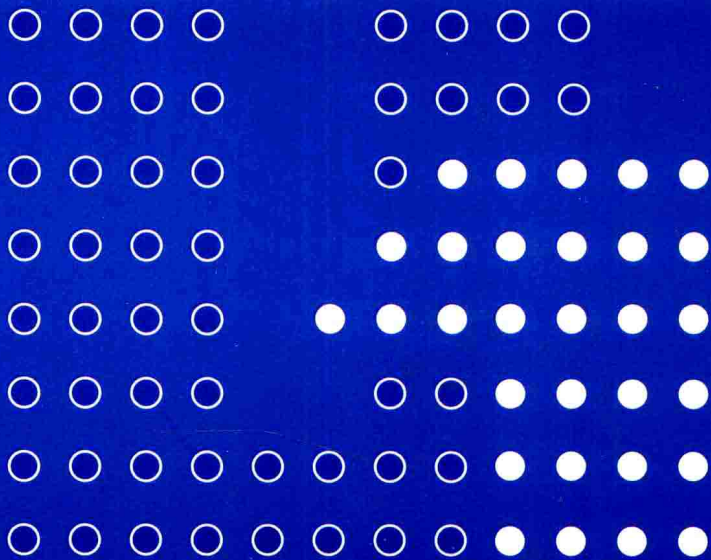




普通高等教育“十一五”国家级规划教材 计算机系列教材

Python程序设计基础



董付国 编著

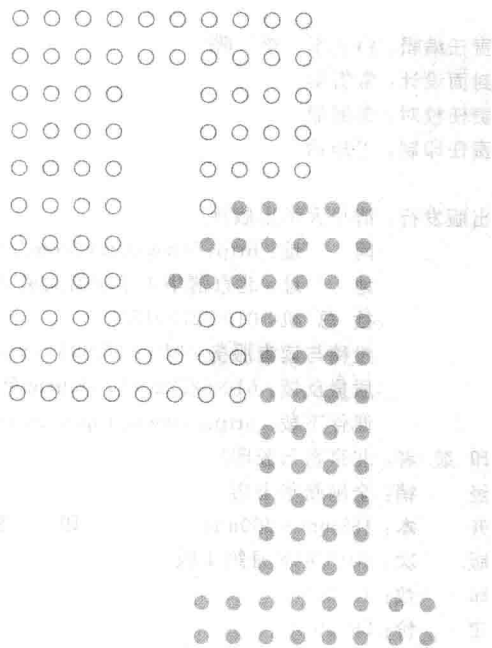


清华大学出版社

计算机系列教材

董付国 编著

Python程序设计基础



清华大学出版社

北京

内 容 简 介

全书共 9 章,主要内容组织如下:第 1 章介绍 Python 基本知识与概念;第 2 章讲解列表、元组、字典、集合等序列的常用方法和基本操作;第 3 章讲解 Python 选择结构、for 循环与 while 循环、break 与 continue 语句;第 4 章讲解字符串编码格式,字符串格式化、替换、分割、连接等基本操作方法,正则表达式语法、正则表达式对象、子模式与 match 对象,以及正则表达式模块 re 的应用;第 5 章讲解函数的定义与使用,关键参数、默认值参数、长度可变参数、变量作用域以及 lambda 表达式;第 6 章讲解类的定义、类成员与实例成员、私有成员与公有成员、特殊方法与运算符重载;第 7 章讲解文件操作基本知识,文本文件内容读取与写入,二进制文件操作与对象序列化,文件复制、移动、重命名、MD5 值计算、压缩与解压缩等文件级操作以及目录操作有关知识;第 8 章讲解 Python 异常类层次结构,不同形式的异常处理结构,以及如何调试 Python 程序;第 9 章讲解如何使用 wxPython 进行 GUI 编程,主要包括窗体、按钮、文本框、单选钮、复选框等控件以及各种对话框的运用。

本书对 Python 内部工作原理进行了一定程度的剖析,对 Python 2. x 和 Python 3. x 之间的区别进行了深入对比和分析,并适当介绍 Python 程序优化和安全编程的有关知识,可以满足不同层次读者的需要。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Python 程序设计基础/董付国编著. --北京:清华大学出版社,2015

