21世纪高等教育计算机规划教材

COMPUTER

计算机编程导论
——Python 程序设计

Introduction to Computer Programming in Python

赵家刚 狄光智 吕丹桔 主编
李俊萩 孙永科 熊飞 林宏 副主编

- 思想 - 方法 - 技巧
- 以问题为核心，以框图为算法工具，简洁而快速编写程序
- 各章配有软件开发实例，手把手教学
- 配套 PPT 课件、源代码、教学大纲

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

计算机编程导论——Python程序设计

Introduction to Computer Programming in Python

赵家刚 狄光智 吕丹桔 主编
李俊萩 孙永科 熊飞 林宏 副主编

人民邮电出版社
北京

图书在版编目（CIP）数据

计算机编程导论 : Python程序设计 / 赵家刚, 狄光智, 吕丹桔主编. -- 北京 : 人民邮电出版社, 2013.10
21世纪高等教育计算机规划教材
ISBN 978-7-115-32914-1

Ⅰ. ①计… Ⅱ. ①赵… ②狄… ③吕… Ⅲ. ①软件工具－程序设计－高等学校－教材 Ⅳ. ①TP311.56

中国版本图书馆CIP数据核字(2013)第220717号

内 容 提 要

本书共分为 16 章：第 1 章～第 11 章侧重于 Python 基础知识的讲解，内容包括顺序程序设计、使用序列、选择结构程序设计、循环结构程序设计、字符串、函数的设计和使用、文件的使用、面向对象程序设计、图形用户界面程序设计、网络程序设计和异常处理；第 12 章～第 16 章侧重于 Python 的高级应用和软件开发，内容包括数据库应用程序开发、游戏开发、语音识别软件开发、屏幕广播程序开发和 web2py 编程，每章都包含创作软件实例，适合 Python 爱好者和开发人员阅读、学习或参考。

本书既可作为高等院校程序设计课程的教材，也可作为高职高专程序设计课的教材，还可作为软件开发人员的参考书。

◆ 主　　编　赵家刚　狄光智　吕丹桔
副 主 编　李俊萩　孙永科　熊　飞　林　宏
责任编辑　王　威
执行编辑　范博涛
责任印制　沈　蓉　焦志炜

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市潮河印业有限公司印刷

◆ 开本：787×1092　1/16
印张：17.5　　2013 年 10 月第 1 版
字数：459 千字　　2013 年 10 月河北第 1 次印刷

定价：49.80 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

Python 是一种功能强大的通用型语言，自 1989 年推出至今已有二十多年的历史，成熟且稳定，它支持命令式编程、面向对象程序设计、函数式编程，包含了完善且容易理解的标准库，还有非常丰富的扩展库，能够轻松完成开发任务。与其他计算机程序设计语言不同的是，Python 采用缩进来定义语句块，使得语法非常简洁和清晰，它的编程效率非常高，使用 py2exe 工具还可以将 Python 源代码转换成可以脱离 Python 解释器执行的程序。Python 是一门简单易学的语言，非常适合初学者使用，其在世界范围内的影响力正在逐步上升。作者多年的教学实践表明：与 C 语言等语言相比，使用 Python 作入门语言能使学生更快地掌握编程思想和编程方法，能更快地提高学生的编程能力。

本书以实际问题为核心进行组织和编写，以框图为工具来描述问题的解决步骤，最终用 Python 语言写出程序，旨在培养学生从整体上思考问题和把握问题，并以直观的方式描述问题的解决步骤，训练学生用简洁而快速的方式编写程序。本书以培养学生的编程思想和编程能力为目标，精心设计了各章节的内容以及习题和思考题。书中介绍了程序设计领域的许多基本问题，示范性地用框图表达了解决问题的算法，并用 Python 语言进行实现，旨在逐步培养学生的编程思想和编程能力。

本书由经验丰富、长期从事软件开发和程序设计教学的教师编写，在例题和习题中包含了许多实用的编程方法和编程技巧，将由易到难、由浅入深的启发式教学方法贯穿于每一章，有利于读者阅读和理解。

