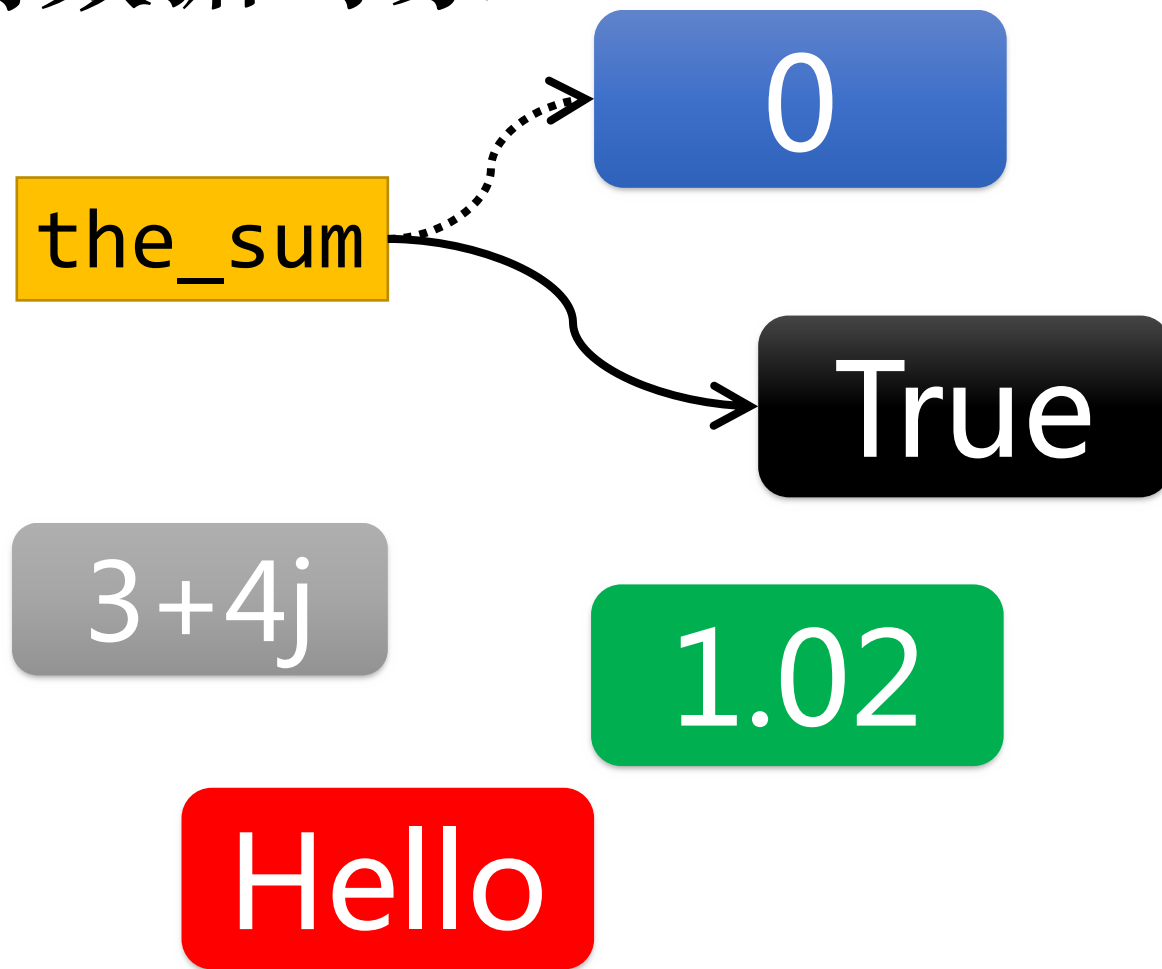
# Python数据类型：字符串

- 一些高级操作：
  - split: 分割; join: 合并
  - upper/lower/swapcase: 大小写相关
  - ljust/center/rjust: 排版左中右对齐
  - replace: 替换子串

```
>>> 'You are my sunshine.'.split(' ')
['You', 'are', 'my', 'sunshine.']
>>> '-'.join(["One", "for", "Two"])
'One-for-Two'
>>> 'abc'.upper()
'ABC'
>>> 'aBC'.lower()
'abc'
>>> 'Abc'.swapcase()
'aBC'
>>> 'Hello World!'.center(20)
'    Hello World!    '
>>> 'Tom smiled, Tom cried, Tom shouted'.replace('Tom', 'Jane')
'Jane smiled, Jane cried, Jane shouted'
```

