

Python语言程序设计

Design and Programming of The Python Language



主讲教师：张小东

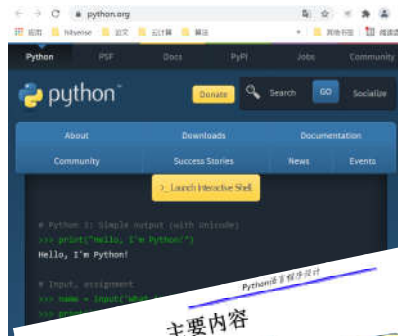
联系方式：z_xiaodong7134@163.com

答疑地点：哈尔滨工业大学(威海)
研学楼423室



群名称:python软工
群 号:220193804

◆ 参考资料



<https://www.python.org/>

教python那个教师给的学习资料



《Python基础教程》，Magnus Lie Hetland, 人民邮电出版社

- 课时安排
 - 授课：32学时
 - 考试：笔试

成绩构成：
大作业30%
+期末考试70%

课程体系结构

数据基础

类 数字
字符串
元组 集合
字节 字典

基本操作

运算符
内置函数
库函数

控制与组织

顺序结构
循环结构
分支结构
函数 异常

高级应用

图形用户界面
网络程序
文件与数据库
Web编程
微服务

面向对象的程序设计

低代码建模

**系统设计
与实现**

- 1、掌握语言基础，设计编写完整的小程序
- 2、以python为实现工具，提高工程业务设计能力
- 3、按软件工程要求，实现一个小型应用系统



第1章 Python基础

主要内容

- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

===Python简介===










Guido van Rossum

- ◆ Python 20世纪80年代末由荷兰人**Guido van Rossum**设计实现的。
- ◆ 1991年，**Ver 0.9.0 Python**，实现了类、函数以及列表、字典和字符串等基本的数据类型，集成了模块系统。
- ◆ 1994年，**Python 1.0**发布了。**1.0**新增了函数式工具。

当前最高版本：python3.11.3

- ◆ 2001年，推出**Python 2.0**，它集成了列表推导式(List comprehension)。
- ◆ 2009年，推出**Python 3.0**，为了不带入过多的累赘，**Python 3.0**在设计的时候没有考虑向下兼容。