本书每章都附有一定数量的习题，可以帮助学生进一步巩固基础知识，也可为学生提供更多思考和锻炼机会。此外还配套了 PPT 课件、源代码、教学大纲等丰富的教学资源，读者可登录人民邮电出版社教学服务与资源网（www.ptpress.com.cn）下载。

本书的参考学时为 64 学时，其中实验 32 学时，各章的参考学时见下面的分配表。

章　节	课程内容	学时分配	
		讲授	实验
第 1 章	顺序程序设计	4	4
第 2 章	使用序列	4	4
第 3 章	选择结构程序设计	2	2
第 4 章	循环结构程序设计	4	4
第 5 章	字符串	1	1
第 6 章	函数的设计和使用	3	3
第 7 章	文件的使用	2	2

续表

章　节	课程内容	学时分配	
		讲授	实验
第 8 章	面向对象程序设计	2	2
第 9 章	图形用户界面程序设计	2	2
第 10 章	网络程序设计	2	2
第 11 章	异常处理	2-自学	2-自学
第 12 章	数据库应用程序开发	4-自学	4-自学
第 13 章	游戏开发	4	4
第 14 章	语音识别软件开发	2	2
第 15 章	屏幕广播程序开发	2-自学	2-自学
第 16 章	web2py 编程	4-自学	4-自学
课时总计		32	32

教师可根据专业需要和学生实际情况调整教学内容。

第 1 章、第 5 章、第 8 章、第 12 章、第 13 章以及第 7 章的 7.1 节至 7.3 节由赵家刚编写，第 2 章和第 9 章由狄光智编写，第 3 章、第 14 章以及第 7 章的 7.4 节和 7.5 节由吕丹桔编写，第 4 章、第 11 章和由李俊萩编写，第 6 章由林宏编写，第 10 章由孙永科和熊飞编写，第 15 章由孙永科编写，第 16 章由熊飞编写。

