Python语言程序设计

第三章　数字与字符串

Part One

第一讲 数字类型

数字类型

- 计算机处理表示信息的数据，数据有多种不同的类型。
- Python中的每个值都属于一个特定的类型，一个值的数据类型决定了数据在计算机中如何表示以及能够对该数据进行什么样的操作。
- Python语言本身提供的数据类型称作基本数据类型。
- 数字类型对Python语言中数字的表示和使用进行了定义和规范。

Python语言最常用的数字类型

- 整数类型（int）
- 浮点数类型（float）

```
>>> type(1)
<class 'int'>
>>> type(1.0)
<class 'float'>
```

整数类型

- 与数学中的整数概念一致，没有取值范围限制
- 示例
  - 1010, 99, -217
  - 0x9a, -0X89　(0x, 0X开头表示16进制数)
  - 0b010, -0B101　(0b, 0B开头表示2进制数)
  - 0o123, -0O456　(0o, 0O开头表示8进制数)

进制种类	引导符号	描述
十进制	无	默认情况
二进制	0b 或 0B	由字符0和1组成
八进制	0o 或 0O	由字符0到7组成
十六进制	0x 或 0X	由字符0到9、a到f、A到F组成，不区分大小写

浮点数类型

- 带有小数点及小数的数字 。
- Python语言中浮点数的数值范围存在限制，小数精度也存在限制。这种限制与不同计算机系统有关。

浮点数类型

- 示例
  - 0.0, -77., -2.17
  - 96e4, 4.3e-3, 9.6E5 （科学计数法）
  - 科学计数法使用字母“e”或者“E”作为幂的符号，以10为基数。科学计数法含义如下：

    <a>e<b> = a * 10^b

- 数字类型之间相互运算所生成的结果是“更宽”的类型，基本规则是：
  - 整数之间运算，如果数学意义上的结果是小数，结果是浮点数；
  - 整数之间运算，如果数学意义上的结果是整数，结果是整数；
  - 整数和浮点数混合运算，输出结果是浮点数；

- 内置的数值运算操作符

操作符	描述
x + y	x与y之和
x - y	x与y之差
x * y	x与y之积
x / y	x与y之商
x // y	x与y之整数商，即：不大于x与y之商的最大整数
x % y	x与y之商的余数，也称为模运算
-x	x的负值，即：x*(-1)
+x	x本身
x**y	x的y次幂，即：x^y

- 算数运算示例
  - 将华氏度（F）转化为摄氏度（C）
  - 转化公式 $C = \frac{5}{9}(F - 32)$
  - 假设 F = 75，则相应的Python代码为：
    - 5 / / 9 * (75 – 32) x
    - 5 / 9 * (75 – 32) √
    - 5.0 / 9 * (75-32) √
  - 为什么？
    - Python 中，“/”是真除法，而“//”表示向下取整除（floor division）
    - “/”的结果一定是浮点数；“//”运算有浮点数参与则结果为浮点数，否则为整数。即使结果为浮点数也将舍弃小数部分

- 求余运算符（%）
  - 如：10 % 3
- 应用
  - 若今天是星期六，则10天后是星期几？
    - (6 + 10) % 7
  - 判断一个数 x 是否为偶数
    - x % 2 是否等于 0

复合赋值

- 算术运算符可以和赋值运算符(=)组合在一起形成复合赋值运算。

```
count = count + 1
可以写成
count += 1
```

提示：复合运算符中不要加空格！

复合赋值

```
a = 2
a *= a + 1
print(a)
```

```
6
```

```
a *= a + 1
等同于
a = a * (a + 1)
```

数字类型的转换

- 数值运算操作符可以隐式地转换输出结果的数字类型。例如，两个整数采用运算符“/”的除法将会输出浮点数结果。
- 通过内置的数字类型转换函数可以显式地在数字类型之间进行转换 。
  - int(4.5) = 4
  - float(4) = 4.0