# Python变量机制： 引用数据对象

- 赋值语句`the_sum = 0`，实际上是创建了名为`the_sum`的变量，然后指向数据对象“0”
- 所以变量可以随时指向任何一个数据对象，比如`True`，`1.02`，或者`"Hello"`
- 变量的类型随着指向的数据对象类型改变而改变！





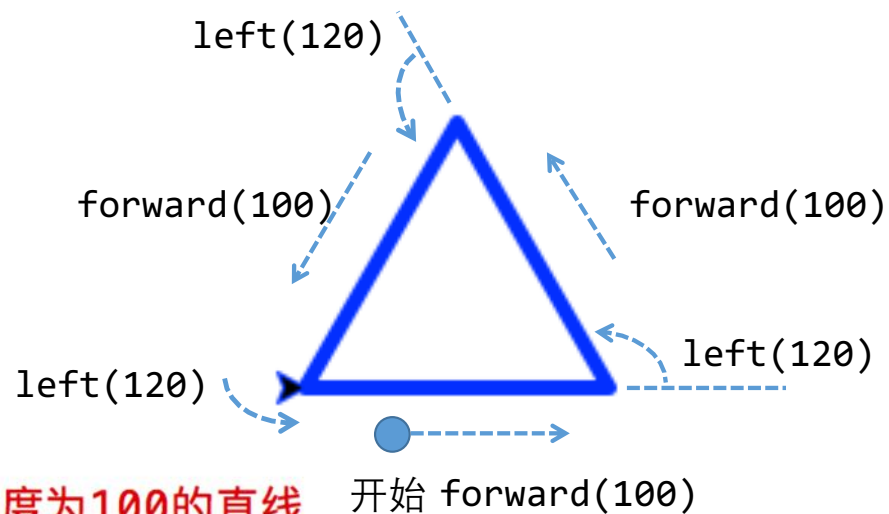
# 上机练习：基本数据类型

- 数值基本运算：33和7
  - +, -, \*, /, //, %, \*\*
  - 十六进制hex(), oct(), bin()
- 类型转换
  - 1, 0, 'abc', None, 1.2, False, ''
  - str(), bool(), int(), float()
  - is None, ==, !=
- 字符串基本操作
  - +, \*, len(), [], in
  - ord(), chr()
  - 含有中文的字符串
- 字符串高级操作
  - s='abcdefg12345'
  - 切片：获得defg12, 获得fg12345, 获得54321, 获得aceg2
  - t='Mike and Tom'
  - split拆分、
  - upper/lower/swapcase修改大小写、
  - ljust/center/rjust排版30位宽度左中右对齐
  - replace将Mike替换为Jerry

# 海龟做图：turtle

- 模拟海龟在沙滩上爬行所描绘的轨迹，从LOGO语言借鉴而来
- 海龟：显隐、外形、是否动画
- 画笔：抬起落下、颜色、粗细

```
import turtle # 导入turtle模块
p = turtle.Pen() # 创建一支画笔（海龟）
p.pencolor('blue') # 设置画笔颜色为蓝色
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
```