书中难免有不当之处，敬请读者提出宝贵意见，以便再版时修改。

编　者

2013 年 6 月

目　录

第 1 章　顺序程序设计 ……1

1.1　用计算机解决问题的方法 ……1

1.2　程序设计方法 ……1

1.2.1　学会用框图来描绘解决实际问题的步骤 ……2

1.2.2　把框图转换为程序 ……3

1.2.3　理解程序运行过程 ……3

1.2.4　掌握一些基本算法 ……3

1.2.5　学习完整的解决问题的过程 ……3

1.3　程序设计的一般过程 ……4

1.4　顺序程序设计问题 ……4

1.5　顺序程序设计基础知识 ……5

1.5.1　Python 的对象模型 ……5

1.5.2　Python 的变量和引用 ……6

1.5.3　数字 ……7

1.5.4　字符串 ……7

1.5.5　操作符和表达式 ……8

1.5.6　常用内置函数 ……9

1.5.7　对象的删除 ……12

1.5.8　输入/输出 ……12

1.5.9　模块的导入 ……13

1.6　顺序程序设计基础知识的应用 ……14

1.7　代码块的缩进 ……17

1.8　在 Ubuntu 操作系统中使用 Python ……17

1.8.1　交互编程窗口 ……18

1.8.2　在交互式窗口中执行 Python 源程序 ……19

1.8.3　在操作系统的控制台上执行 Python 源程序 ……19

1.8.4　在集成编程环境 IDLE 中编写和执行程序 ……19

1.8.5　集成编程环境 IDLE 中的对象成员提示 ……20

1.9　在 Windows 操作系统中使用 Python ……21

1.10　使用帮助 ……21

本章小结 ……22

习题 ……23

第 2 章　使用序列 ……24

2.1　序列问题 ……24

2.2　序列基础知识 ……28

2.3　列表 ……29

2.3.1　列表定义 ……29

2.3.2　列表的常用操作 ……29

2.3.3　列表常用函数 ……32

2.4　元组 ……33

2.4.1　元组的定义 ……33

2.4.2　元组的常用操作 ……33

2.4.3　元组和列表的区别和转换 ……35

2.4.4　同时赋多个值 ……35

2.5　字典 ……35

2.5.1　字典定义 ……35

2.5.2　字典的常用操作 ……35

2.6　序列基础知识的应用 ……37

本章小结 ……39

习题 ……40

第 3 章　选择结构程序设计 ……41

3.1　选择结构基本问题 ……41

3.2　选择结构基础知识及应用 ……41

3.2.1　表达式与表达式的值 ……42

3.2.2　复合表达式 ……45

3.2.3　选择结构 ……46

本章小结 ……55

习题 ……56

第 4 章　循环结构程序设计 ……57

4.1　循环结构程序设计问题 ……57

4.2 循环结构概述……58
4.3 while 语句……59
4.3.1 while 语句解决不确定循环次数的问题……59
4.3.2 while 语句解决确定循环次数的问题……61
4.3.3 while 语句用于无限循环……62
4.3.4 while 语句应用举例……63
4.4 for 语句……66
4.4.1 for 语句用于序列类型……66
4.4.2 for 语句用于计数循环……68
4.5 break 语句……71
4.6 continue 语句……73
本章小结……74
习题……75

第 5 章 字符串……76

5.1 字符串问题……76
5.2 字符串基础知识……76
5.2.1 字符串格式化……77
5.2.2 字符串的截取……78
5.2.3 字符串的方法……79
5.2.4 与字符串相关的重要内置方法……80
5.3 字符串基础知识的应用……81
本章小结……82
习题……82

第 6 章 函数的设计和使用……83

6.1 问题的引入……83
6.2 黑箱模型……84
6.3 函数基础知识……84
6.3.1 函数的概念及定义……84
6.3.2 形参和实参……85
6.3.3 return 语句……87
6.4 变量的作用域……87
6.4.1 局部变量……87
6.4.2 全局变量……88
6.4.3 命名空间……89
6.5 参数的类型……89
6.5.1 默认参数……89
6.5.2 关键参数……90
6.5.3 可变长度参数……91
6.5.4 序列作实参……92
6.6 函数基础知识的应用……92
本章小结……94
习题……94

第 7 章 文件的使用……95

7.1 与文件有关的问题……95
7.2 文件基础知识……97
7.2.1 文件的打开或创建……98
7.2.2 字符编码……99
7.2.3 文本文件的写入……100
7.2.4 文本文件的读取……102
7.2.5 文件指针的移动……103
7.2.6 二进制文件的写入……103
7.2.7 二进制文件的读取……105
7.3 文件基础知识的应用……107
7.4 文件操作……113
7.4.1 常用文件操作函数……113
7.4.2 文件的复制……115
7.4.3 文件的删除……115
7.4.4 文件的重命名……115
7.4.5 文件的比较……118
7.5 目录操作……119
7.5.1 目录的创建……119
7.5.2 目录的删除……119
7.5.3 目录的遍历……119
本章小结……123
习题……124

第 8 章 面向对象程序设计……125

8.1 面向对象程序设计问题……125
8.2 面向对象程序设计基础知识……126
8.2.1 类和对象……126
8.2.2 实例属性和类属性……127
8.2.3 类的方法……128
8.2.4 构造函数……129
8.2.5 析构函数……129
8.2.6 运算符的重载……130

8.2.7 继承……131
本章小结……133
习题……133

第 9 章 图形用户界面程序设计……134

9.1 图形用户界面的选择和安装……134
9.1.1 常用 GUI 工具介绍……134
9.1.2 wxPython 下载安装……135
9.2 图形用户界面程序设计基本问题……135
9.3 框架的创建和使用……136
9.3.1 wx.Frame 的格式……136
9.3.2 wxPython 的 ID 参数……138
9.3.3 wx.Point 和 wx.Size……138
9.3.4 设置 wx.Frame 的样式……139
9.4 添加窗体控件……140
9.4.1 命令按钮……141
9.4.2 文本控件……142
9.4.3 菜单栏、工具栏和状态栏……146
9.4.4 对话框……148
9.4.5 复选框……150
9.4.6 单选按钮……151
9.4.7 列表框……151
9.4.8 组合框……153
9.4.9 树型控件……153
9.5 使用 Boa-constructor 开发图形用户界面程序……155
9.5.1 Boa-constructor 的安装……155
9.5.2 使用 Boa–constructor 开发图形用户界面程序……156
9.6 图形界面程序设计基础知识的应用……160
本章小结……161
习题……162