函数	描述
int(x)	**将x转换为整数，x可以是浮点数或字符串**
float(x)	**将x转换为浮点数，x可以是整数或字符串**

Part Two

第二讲 数学函数

内置的数值运算函数

- Python解释器提供了一些内置函数，在这些内置函数之中，有6个函数与数值运算相关

函数	描述
abs(x)	x的绝对值
divmod(x, y)	(x//y, x%y)，输出为二元组形式（也称为元组类型）
pow(x, y[, z])	(x**y)%z，[..]表示该参数可以省略，即：pow(x,y)，它与x**y相同
round(x[, ndigits])	对x四舍五入，保留ndigits位小数。round(x)返回四舍五入的整数值
max(x_1, x_2, …, x_n)	x_1, x_2, …, x_n的最大值，n没有限定
min(x_1, x_2, …, x_n)	x_1, x_2, …, x_n的最小值，n没有限定

- 模块（module）
  - 实现一定的功能的 Python 脚本集合
- 引入模块
  - import module_name
- math模块
  - import math
  - 查看模块内容
    - dir(math)
  - 查看帮助
    - help(math.sin)

```
>>> dir(math)
['__doc__', '__file__', '__name__',
'__package__', 'acos', 'acosh', 'asin', 'asinh',
'atan', 'atan2', 'atanh', 'ceil', 'copysign',
'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp',
'expm1', 'fabs', 'factorial', 'floor', 'fmod',
'frexp', 'fsum', 'gamma', 'hypot', 'isinf',
'isnan', 'ldexp', 'lgamma', 'log', 'log10',
'log1p', 'modf', 'pi', 'pow', 'radians', 'sin',
'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

math库概述

- 首先使用保留字import引用该库
  - 第一种：import math

    对math库中函数采用math.<b>()形式使用

```
>>>import math
>>>math.ceil(10.2)
11
```

  - 第二种: from math import <函数名>

    对math库中函数可以直接采用<函数名>()形式使用