计算机系列教材

ISBN 978-7-302-41058-4

I. ①P… II. ①董… 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(2015)第 173346 号

责任编辑:白立军 李 晔

封面设计:常雪影

责任校对:焦丽丽

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:16 字 数:372 千字

版 次:2015 年 8 月第 1 版 印 次:2015 年 8 月第 1 次印刷

印 数:1~2000

定 价:29.00 元

产品编号:065428-01

Python 由 Guido van Rossum 于 1989 年底研制,第一个公开发行人版本发行于 1991 年。Python 推出不久就迅速得到了各行业人士的青睐,经过二十多年的发展,已经渗透到计算机科学与技术、统计分析、移动终端开发、科学计算可视化、逆向工程与软件分析、图形图像处理、人工智能、游戏设计与策划、网站开发等几乎所有专业和领域。目前,Python 已经成为卡耐基-梅隆大学、麻省理工学院、加州大学伯克利分校、哈佛大学等国外很多大学计算机专业或非计算机专业的程序设计入门教学语言,国内也有不少学校的多个专业陆续开设了 Python 程序设计课程。Python 语言连续多年在 TIOBE 网站的编程语言排行榜上排名前十位,并于 2011 年 1 月被 TIOBE 网站评为 2010 年度语言。在 2014 年 12 月 IEEE Spectrum 推出的编程语言排行榜中,Python 更是取得了第 5 位的好成绩。

Python 是一门免费、开源的跨平台高级动态编程语言,支持命令式编程、函数式编程,完全支持面向对象程序设计,语法简洁清晰,并且拥有大量功能丰富而强大的标准库和扩展库以及众多狂热的支持者,可以帮助各领域的科研人员或策划师甚至管理人员快速实现和验证自己的思路与创意。Python 用户可以把主要精力放在业务逻辑的设计与实现上,而不用过多考虑语言本身的细节,开发效率非常高,其精妙之处令人击节赞叹。

Python 是一门快乐的语言,学习和使用 Python 也是一个快乐的过程。与 C 语言系列和 Java 等语言相比,Python 更加容易学习和使用,但这并不意味着可以非常轻松地掌握 Python。熟练掌握和运用 Python 仍需要通过大量的练习来锻炼自己的思维和熟悉 Python 编程模式,同时还需要经常关注 Python 社区优秀的代码以及各种扩展库的动态。当然,如果能够适当了解 Python 及其扩展库的内部工作原理,对于编写正确而优雅的 Python 程序也是有很大帮助的。

Python 是一门优雅的语言。Python 语法简洁清晰,并且提供了大量的内置对象和内置函数,编程模式非常符合人类的思维方法和习惯。在有些编程语言中需要编写大量代码才能实现的功能,在 Python 中仅需要调用内置函数或内置对象的方法即可实现。如果有其他程序设计语言的基础,那么在学习和使用 Python 的时候,一定不要将其他语言的编程习惯和风格带到 Python 中来,因为这不仅可能会使得代码变得非常冗长、烦琐,还可能会严重影响代码的效率。应该尽量尝试从最自然、最简洁的角度出发去思考和解决问题,这样才能写出更加优雅、更加 Pythonic 的代码。

本书内容组织

对于 Python 程序员来说,能够熟练运用各种扩展库毫无疑问是非常重要的,使用优秀、成熟的扩展库可以帮助我们快速实现自己的业务逻辑和创意。但是也必须清楚地认识到,Python 语言基础知识和基本数据结构的熟练掌握是理解和运用其他扩展库的必备条件之一。因此,本书把重点和主要篇幅放在 Python 编程基础知识的介绍上,通过大量案例介绍 Python 在实际开发中的应用,关于不同应用领域的扩展库可以参考附录 B,并结合自己的专业领域查阅相关文档。全书共 9 章,主要内容组织如下:

第 1 章 基础知识。介绍如何选择 Python 版本,Python 对象模型,数字、字符串等基本数据类型,运算符与表达式,内置函数,基本输入输出,Python 程序文件名,扩展库的管理与使用,Python 代码编写规范,等等。