第 10 章 网络程序设计……164

10.1 问题的引入……164
10.2 一个简单邮寄过程……165
10.3 TCP/IP 协议簇简介……165
10.4 TCP 和 UDP……168
10.5 UDP 网络编程……168
10.6 UDP 代码解释……169
10.7 UDP 函数介绍……170
10.7.1 socket……170
10.7.2 sendto……171
10.7.3 recvfrom……171
10.8 TCP 网络编程……171
10.9 TCP 代码详解……173
10.10 TCP 函数介绍……174
10.10.1 connect……174
10.10.2 send……174
10.10.3 recv……174
10.10.4 bind……174
10.10.5 listen……174
10.10.6 accept……174
10.11 局域网聊天室……175
10.11.1 需求分析……175
10.11.2 概要设计……175
10.11.3 详细设计……176
10.11.4 编码和测试……177
本章小结……182
习题……183

第 11 章 异常处理……184

11.1 什么是异常……184
11.2 Python 中的异常类……184
11.3 捕获和处理异常……185
11.3.1 try ... except ... 语句……185
11.3.2 try ... except ... else ...语句……186
11.3.3 带有多个 except 的 try 语句……187
11.3.4 捕获所有异常……188
11.3.5 finally 子句……188
11.4 两种处理异常的特殊方法……189
11.4.1 断言语句（assert 语句）……189
11.4.2 上下文管理（with 语句）……189
11.5 引发异常（raise 语句）……190
11.6 采用 sys 模块回溯最后的异常……192
本章小结……192
习题……192

第 12 章 数据库应用程序开发……193

12.1 数据库应用程序的问题描述……193

12.2 Python 数据库应用程序开发概述……194
12.3 SQLite 简介……195
12.4 SQLite 基本功能……196
12.5 SQLite 的可视化工具……200
12.6 数据库应用程序开发……205
本章小结……209
习题……209

第 13 章 游戏开发……210

13.1 图形化的问候问题……210
13.2 Pygame 基础知识……212
13.2.1 Pygame 的安装……212
13.2.2 Pygame 的模块……213
13.2.3 Pygame 的使用……215
13.3 游戏开发……218
本章小结……228
习题……229

第 14 章 语音识别软件开发……230

14.1 speech.py 语音模块的简介……230
14.2 语音识别开发环境的建立……230
14.3 语音识别的配置……231
14.4 语音模块的运用……232
本章小结……239
习题……239

第 15 章 屏幕广播程序开发……240

15.1 屏幕广播程序原理……240
15.2 教师端……241
15.3 学生端……243
15.4 程序运行……246
本章小结……247
习题……247

第 16 章 web2py 编程……248

16.1 网页与 HTML……248
16.1.1 HTML 语言简介……248
16.1.2 HTML 标签简介……249
16.2 web2py 与 MVC……249
16.2.1 安装 web2py……249
16.2.2 web2py 的应用……251
本章小结……257
习题……257

附录 A 一些重要的内建函数……258

附录 B 列表方法……261

附录 C 字典方法……262

附录 D 字符串对象的方法……263

附录 E 在线资源……265

附录 F 使用 py2exe 创建可执行程序……266

附录 G 使用 WinRAR 处理发布的文件清单……269

参考文献……271

第 1 章 顺序程序设计

程序设计的目的是让计算机解决实际问题，其方法就是把人解决问题的思想和方法转化为计算机可以执行的程序。本章研究如何用计算机来解决实际问题，并介绍程序设计的一般方法、顺序程序设计的语言基础和解决实际问题的方法。

1.1 用计算机解决问题的方法

对于一些较为简单的问题，运用数学知识进行简单计算就能解决；对于一些复杂的问题，则可能还需要用到分支、循环等知识。在生活、娱乐、经济、工程、科研等领域中，人们不仅需要用计算机解决简单问题，而且需要解决较为复杂的问题。