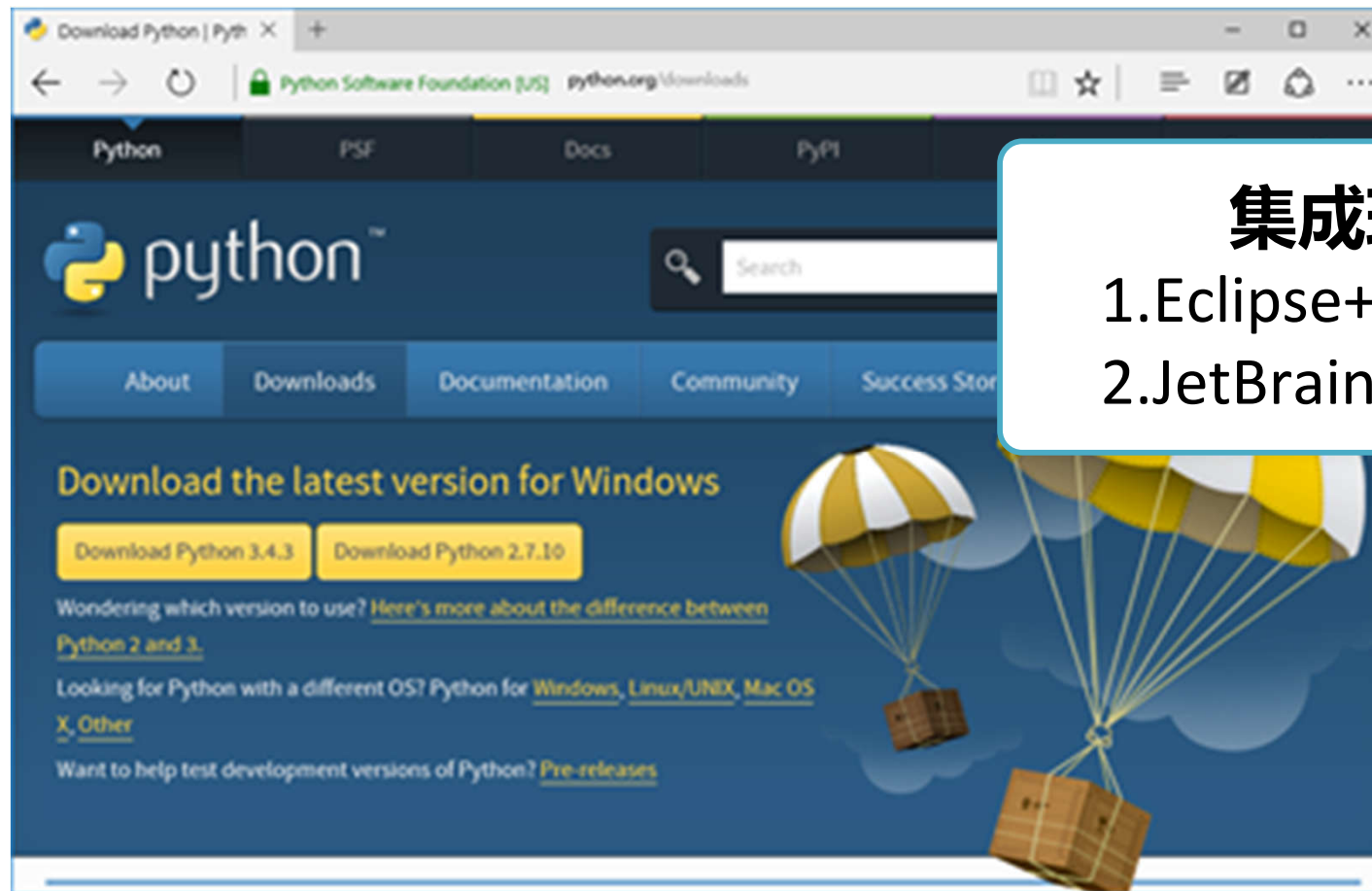
===Python简介===

Apr 2023	Apr 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.51%	+0.59%
2	2			C	14.41%	+1.71%
3	3			Java	13.23%	+2.41%
4	4			C++	12.96%	+4.68%
5	5			C#	8.21%	+1.39%
6	6			Visual Basic	4.40%	-1.00%
7	7			JavaScript	2.10%	-0.31%
8	9	⬆		SQL	1.68%	-0.61%
9	10	⬆		PHP	1.36%	-0.28%
10	13	⬆		Go	1.28%	+0.20%
11	12	⬆		Delphi/Object Pascal	1.23%	+0.05%
12	8	⬇		Assembly language	1.03%	-1.31%
13	16	⬆		Classic Visual Basic	0.92%	+0.09%
14	20	⬆		MATLAB	0.86%	+0.12%
15	24	⬆		Scratch	0.79%	+0.13%

===Python简介

python-3.7-amd64.exe

◆ Python的安装

<https://www.python.org/downloads/>

集成环境

1. Eclipse+pyDev
2. JetBrains PyCharm

===Python的基本语法===

 请问你的pyhton学习目标是什么?

第1章 Python基础

主要内容

- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

===Python的基本语法===

◆ 标识符

- 作用：表示变量、函数、类、模块等对象的名称
- 规则：由字母、数字、“_”组成
首字符不能是数字
大小写敏感

◆ 关键字

- Python语言规定了语法含义，如False, class
- 查看关键字的方法

```
>>>help()
```

```
Help>keywords
```

```
Help>return
```

===Python的基本语法===

◆ 预定义标识符

- Python语言包含的预定义的内置类、异常、函数等，如 float、input、print。

```
>>>dir(__builtins__) #查看异常名和函数名
```