```
>>>from math import floor
>>>floor(10.2)
10
```

□math库解析

□math库包括4个数学常数

常数	数学表示	描述
math.pi	π	圆周率，值为3.141592653589793
math.e	*e*	自然对数，值为2.718281828459045
math.inf	∞	正无穷大，负无穷大为-math.inf
math.nan		非浮点数标记，NaN（Not a Number）

□math库解析

- □math库包括16个数值表示函数

函数	数学表示	描述
math.fabs(x)	$\|x\|$	返回x的绝对值
math.fmod(x, y)	x % y	返回x与y的模
math.fsum([x,y,…])	x+y+…	浮点数精确求和
math.ceil(x)	$\lceil x \rceil$	向上取整，返回不小于x的最小整数
math.floor(x)	$\lfloor x \rfloor$	向下取证，返回不大于x的最大整数
math.factorial(x)	x!	返回x的阶乘，如果x是小数或负数，返回ValueError
math.gcd(a, b)		返回a与b的最大公约数
math.frepx(x)	$x = m * 2^e$	返回(m, e)，当x=0，返回(0.0, 0)
math.ldexp(x, i)	$x * 2^i$	返回$x * 2^i$运算值，math.frepx(x)函数的反运算
math.modf(x)		返回x的小数和整数部分
math.trunc(x)		返回x的整数部分
math.copysign(x, y)	$\|x\| * \|y\|/y$	用数值y的正负号替换数值x的正负号
math.isclose(a,b)		比较a和b的相似性，返回True或False
math.isfinite(x)		当x为无穷大，返回True；否则，返回False
math.isinf(x)		当x为正数或负数无穷大，返回True；否则，返回False
math.isnan(x)		当x是NaN，返回True；否则，返回False

□math库解析

□math库包括8个幂对数函数

函数	数学表示	描述
math.pow(x,y)	x^y	返回x的y次幂
math.exp(x)	e^x	返回e的x次幂，e是自然对数
math.expml(x)	e^x-1	返回e的x次幂减1
math.sqrt(x)	$\sqrt{x}$	返回x的平方根
math.log(x[,base])	$\log_{base} x$	返回x的对数值，只输入x时，返回自然对数，即$\ln x$
math.log1p(x)	$\ln(1+x)$	返回1+x的自然对数值
math.log2(x)	$\log x$	返回x的2对数值
math.log10(x)	$\log_{10} x$	返回x的10对数值

□math库解析

- □math库包括六个“三角双曲函数”

函数	数学表示	描述
math.degree(x)		角度x的弧度值转角度值
math.radians(x)		角度x的角度值转弧度值
math.hypot(x,y)	$\sqrt{x^2+y^2}$	返回(x,y)坐标到原点(0,0)的距离
math.sin(x)	sin x	返回x的正弦函数值，x是弧度值
math.cos(x)	cos x	返回x的余弦函数值，x是弧度值
math.tan(x)	tan x	返回x的正切函数值，x是弧度值
math.asin(x)	arcsin x	返回x的反正弦函数值，x是弧度值
math.acos(x)	arccos x	返回x的反余弦函数值，x是弧度值
math.atan(x)	arctan x	返回x的反正切函数值，x是弧度值
math.atan2(y,x)	arctan y/x	返回y/x的反正切函数值，x是弧度值
math.sinh(x)	sinh x	返回x的双曲正弦函数值
math.cosh(x)	cosh x	返回x的双曲余弦函数值
math.tanh(x)	tanh x	返回x的双曲正切函数值
math.asinh(x)	arcsinh x	返回x的反双曲正弦函数值
math.acosh(x)	arccosh x	返回x的反双曲余弦函数值
math.atanh(x)	arctanh x	返回x的反双曲正切函数值

实例: 天天向上的力量

一年365天，以第1天的能力值为基数，记为1.0，当好好学习时能力值相比前一天提高1‰，当没有学习时由于遗忘等原因能力值相比前一天下降1‰。每天努力和每天放任，一年下来的能力值相差多少呢？

实例: 天天向上的力量

一年365天，以第1天的能力值为基数，记为1.0，当好好学习时能力值相比前一天提高1‰，当没有学习时由于遗忘等原因能力值相比前一天下降1‰。每天努力和每天放任，一年下来的能力值相差多少呢？

运行结果，每天努力1‰，一年下来将提高44%

```
import math
dayup=math.pow((1.0+0.001),365)  #提高0.001
daydown=math.pow((1.0-0.001),365)  #放任0.001
print("向上：",dayup,"向下：",daydown,"能力差值：",dayup-daydown)
```

Part Three

第三讲 字符串类型

字符和字符串

- 字符串必须被括在单引号’ ’ 、双引号 “” 或者三引号''' 里。
- Python没有字符数据类型，一个字符的字符串代表一个字符

```
>>> letter = 'A'
>>> numchar = '4'
>>> message = 'Good morning'
>>> print(letter)
A
>>> message
'Good morning'
>>>
```

ASCII码

- 计算机内部使用二进制数。
- 把一个字符映射成它对应的二进制称为字符编码
- 最早的字符编码是美国标准信息交换码ASCII，仅对10个数字、26个大写英文字母、26个小写英文字母及一些其他符号进行了编码。

  ‘A’ - 65

  ‘a’ - 97

  ‘0’ - 48

Unicode码

■Python的字符串使用Unicode编码。

■Unicode编码兼容ASCII码，支持世界上各种语言的文本。

■Python提供函数ord返回字符的编码，chr返回编码所代表的字符。

```
>>> ord('A')
65
>>> ord('中')
20013
>>> chr(65)
'A'
>>> chr(20013)
'中'
```

- 无论是一个数字、英文字母，还是一个汉字，在统计字符串长度时都按一个字符对待和处理。

```
>>> s="中国辽宁沈阳"
>>> len(s)
6
>>> s="中国辽宁沈阳ABCDE"
>>> len(s)
11
```

转义符

- 输出带有引号的字符串：