用计算机解决问题的一般方法如下

（1）用框图或自然语言描绘出解决问题的步骤。本书用框图描绘。描绘出解决问题的步骤也称为算法。

（2）用程序设计语言来实现解决问题的步骤。即用程序设计语言把框图表示的算法翻译成机器能够理解并执行的程序。

由于计算机不能直接执行高级程序设计语言编写的程序，这里的执行是指由翻译或编译程序进行解释执行或编译成机器代码再执行。本书所用的高级程序设计语言是 Python，关于 Python 的安装请参阅 1.8 节和 1.9 节“Python 的使用”。

用高级程序设计语言编写的程序称为源程序。

1.2 程序设计方法

用计算机解决实际问题的过程称为程序设计。

程序设计的一般方法为：首先用框图描绘出实际问题的解决方案，然后用程序设计语言表达出来，最后在计算机上执行求得计算结果。

学习程序设计要做如下 6 件事。

（1）学会用框图来描绘解决实际问题的步骤。

（2）学习至少一门高级程序设计语言，并熟练使用该语言把自己设计的框图转换为程序。

（3）观看现成的框图，体会解决问题的思想。

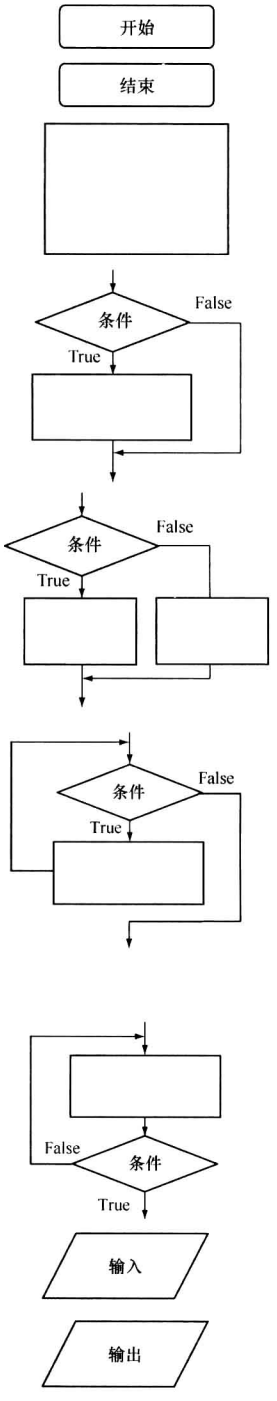
（4）掌握一些常用的基本计算方法，作为搭建自己的框图和程序的基础。

（5）通过一些完整的问题实例，掌握从分析问题、绘制框图到程序实现的全过程。

（6）多做练习并善于总结经验，包括独立分析问题、设计框图、根据框图写出代码、阅读大量代码、模仿例题解决类似问题。

1.2.1 学会用框图来描绘解决实际问题的步骤

框图又称流程图，是表达程序设计思想和程序设计步骤的一种直观的工具。学习程序设计应该首先学会使用框图。下面是框图常用的部件及说明（见图 1-1）。



开始框——用于流程的开始。

结束框——用于流程的结束。

功能框——用于完成计算等功能。

单分支判断框——用于解决单分支问题。
例子：

```
if x>0:
    n=n+1
```

双分支判断框——用于解决双分支问题。
例子：

```
if x>0:
    y=1+2*x
else:
    y=0
print ' y=', y
```

循环框1——用于解决需要反复进行的问题。
例子 1：

```
s=[1, 2, 3]
for x in s:
    print x
```

例子2：

```
i=1
s=0
while i<=100:
    s=s+i
    i=i+1
print s
```

循环框2——用于解决需要反复进行的问题。

```
i=1
s=0
while True:
    s=s+i
    i=i+1
    if i>100:
        break
print s
```

输入框——向程序输入数据。
例子：

```
x=input('x=')
```

输出框——程序向外输出信息。
例子：

```
print's=', s
```

图 1-1　框图常用部件

框图直观且易于修改，有利于人们表达解决问题的思想和方法。对于十分复杂难解的问题，框图起初可以画得粗放一些、抽象一些，首先表达出解决问题的轮廓，然后再细化，或不细化而直接用高级程序设计语言实现框图；对于较为简单的问题，框图可以画得粗放一些也可以画得细致一些。