◆ 保留标识符

- “_” 在交互式执行中使用，代表计算结果，如

```
>>>100+200    #300
```

```
>>>_+200      #500
```

- “__*__”系统定义的函数名字

```
__new__() #创建新对象的函数
```

```
__init__() #构造函数。
```

===Python的基本语法===

◆ Python语句

- 通常一行编写一个语句例如：

```
print('Hello')
```

```
print('I am Python')
```

- Python语句的结束符：换行、；(多条语句同行)

◆ 缩进规则

- 在代码行前面添加空格或Tab，代表从属关系
- 平级的语句行（代码块）的缩进必须相同
- 分支、循环、函数定义都采用缩进格式，且末尾有“:”


◆ 注释

单行注释：#

多行注释：单引号（'''）

双引号（"""）

===Python的基本语法===

 请问python单词有哪些分类？ 缩进规则有哪些？

===Python的基本语法===

◆ 输入

<变量>=input([提示])

【例】 name = input("请输入名字：")
 strc = input("请输入字符串：")

◆ 输出

print(<输出项列表>,sep=<分隔符>,end=<结束符>)

【例】 print(name, strc , sep=', ', end=';')

```
>>> name=input("请输入名字：")
请输入名字：张三
>>> strc=input("请输入字符串：")
请输入字符串：String
>>> print(name, strc, sep=', ', end=';')
张三,String;
```

===Python的基本语法===

【例1-1】打印贺卡，如下所示：

```
#####  
<人名1>  
    Happy holiday to you.  
        yours <人名2>  
#####
```

要求：人名由键盘输入。

分析：（1）“#####”与祝福语为输出常量
（2）人名是输入变量的输出。

===Python的基本语法===

【例1-1】打印贺卡

```
name1=input("请输入收卡人：")
name2=input("请输入送卡人：")
print("#####")
print(name1)
print()
print("happy holiday to you.")
print()
print("          Yours",name2)
print("#####")
```


===Python的基本语法===

 请打印一张名片，格式如下所示。

```
#####
*
***    <人名2>
*
哈工大 软件学院 python设计小组头儿
#####
```

第1章 Python基础

主要内容

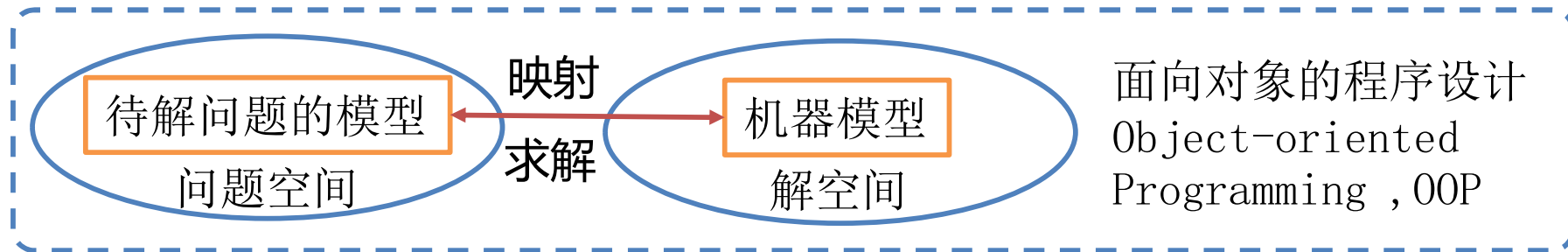
- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

===类与对象===

◆ 对象

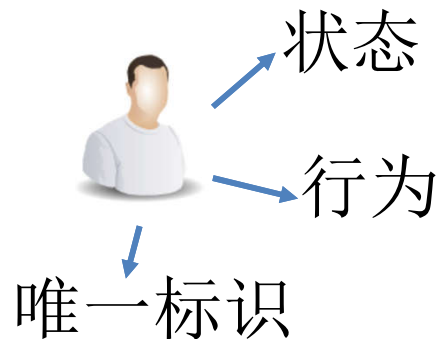
【问题求解】

OO P技术：把相互作用的事物看成集合，用
属来描述事物，而把对它的操作定义为方法



【定义】 问题空间的元素及其在解空间中表示

【特征】



类型不能改变的



可变对象(Mutable)



不可变对象(Immutable)



某种类型事务的一个具体实例， **python** — 一切皆对象。

===类与对象===

◆ 对象

【观测对象】

id: 获取对象的标识

type: 获取对象类型

```
>>> id(11)
```

```
1657845120
```

```
>>> id('python')
```

```
48172648
```

```
>>> type(12)
```


```
<class 'int'>
```

```
>>> type(1.2)
```

```
<class 'float'>
```

del: 对象的删除:

del(对象名称)

 如何检测对象是否存在?