```
>>> print("He said, "I am student"")
SyntaxError: invalid syntax
```

有错误？

- Python语言转义符： \
  - 输出带有引号的字符串，可以使用转义符符号\和 ‘’ 在一起代表一个字符。
  - 使用 \\ 输出带有转义符\的字符串

```
>>> print("\"大家好！\"")
"大家好！"
>>> print("符号\\是转义符")
符号\是转义符
>>>
```

转义符

字符转义序列	名称
\b	退格符
\t	制表符
\n	换行符
\r	回车符
\\	反斜线
\’	单引号
\”	双引号

- 字符串之间可以通过+或*进行连接
  - 加法操作(+)将两个字符串连接成为一个新的字符串
  - 乘法操作(*)生成一个由其本身字符串重复连接而成的字符串

```
>>> message="Welcome "+"to "+"Python"
>>> message
'Welcome to Python'
>>> message=3*"Hello "
>>> message
'Hello Hello Hello '
```

- 数字转换成字符串
  - str函数

```
>>> name = "Jack" + 1001
TypeError: must be str, not int
>>> id = 1001
>>> name = "Jack" + str(id)
>>> print(name)
Jack1001
```

Part Four

第四讲 输出格式控制

print函数参数

- print函数的sep和end参数

```
print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

```
print('AA', 'BB', sep='1', end=' ')
print('CC', end='D')
print('EE', 'FF', sep='2')
```

AA1BB CCDEE2FF

格式化控制方法

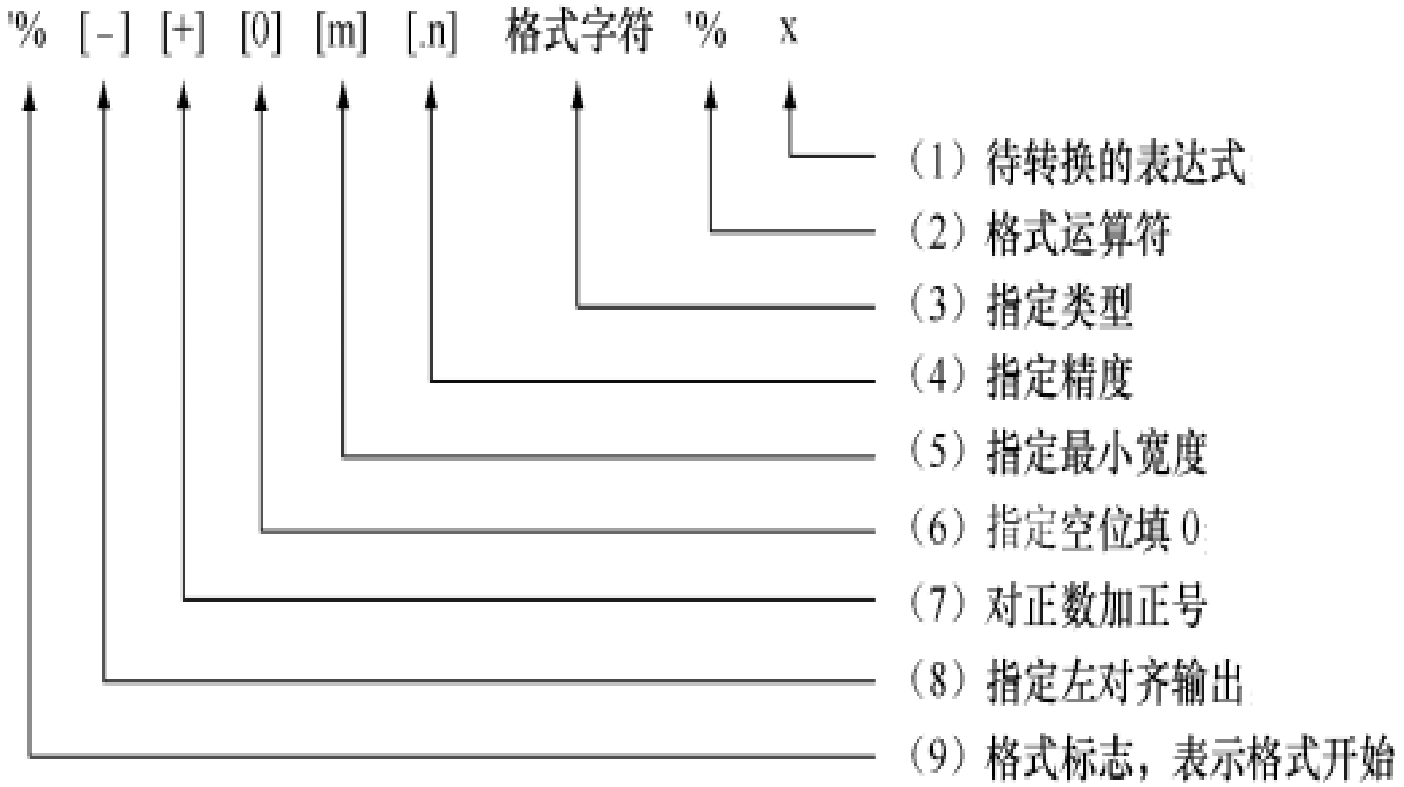
- ✓字符串格式控制表达式
- ✓format函数
- ✓字符串format方法