1.2.2　把框图转换为程序

框图表达了解决问题的思想、方法和步骤，但计算机还不能执行，因而不能得到结果。只有把框图转换为程序，计算机才能执行，从而得到结果。把框图转换为程序需要一种计算机程序设计语言。本书采用简单易学的 Python 语言。下面举例说明。

【问题 1-1】 用户输入一个三位自然数，让计算机输出百位、十位和个位。

分析： 该问题需要把三位数的百位、十位、个位分离出来。三位数除以 100，其整数商就是百位数；用去掉百位数的剩余部分除以 10，其整数商就是十位数；依次类推，可得到个位数。按此思路可画出如下框图（见图 1-2）。

框图：

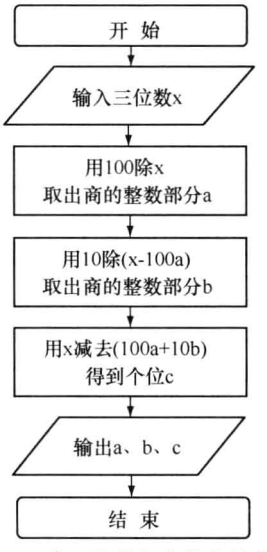


图 1-2　求三位数各位数字的步骤

程序：

根据框图写出

```
#Ques1_1.py
x=input('请输入一个三位数：')
a=x//100
b=(x-100*a)//10
c=x-100*a-10*b
print a, b, c
```

正确性检验：

执行程序，从键盘输入 123，得到输出结果 1　2　3

说明程序是正确的。

1.2.3　理解程序运行过程

编写的程序由一条一条的语句组成。一般情况下语句按顺序逐条在机器中执行，虽然分支语句、循环语句和函数调用语句等可以改变语句运行的顺序，但是改变后语句仍是逐条顺序执行的。编程者需要充分理解计算机程序在内存中的运行原理和过程，能够在程序运行过程中的任意时刻都清楚语句运行到哪里了，当前的变量连接到了哪个对象。只有清楚这些才能在程序调试过程中及时找到出错位置并修改错误，最终让程序按照设计者的意图执行。

1.2.4　掌握一些基本算法

学习程序设计可以从模仿开始。开始阶段可以学习一些常用的基本算法，这样在解决复杂问题之前，有一些基本方法可用。例如，数据的累加算法、累乘算法、求最大值（最小值）算法、求平均值的算法，如何判断某个数是否是素数，如何利用列表解决一维数据问题、二维数据问题，如何利用字符串解决实际问题，如何判断某一年是否为闰年等。本书将通过例子逐步给出这些算法。

1.2.5　学习完整的解决问题的过程

学习程序设计是为了帮助我们利用计算机解决一些实际问题，而不是单纯为了学会一种编程

语言。因此应该掌握整个程序设计的过程，围绕一些具体的问题实例，通过分析问题，描绘出解决问题的框图，最终实现代码并调试成功。只有通过这样一个完整的程序设计过程，才能充分理解程序设计需要完成哪些工作，并且学会判断什么样的问题适合用计算机来解决。

1.3 程序设计的一般过程

程序设计的本质是以数据对象为中心的处理过程再加上输入输出。编写程序解决实际问题的阶段包括分析问题找出解决问题的关键所在、用框图描绘出对实际问题的解决步骤、编写代码、运行调试、正确性检验和观察运行结果。下面通过一个实际问题说明程序设计的一般过程。

框图：

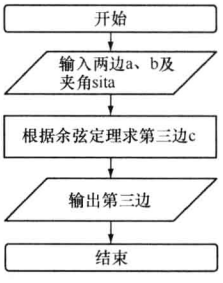


图 1-3 解三角形的步骤

【问题 1-2】 已知三角形的两边及夹角，求第三边。

分析： 这是解三角形的问题，已知两边及夹角，根据余弦定理可求出第三边。因而可画出如下框图（见图 1-3）。

程序： 根据框图写出

```
#Ques1_2.py
import math   #导入数学模块，从而可以使用模块中的数学函数和数学常量
x=input('输入两边及夹角（度）（以逗号分隔）：')
a, b, sita=x
c=math.sqrt(a**2+b**2-2*a*b*math.cos(sita*math.pi/180))
print 'c='+str(c)
```