===类与对象===

◆ 类

【定义】

```

class ClassName(Base1, Base2, Base3):
    #定义私有属性,在类外部无法直接进行访问
    #定义构造方法
    def __init__(self,...):
        ...
    #定义其它方法
    ...
    #定义析构方法
    def __del__(self,...):
        ...

```



类规定了对象所具有的属性和操作这些属性的方法

===类与对象===

◆ 常量

【定义】 内存中用于保存固定值的单元，在程序运行中其值不能发生改变。

【类型】 数字、字符串、布尔值和空值等。

【引用库中常量】

```
>>> from math import *  
>>> pi  
3.141592653589793  
>>> e  
2.718281828459045
```

===类与对象===

◆ 变量

【定义】运行过程中，值可以改变的量。

【创建方法】变量是一个标识符，通过赋值运算符(=)创建，指向一个对象。

```
>>> a=112
```

创建一个变量a

```
>>> type(a)
```

```
<class 'int'>
```

a指向整数类型对象

```
>>> id(a)
```

```
1657848352
```

```
>>> id(112)
```

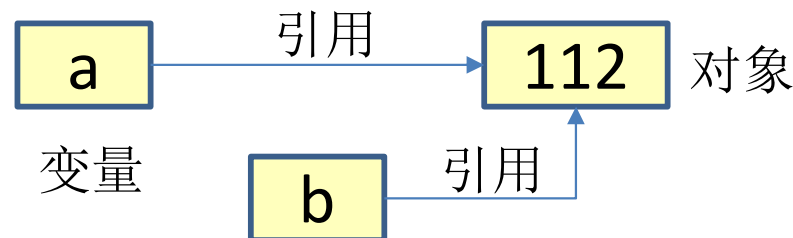
```
1657848352
```

id标识一样

```
>>> b=112
```

```
>>> id(b)
```

```
1657848352
```



===类与对象===

◆ 变量

【特色】

- (1) 不需要声明
- (2) 可以随时赋不同类型的值
- (3) 可以使用一个赋值符号给多个变量赋值

```
>>> x,y,z=5,7,8
```

◆ 对象和数据

对象包含属性和方法

数据是某类对象的属性值

```
>>> a=5  
>>> a.bit_length()
```

5为对象，
通过a指向5

调用对象的方法

◆ 调用对象的方法

<变量名/对象>.<方法> (<参数>)

===Python的基本语法===



请简述python变量声明的特点。

第1章 Python基础

主要内容

- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

===数字类型===

◆ 整数型

【类】 `int`

【说明】 长整型，内存的长度

【例】 `a = 5`

◆ 浮点型

【类】 `float`

【说明】 科学记数法：<实数>E<整数>，如

【例】 `a=1.25; a=1.2E3 #1.2*103`

如何查看数据类型 ?

◆ 复数类型

【类】 `complex`

【说明】 复数的实部和虚部都是浮点数，且至少有一个虚部。

【例】 `x,y,z,m =3+4j, 8-7j, 0.0j, 1j`

不能省略

===数字类型===

◆ 布尔型

【类】 bool

【说明】 用于表示逻辑判断的结果，如True和False

区分大小写

【例】 a=True

◆ 高精度数与分数

(1) Decimal类型

```
>>>from decimal import Decimal  
>>> a=Decimal(1)/Decimal(3)
```

(2) 分数

from fractions import Fraction

Fraction(numberator = 0, denominator=1)

Fraction(other_fraction)

===数字类型===

(2) 分数

from fractions import Fraction**Fraction(float)****Fraction(decimal)****Fraction(string)****a=Fraction(6.25)****print(a)**

输出: ?