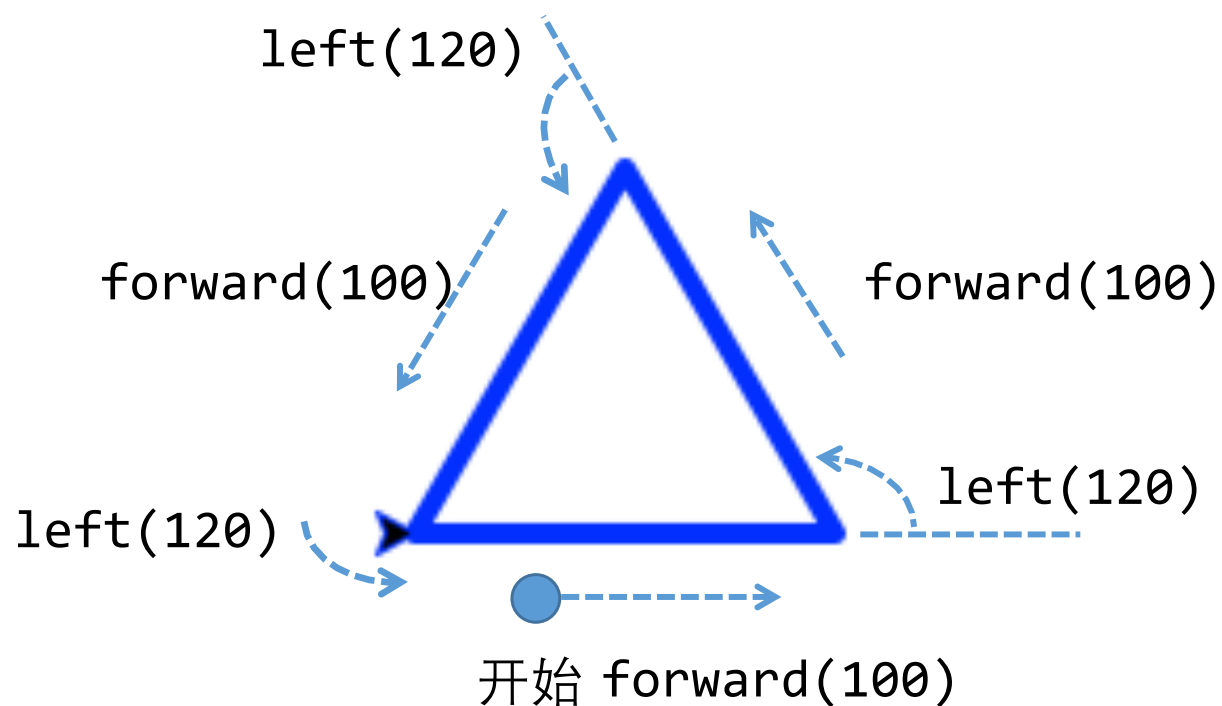
# 重复：简单的循环语句for

```
1 print("range:5")
2 for i in range(5):
3     print(i)
4
5 print("range:1,5,2")
6 for i in range(1, 5, 2):
7     print(i)
8
9 print("range:4,1,-1")
10 for n in range(4, 1, -1):
11     print(n)
```

```
range:5
0
1
2
3
4
range:1,5,2
1
3
range:4,1,-1
4
3
2
```

# 循环语句的应用


- 如何用循环语句画出等边三角形?