说明

（1）使用了数学模块中的算术平方根函数 sqrt( )和余弦函数 cos( )及圆周率 pi；

（2）函数 str( ) 的功能是把数字转换为字符串，“+”可用于合并字符串。

正确性检验：

执行程序，从键盘输入 3, 4, 90

得到输出结果 c=5.0

满足勾股定理，说明程序是正确的。

1.4 顺序程序设计问题

【问题 1-3】 输入两只电阻的阻抗，把它们并联，求并联后的阻抗。

分析： 这是电路的并联问题，根据并联公式 1/R=1/r1+1/r2 可算出并联后的阻抗。

程序： 根据框图（见图 1-4）写出

```
#Ques1_3.py
r1, r2=input('请输入两只电阻的阻抗（以逗号分隔）：')
R=1.0/(1.0/r1+1.0/r2)
print('R='+'%6.2f' % R)
```

框图：

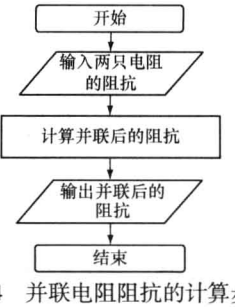


图 1-4 并联电阻阻抗的计算步骤

'%6.2f' % R 是格式化字符串，把浮点数R转换成字符串，保留两位小数（对第 3 位四舍五入），占 6 个字符，不足时，左边补空格。

输入及程序运行结果：

请输入两只电阻的阻抗（以逗号分隔）：100, 200
R=66.67

1.5　顺序程序设计基础知识

用 Python 进行程序设计，涉及许多知识。本节介绍进行顺序程序设计所需的基础知识。分支程序设计和循环程序设计也需要这些基础知识。

1.5.1　Python 的对象模型

对象是内存中的一个单元，包括数值和相关操作的集合。

对象是 Python 语言中最基本的概念，在 Python 中处理的每样东西都是对象。Python 中还有一些基本概念，它们的关系如下：

（1）程序由模块构成；

（2）模块包含语句；

（3）语句包含表达式；

（4）表达式建立并处理对象。

Python 中有许多内置对象可供编程者使用。有些内置对象可直接使用，如数字、字符串、列表、del 等，如表 1-1 所示；有些内置对象需要导入模块才能使用，如正弦函数 sin(x)、随机数产生函数 random( )等。

表 1-1　　常用内置对象

对 象 类 型	例　子
数字	1234, 3.14, 3+4j
字符串	'swfu', "I'm student"
日期	2012-08-25
列表	[1, 2, 3]
字典	{1:'food' ,2:'taste', 3:'import'}
元组	(2, -5, 6)
文件	f=open('data.dat', 'r')
集合	set('abc'), {'a', 'b', 'c'}
布尔型	True, False
空类型	None
编程单元类型	函数、模块、类

下面的代码建立了一个整数对象：

```
>>>110
```

下面的代码建立了一个字符串对象：

```
>>>'Python'
```

下面的代码建立了一个复数对象：

```
>>>3-2j
```

下面的代码建立了一个列表对象：

```
>>>[1,2,3]
```

下面的代码建立了一个元组对象：

```
>>>(1,2,3)
```

下面的代码也建立了一个元组对象：

```
>>>1,2,3
```

1.5.2 Python 的变量和引用

变量是由赋值语句创建的，如 x=3 创建了变量 x。

1. 变量的创建

一个变量（也就是变量名），如 x，当代码第 1 次给它赋值时就创建了它。

2. 引用

在 Python 中从变量到对象的连接称为引用。下面的语句创建了整数型对象 3、变量 x，并使变量 x 连接到对象 3，也称变量 x 引用了对象 3。如图 1-5 所示。

图 1-5　变量连接到对象

```
>>>x=3       #创建了变量 x，并使变量 x 引用整型对象 3
```

事实上，变量拥有自己的存储空间，变量连接到对象是该变量存储了对象单元的内存地址，并没有存储对象的值。

内置对象是可以直接使用的，而变量需要通过赋值语句创建。

第 1 次给变量赋值则创建了变量，再次给该变量赋值则改变了该变量的引用，如：

```
>>>a=3              #创建了变量 a，并使变量 a 引用整型对象 3
>>>a=9.5            #改变了变量 a 的引用，使变量 a 引用浮点型对象 9.5
>>>a=(1, 2, 3)     #改变了变量 a 的引用，使变量 a 引用元组对象(1, 2, 3)
```