第 2 章 Python 序列。讲解序列常用方法和基本操作,成员测试运算符,切片操作,列表基本操作与常用方法,列表推导式,元组与生成器推导式,序列解包,字典、集合基本操作和常用方法,以及如何使用 Python 基本数据类型实现栈、二叉树、有向图等复杂数据结构。

第 3 章 选择与循环。讲解 Python 选择结构,for 循环与 while 循环,带有 else 子句的循环结构,break 与 continue 语句,选择结构与循环结构的综合运用。

第 4 章 字符串与正则表达式。讲解字符串编码格式,字符串格式化、替换、分割、连接等基本操作方法,正则表达式语法、正则表达式对象、子模式与`.match`对象,以及 Python 正则表达式模块`re`的应用。

第 5 章 函数设计与使用。讲解函数的定义与使用,关键参数、默认值参数、长度可变参数等不同参数类型,全局变量与局部变量,参数传递时的序列解包,return 语句,lambda 表达式,等等。

第 6 章 面向对象程序设计。讲解类的定义与继承、self 与 cls 参数、类成员与实例成员、私有成员与公有成员、特殊方法与运算符重载等内容。

第 7 章 文件操作。讲解文件操作基本知识与 Python 文件对象,文本文件内容读取与写入,二进制文件操作与对象序列化,文件复制、移动、重命名、文件类型检测、MD5 值计算、压缩与解压缩等文件级操作以及目录操作有关知识。

第 8 章 异常处理结构与程序调试。讲解 Python 异常类层次结构与自定义异常类,

多种不同形式的异常处理结构,以及如何使用 IDLE 和 pdb 模块调试 Python 程序。

第 9 章 GUI 编程。讲解如何使用 wxPython 进行 GUI 编程,主要包括窗体、按钮、文本框、单选按钮、复选框、组合框、列表框、树形等控件以及各种对话框的运用。

本书最大特点是信息量大、知识点紧凑、案例丰富。全书没有多余的文字和软件安装截图,充分利用宝贵的篇幅来介绍和讲解尽可能多的知识点,可以说是物超所值。本书作者具有 15 年程序设计教学经验,讲授过汇编语言、C/C++/C#、Java、PHP、Python 等多门程序设计语言,并编写过大量的应用程序。在本书内容的组织和安排上,结合了多年教学与开发过程中积累的许多案例,并巧妙地糅合进了相应的章节。

本书对 Python 内部工作原理进行了一定程度的剖析,对 Python 2.x 和 Python 3.x 之间的区别进行了深入对比和分析,并适当介绍了 Python 程序优化和安全编程的有关知识,可以满足不同层次读者的需要。

本书适用读者

本书可以作为(但不限于):

- 数字媒体技术、软件工程、网络工程、信息安全、会计、经济、金融、心理学、统计以及其他非计算机专业本科或专科的程序设计教材。如果作为本科非计算机专业程序设计语言公共课或选修课教材,建议采用 64 学时或 48 学时边讲边练的教学模式。
- 具有一定 Python 基础的读者进阶学习资料。
- 打算利用业余时间学习一门快乐的程序设计语言并编写几个小程序来娱乐的读者首选学习资料。
- 少数对编程具有浓厚兴趣和天赋的中学生课外阅读资料。

教学资源

本书提供全套教学课件、源代码、课后习题答案与分析以及授课计划和学时分配表,配套资源可以登录清华大学出版社官方网站下载或与作者联系索取,作者 QQ 号码是 306467355,微信号是 Python_dfg,电子邮箱地址是 dongfuguo2005@126.com。

由于时间仓促,作者水平有限,书中难免出现错误,不足之处还请指正并通过作者联系方式进行反馈,作者将不定期在 QQ 空间和微信发布和更新勘误表。

感谢

首先感谢父母的养育之恩,在当年那么艰苦的条件下还坚决支持我读书,而没有让我像其他同龄的孩子一样辍学。感谢姐姐、姐夫多年来对我的爱护以及在老家对父母的照顾,感谢善良的弟弟、弟媳在老家对父母的照顾,正是有了你们,我才能在远离家乡的城市安心工作。感谢我的妻子在生活中对我的大力支持,也感谢懂事的小女儿在我工作的时候能够在旁边安静地读书而尽量不打扰我,并在定稿前和妈妈一起帮我阅读全文并检查出了几个错别字。

感谢每一位读者,感谢您在茫茫书海中选择了本书,并衷心祝愿您能够从本书中受益,学到您需要的知识!同时也期待每一位读者的热心反馈,随时欢迎您指出书中的不足!