语法　formatstring % ($value_1$, $value_2$, ···, $value_n$)

示例

```
>>> "a = %d, b = %.2f" % (123, 123.45678)
'a = 123, b = 123.46'
```

字符串格式控制表达式


% [-] [+] [0] [m] [.n] 格式字符 % x
(1) 待转换的表达式
(2) 格式运算符
(3) 指定类型
(4) 指定精度
(5) 指定最小宽度
(6) 指定空位填 0
(7) 对正数加正号
(8) 指定左对齐输出
(9) 格式标志，表示格式开始

常用的格式字符

格式字符	说明
%s	字符串 (采用str()的显示)
%r	字符串 (采用repr()的显示)
%c	单个字符
%d	十进制整数
%i	十进制整数
%o	八进制整数
%x	十六进制整数
%e	指数 (基底写为e)
%E	指数 (基底写为E)
%f、%F	浮点数
%g	指数(e)或浮点数 (根据显示长度)
%G	指数(E)或浮点数 (根据显示长度)
%%	一个字符"%"

字符串格式控制表达式

```
>>> x=1234
>>> sa="%o"%x
>>> sa
'2322'
>>> sm="%x"%x
>>> sm
'4d2'
>>> "%s"%65
'65'
>>>
```

语法　format(value, 格式控制字符串)

示例

```
>>> format(123.456,".2f")
'123.46'
```

- 字符串format方法

语法 <模板字符串>.format(<逗号分隔的参数>)

示例

```
>>> "{}年全国棉花产量{}万吨，增长{}%".format("2018", 44.4, 7.8)

'2018年全国棉花产量44.4万吨，增长7.8%'
```

对象和方法

- Python中所有的数据实际都是对象。
- 一个对象的类型由类决定。
  - 字符串的类是str
  - 整数的类是int
  - 浮点数的类是float
- 可以在对象上执行由函数定义的操作，Python中对象所用的函数称作方法。
- 方法的调用格式 object.method()

```
>>> s = "Welcome"
>>> s.lower()
'welcome'
>>> s.upper()
'WELCOME'
>>> s
'Welcome'
```

- format()方法的基本使用
  - format()方法中模板字符串的槽除了包括参数序号，还可以包括格式控制信息。此时，槽的内部样式如下：{<参数序号>: <格式控制标记>}
  - 其中，格式控制标记用来控制参数显示时的格式。格式控制标记包括：<填充><对齐><宽度>,<.精度><类型>6个字段，这些字段都是可选的，可以组合

:	<填充>	<对齐>	<宽度>	,	<.精度>	<类型>
引导符号	用于填充的单个字符	< 左对齐 > 右对齐 ^ 居中对齐	槽的设定输出宽度	数字的千位分隔符 适用于整数和浮点数	浮点数小数部分的精度 或 字符串的最大输出长度	整数类型 b, c, d, o, x, X 浮点数类型 e, E, f, %