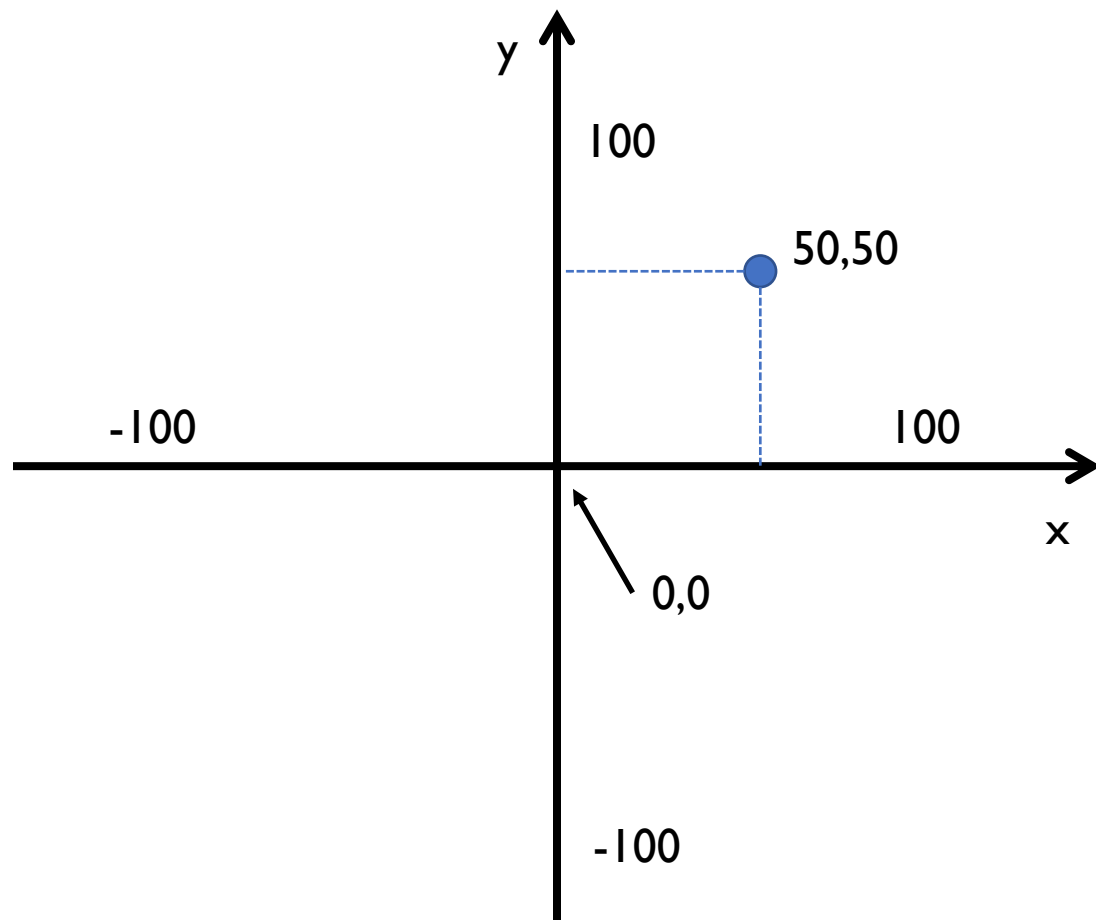
# 程序框架

- 首先: import海龟模块
- 然后: 生成一个海龟
- 进行各种绘制
- 最后:
  - (可选) 隐藏海龟
  - 结束绘制



```
1  # Archimedean spiral
2  # 阿基米德螺线
3  #  $r=a+b*\theta$ 
4
5  import turtle
6  import cmath
7
8  t = turtle.Turtle()
9  a, b = 0, 5
10 for angle in range(0, 3600, 5):
11     theta = angle / 360 * 2 * cmath.pi
12     r = a + b * theta
13     c = cmath.rect(r, theta)
14     t.goto(c.real, c.imag)
15 t.hideturtle()
16 turtle.done()
```

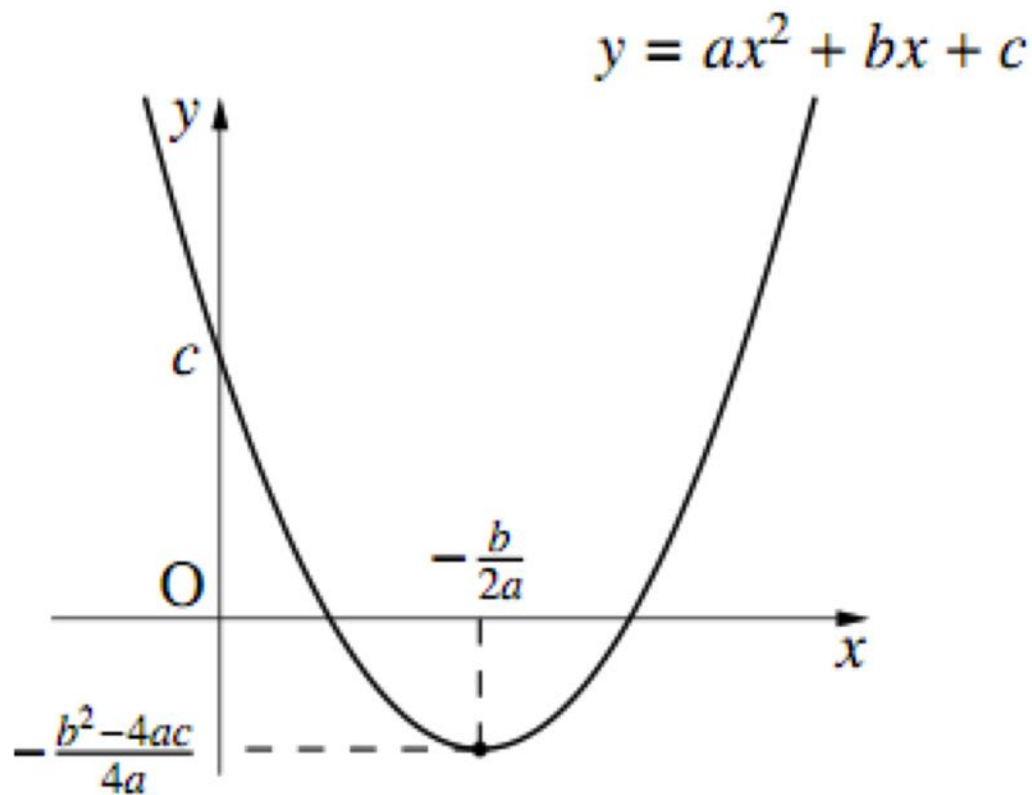
# 让海龟直接去到某个位置goto(x,y)