>>> from fractions import Fraction

>>> Fraction(1,2) #Fraction(1, 2)

>>> Fraction(12,20) #Fraction(3, 5)

>>> Fraction(1.25) #Fraction(5, 4)

>>> a=Fraction(1,3)

>>> b=Fraction(2,3)

>>> print(a+b) #1

第1章 Python基础

主要内容

- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

==运算符==

【运算】对数据的变换

【运算符】运算的符号

【操作数】运算数、操作对象

【表达式】按计算需求将操作数和运算符组合


【学习要点】优先级、操作数

◆ 算术运算符

(1) 符号：+，-，*，/，%，//，**

(2) 数字：全部适用，其中//为整除，**为指数

(3) 字符串：+为两个字符串连接，x*n为复制x n次

求： $\sqrt[2]{7}$ 

==运算符==

◆ if ...else

```
>>> x=1;y=2;z=3
>>> m = x if y else z    #m=1
>>> y=0
>>> m = x if y else z    #m=3
```

◆ 逻辑运算符

符号: **or, and, not**

```
>>>x=0;y=2
>>>x or y
>>>x and y
>>>not x
```


==运算符==

◆ 关系运算符

符号: <, <=, >, >=, !=, ==

```
>>>x=0;y=2
>>>x > y
>>>x <= y
```

```
a=3
a*=a+3
print(a)
输出: ?
```

◆ 复合运算符

符号: +=, -=,

◆ 位运算符

符号: |, ^, &, >>, <<, ~

◆ 成员运算符

符号: in, not in, is, is not

```
>>> y="abcde";x="bc"
>>> x in y      #True
>>> x not in y  #False
>>> x is y      #False
>>> x is not y  #True
```

二进制：计算机都按补码形式存储数字

补码：正数=原数 5 5 0000 0101

负数：原数的符号位不动，各位求反+1

-5 1000 0101 1111 1011

3|5= 3^5

0000 0011 0000 0011

| 0000 0101 ^0000 0101

0000 0111 0000 0110

◆ 位运算符

符号：|, ^, &, >>, <<, ~

>> 3>>1 0000 0011 -> 0000 0001 3//2=1

<< 3<<1 0000 0011 -> 000 00110 3*2=6

~3 0000 0011 1111 1100 -> 1 000 0011 -> 1 000 0100

==运算符==

◆ 运算符优先级

运算符	描述
<code>(expressions...),</code> <code>[expressions...], {key: value...}, {expressions...}</code>	绑定或加圆括号的表达式, 列表显示, 字典显示, 集合显示
<code>x[index], x[index:index], x(arguments...), x.attribute</code>	抽取, 切片, 调用, 属性引用
<code>await x</code>	await 表达式
<code>**</code>	乘方 [5]
<code>+x, -x, ~x</code>	正, 负, 按位非 NOT
<code>*, @, /, //, %</code>	乘, 矩阵乘, 除, 整除, 取余 [6]
<code>+, -</code>	加和减
<code><<, >></code>	移位
<code>&</code>	按位与 AND
<code>^</code>	按位异或 XOR
<code> </code>	按位或 OR
<code>in, not in, is, is not, <, <=, >, >=, !=, ==</code>	比较运算, 包括成员检测和标识号检测
<code>not x</code>	布尔逻辑非 NOT
<code>and</code>	布尔逻辑与 AND
<code>or</code>	布尔逻辑或 OR
<code>if -- else</code>	条件表达式
<code>lambda</code>	lambda 表达式
<code>:=</code>	赋值表达式

第1章 Python基础

主要内容

- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

==内置函数==

【定义】 能够直接被调用的函数， **Python**提供**67**个内置函数

◆ 转换函数

【常用函数名】

abs(x), bin(x), bool(x), int(x), str(x), float(x)

ord(x), complex([real[,imag]]),

bytes([source[,encoding[,errors]]])

【示例】

```
>>> x=-1
>>> abs(x)                #1
>>> bool(x)               #True
>>> complex(x)            #(-1+0j)
>>> str(x)
>>> bytes("string哈","utf-8")
```