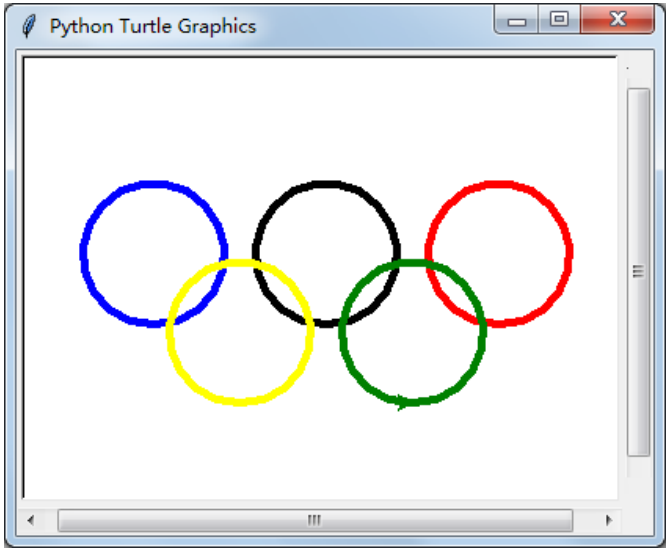
◘format()方法的基本使用

```
>>> print("The number {0:,} in hex is: {0:#x}, the number {1} in oct is {1:#o}".format(5555,55))
The number 5,555 in hex is: 0x15b3, the number 55 in oct is 0o67
>>> print("The number {1:,} in hex is: {1:#x}, the number {0} in oct is {0:o}".format(5555,55))
The number 55 in hex is: 0x37, the number 5555 in oct is 12663
>>> print("my name is {name}, my age is {age}, and my QQ is {qq}".format(name="Dong
    Fuguo",age=40,qq="30646****"))
my name is Dong Fuguo, my age is 40, and my QQ is 30646****
>>> '{0:<8d},{0:^8d},{0:>8d}'.format(65) #设置对齐方式
'65      ,   65   ,      65'
```

Part Five

第五讲 Turtle案例

问题：绘制奥林匹克环标志

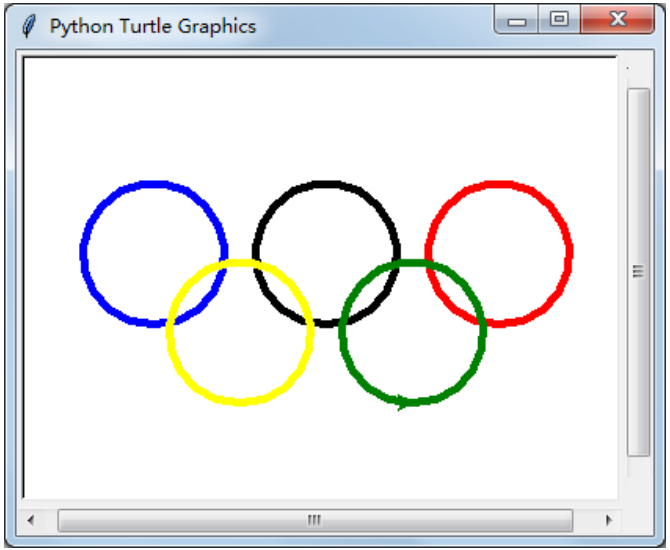


turtle常用函数（二）

函数名	功能
turtle.circle(r)	画圆，半径r为正(负)，表示圆心在画笔的左边(右边)画圆
turtle.circle(r,steps=[3,4,5…])	step等于几，画以r为边长的几边形
setheading(angle)	设置当前朝向为angle角度
turtle.fillcolor(colorstring)	绘制图形的填充颜色
turtle.color(color1, color2)	同时设置pencolor=color1, fillcolor=color2
turtle.begin_fill()	准备开始填充图形
turtle.end_fill()	填充完成
turtle.clear()	清空turtle窗口，但是turtle的位置和状态不会改变
turtle.reset()	清空窗口，重置turtle状态为起始状态
turtle.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.mainloop()或turtle.done()	启动事件循环，必须是turtle图形程序中的最后一个语句