- 最开始海龟在 (0,0)
- 可以用goto让海龟去到任何地方
- 用position()获取海龟当前位置

# 平面直角坐标系曲线绘制

- 如何用循环语句绘制数学曲线？
- $y=ax+b$
- $y=ax^2+bx+c$
- $y=\sin(x)$



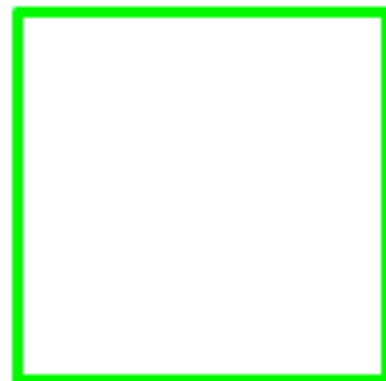
# 海龟函数的小结

- 前进forward(n)后退backward(n)
  - 缩写: fd(n)、bk(n)
- 左转left(n)、右转right(n)
  - 缩写: lt(n)、rt(n)
- 画笔
  - 笔画颜色pencolor(颜色名称)
  - 笔画粗细pensize(n)
- 抬笔penup()、落笔pendown()
  - 缩写pu()、pd()
- 填充
  - 填充颜色fillcolor(颜色名称)
  - 填充开始begin\_fill()
  - 填充结束end\_fill()
- 坐标控制
  - 直接到达goto(x,y)
  - 获取坐标position()
  - 计算距离distance(x,y)



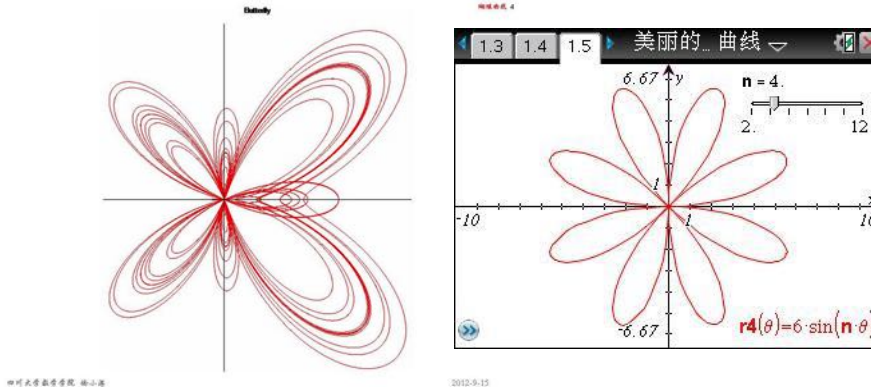
# 上机练习

- 绘制一个边长100的线宽为5的绿色正方形
- 绘制一个边长100的线宽为7的红色五角星。
- 绘制正弦余弦两条曲线 $x=0\sim 2\pi$ 
  - `import math`
  - `math.pi`
  - `math.sin(x)`
  - `math.cos(x)`



# 【H4】 绘制美丽的曲线图

- 上网搜索椭圆、心形线、蝴蝶曲线、玫瑰线等各种美丽曲线的数学方程
- 用海龟作图编程绘制，并通过各种图形的组合、填充、着色等
- 绘制美丽的曲线图
- 提交python程序，和作品截图



## • 参考

- <https://blog.csdn.net/Decting/article/details/8580634>
- <https://baike.baidu.com/item/%E5%BF%83%E5%BD%A2%E7%BA%BF/10018818>
- <https://baike.baidu.com/item/%E6%A4%AD%E5%9C%86%E5%8F%82%E6%95%B0%E6%96%B9%E7%A8%8B/17586034>
- <http://xuxzmail.blog.163.com/blog/static/2513191620128150938651/>
- <https://baike.baidu.com/item/%E7%8E%AB%E7%91%B0%E7%BA%BF>