==内置函数==

【其它函数】

(1) **divmod(a,b)** 返回a除以b的商和余数

```
>>> x,y=divmod(23,5)
```

(2) **eval(expression)** 返回字符串表达式的值

```
>>> a=1;b=23  
>>> eval("a+b")
```

(3) **pow(x,y)** 返回x的y次方

```
>>> pow(3,3)
```

(4) **sorted()** 返回有序列表

```
>>> a=[5,1,6]  
>>> sorted(a)  
[1, 5, 6]
```

第1章 Python基础

主要内容

- **Python**简介
- **Python**的基本语法
- 类与对象
- 数字类型
- 运算符
- 内置函数
- 综合应用实例

===应用实例===

【1-2】 判断一个4位整数是否为回文数

➤问题分析：回文数例子，**1221**是回文数，因为其千位与个位互换，百位和十位互换后，仍是它本身

➤计算模型：设输入数为**n**，则有

(1) 取出每一位的值

n的千位： **$a=n//1000$**

n的百位： **$b=n//100\%10$**

n的十位： **$c=n//10\%10$**

n的个位： **$d=n\%10$**

(2) 构成逆序数

$m=d*1000+c*100+b*10+a$

(3) **$n==m?$**

===应用实例===

【1-2】判断一个4位整数是否为回文数

➤程序实现

```
n=input("请输入一个四位数：")
n=int(n)
a=n//1000
b=n//100%10
c=n//10%10
d=n%10
m=d*1000+c*100+b*10+a
result=(n==m)
print("回文数的判断结果是：", result)
```

===应用实例===

【1-3】 输入一个年份，判断其是否为闰年

➤ 问题分析：

(1) 能被4整除不能被100整除

(2) 能被400整除

➤ 计算模型：设输入数为 y ，则有

(1) $y \% 4 == 0$ and $y \% 100 != 0$

(2) $y \% 400 == 0$

```
stry=input("请输入年份:")  
y=int(stry)  
result=(y%4==0 and y%100!=0) or (y%400==0)  
print("闰年的判断结果为：",result)
```

===应用实例===

【思考题】将输入的字符转换为小写，并输出其ASCII值

➤ 提示：

- (1) 条件表达式为 **z if x else y**
- (2) 获得字母的**ASCII**值函数**ord()**
- (3) 字符转换函数**chr()**
- (4) 大写字母为'**A'~'Z'**

➤ 程序实现：

```
c=input("请输入一个字符： ")
y=(c if(c>='a' and c<='z') else chr(ord(c)+32))
print("字符： ",c,"ASCII:",ord(c),"转换为： ",
y,"ASCII： ",ord(y))
```

===应用实例===

◆ 程序的组织方式

【模块】 多个函数及变量可以组成模块

【包】 多个模块可以组成包

【库】 多个包可以组成库

◆ 模块和函数的使用

要使用模块中的函数，需要导入模块

导入和使用有两种方式：

(1) import

import 模块名

模块名.函数名(参数)

```
>>>import math  
>>>math.sqrt(2)  
>>>math.sin(0.7)
```

===应用实例===

◆ 模块和函数的使用

(2) from...import...

from 模块名 import 函数名或变量名
函数名(参数)

```
>>>from math import sqrt,sin  
>>>sqrt(2)  
>>>sin(0.7)
```

===应用实例===

【例1-4】输入直角三角形的两个直角边长度a, b, 求斜边

➤ 问题分析

$$(1) c = (a^2 + b^2)^{1/2}$$

(2) 输入的都字符串, 需转换为float类型

➤ 程序实现

```
from math import *
a=float(input('请输入边长:'))
b=float(input("请输入边长:"))
c=sqrt(a*a+b*b)
print("c=",c)
```



```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
===== RESTART: D:\python
请输入边长:3
请输入边长:4
c= 5.0
>>>
===== RESTART: D:\python
请输入边长:4
请输入边长:5
c= 6.4031242374328485
>>>
```

本章小结

- Python的学习内容与计划
- Python的基本概念
- Python的基本语法