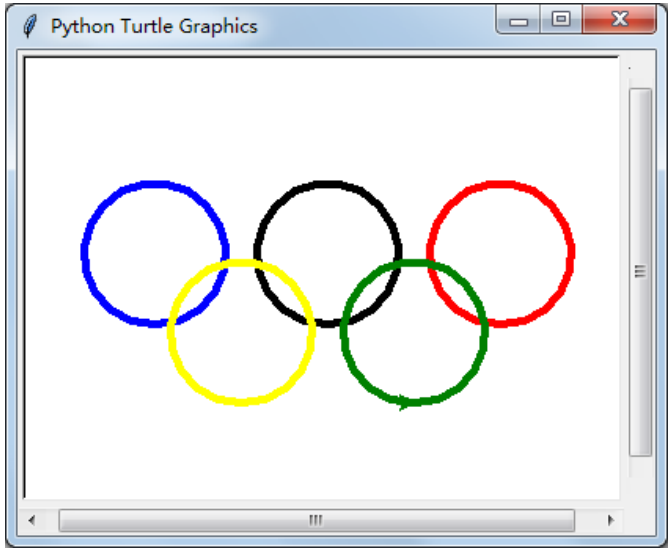
问题：绘制奥林匹克环标志

圆颜色	X轴坐标值	Y轴坐标值
蓝色	-110	-25
黑色	0	-25
红色	110	-25
黄色	-55	-75
绿色	55	-75



问题：绘制奥林匹克环标志

```
import turtle
turtle.pensize(5)          #设置线条宽为5像素
turtle.color("blue")       #设置颜色为蓝色
turtle.penup()             #抬起笔
turtle.goto(-110, -25)     #移动到点(-110,25 )
turtle.pendown()           #放下笔
turtle.circle(45)          #绘制半径为45的圆形
```



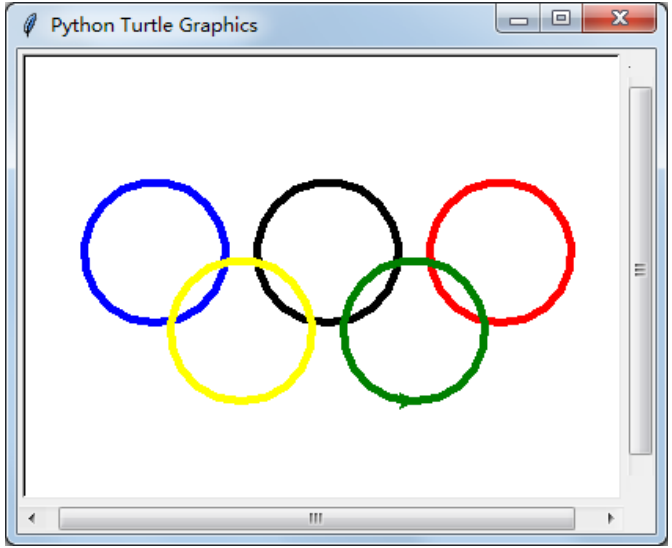
问题：绘制奥林匹克环标志

```
turtle.color("black")
turtle.penup()
turtle.goto(0, -25)
turtle.pendown()
turtle.circle(45)

turtle.color("red")
turtle.penup()
turtle.goto(110, -25)
turtle.pendown()
turtle.circle(45)
```

```
turtle.color("yellow")
turtle.penup()
turtle.goto(-55, -75)
turtle.pendown()
turtle.circle(45)

turtle.color("green")
turtle.penup()
turtle.goto(55, -75)
turtle.pendown()
turtle.circle(45)
turtle.done()
```



- 绘制带颜色填充的图形
- 问题：使用turtle库画出如图所示的三个图形。

Python语言程序设计