变量在进行运算和输出时，自动使用它所引用的对象的值。

3. 共享引用

共享引用是指多个变量引用同一个对象。下面的语句使两个变量都引用同一个对象 3：

```
>>>a= 3
>>>b=a
>>>a='swfu'
>>>a
'swfu'
>>>b
3
```

改变了 a 的引用，但 b 仍然引用对象 3（见图 1-6）。

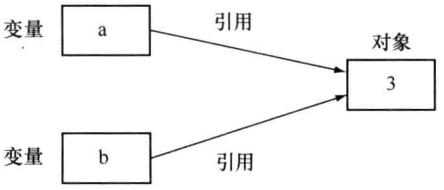


图 1-6　两个变量引用同一个对象

下面的代码展示了 a、b 引用同一个列表对象，通过变量 b 修改对象后的输出情况：

```
>>>a=[1, 2, 3]
>>>b=a
>>>b[0]=9
>>>b
[9, 2, 3]
>>>a
[9, 2, 3]
```

在程序设计中要注意共享引用带来的影响。

1.5.3　数字

数字是 Python 中最常用的对象。

1. 整数

十进制整数：如 0、-1、9、123 等。

十六进制整数：需要 16 个数字 0、1、2、3、4、5、6、7、8、9、a、b、c、d、e、f 来表示整数。为了告诉计算机这是一个十六进制数，必须以 0x 开头，如 0x10、0xfa、0xabcdef。

八进制整数：只需要 8 个数字 0、1、2、3、4、5、6、7 来表示整数。为了告诉计算机这是一个八进制数，必须以 0o 开头，如 0o35、0o11。

二进制整数：只需要 2 个数字 0、1 来表示整数。为了告诉计算机这是一个二进制数，必须以 0b 开头，如 0b101、0b100。

2. 浮点数

浮点数又称小数，如 15.0、0.37、-11.2、9.3e2、314.15qe-2。

3. 复数

复数是由实部和虚部构成的数，如 3+4j、0.1-0.5j。

下面是复数的例子：

```
>>>a=3+4j
>>>b=5+6j
>>>c=a+b
>>>c
8+10j
>>> c.real    #复数的实部
8.0
>>> c.imag  #复数的虚部
10.0
```

1.5.4　字符串

用单引号或双引号引起来的符号系列称为字符串，如'abc'、'123'、'中国'、"Python"。

空串表示为''或 ""，注意是一对单引号或一对双引号。

1．字符串合并

字符串合并运算符是“+”，用法如下：

```
>>>'abc'+'123'
'abc123'
```

2．转义字符

计算机中存在可见字符与不可见字符。可见字符是指可以在屏幕上显示的字符；不可见字符是指换行、制表符等，通常起一定的控制功能，在屏幕上没有直接的显示。不可见字符只能用转义字符来表示，可见字符也可用转义字符表示。转义字符以“\”开头，后接字符或数字，如表1-2所示。

表1-2　　转义字符

转义字符	说　明
\'	单引号
\"	双引号
\\	表示\
\a	发出系统铃声
\n	换行符
\t	纵向制表符
\v	横向制表符
\r	回车符
\f	换页符
\y	八进制数 y 代表的字符
\xy	十六进制数 y 代表的字符

3．三引号的用法

三引号表示的字符串可以换行，因此可用来表示超长字符串或为程序添加较长的注释。

三引号的用法演示如下：

```
>>> s='''insert into addressList
(name , sex , phon , QQ , address)
values('王小明' , '男' , '13888997011' , '66735' , '北京市' )'''
>>>print s
insert into addressList
(name , sex , phon , QQ , address)
values('王小明' , '男' , '13888997011' , '66735' , '北京市' )
```

1.5.5　操作符和表达式

常用操作符如表1-3所示。

表1-3　　常用操作符

操 作 符	描　　述
x+y , x-y	加法／字符串合并，减法／集合差集