本书的出版获 2014 年山东省普通高校应用型人才培养专业发展支持计划项目资助。我校专业共建合作伙伴——浪潮优派科技教育有限公司总裁邵长臣先生——审阅了全书,并提出了很多宝贵的意见,在此致以诚挚的谢意。本书在编写出版过程中也得到清华大学出版社的大力支持和帮助,在此表示衷心的感谢。

董付国

于山东烟台

2015 年 5 月

第 1 章 基础知识 /1

- 1.1 如何选择 Python 版本 /1
- 1.2 Python 安装与简单使用 /3
- 1.3 使用 pip 管理 Python 扩展库 /5
- 1.4 Python 基础知识 /6
 - 1.4.1 Python 对象模型 /6
 - 1.4.2 Python 变量 /6
 - 1.4.3 数字 /10
 - 1.4.4 字符串 /11
 - 1.4.5 运算符与表达式 /12
 - 1.4.6 常用内置函数 /15
 - 1.4.7 对象的删除 /19
 - 1.4.8 基本输入输出 /20
 - 1.4.9 模块导入与使用 /22
- 1.5 Python 代码编写规范 /24
- 1.6 Python 文件名 /26
- 1.7 Python 脚本的 `__name__` 属性 /27
- 1.8 编写自己的包 /27
- 1.9 Python 编程快速入门 /28
- 1.10 The Zen of Python /30
- 本章小结 /31
- 习题 /32

第 2 章 Python 序列 /33

- 2.1 列表 /33
 - 2.1.1 列表创建与删除 /34
 - 2.1.2 列表元素的增加 /36
 - 2.1.3 列表元素的删除 /40

2.1.4	列表元素访问与计数	/44
2.1.5	成员资格判断	/45
2.1.6	切片操作	/46
2.1.7	列表排序	/48
2.1.8	用于序列操作的常用内置函数	/49
2.1.9	列表推导式	/52
2.2	元组	/55
2.2.1	元组的创建与删除	/55
2.2.2	元组与列表的区别	/56
2.2.3	序列解包	/57
2.2.4	生成器推导式	/58
2.3	字典	/59
2.3.1	字典创建与删除	/59
2.3.2	字典元素的读取	/60
2.3.3	字典元素的添加与修改	/62
2.3.4	字典应用案例	/62
2.3.5	有序字典	/63
2.4	集合	/64
2.4.1	集合的创建与删除	/64
2.4.2	集合操作	/65
2.5	再谈内置方法 sorted()	/66
2.6	复杂数据结构	/68
2.6.1	堆	/68
2.6.2	队列	/69
2.6.3	栈	/72
2.6.4	链表	/74
2.6.5	二叉树	/75
2.6.6	有向图	/78
	本章小结	/79
	习题	/80

第 3 章 选择与循环	/81
3.1 条件表达式	/81
3.2 选择结构	/83
3.2.1 单分支选择结构	/83
3.2.2 双分支选择结构	/84
3.2.3 多分支选择结构	/85
3.2.4 选择结构的嵌套	/86
3.2.5 选择结构应用案例	/87
3.3 循环结构	/88
3.3.1 for 循环与 while 循环	/88
3.3.2 循环结构的优化	/90
3.4 break 和 continue 语句	/91
3.5 案例精选	/93
本章小结	/97
习题	/97
第 4 章 字符串与正则表达式	/99
4.1 字符串	/100
4.1.1 字符串格式化	/101
4.1.2 字符串常用方法	/103
4.1.3 字符串常量	/110
4.1.4 可变字符串	/111
4.2 正则表达式	/112
4.2.1 正则表达式元字符	/112
4.2.2 re 模块主要方法	/114
4.2.3 直接使用 re 模块方法	/115
4.2.4 使用正则表达式对象	/116
4.2.5 子模式与 match 对象	/118
4.2.6 正则表达式应用案例精选	/122
本章小结	/127

习题 /128

第 5 章 函数设计与使用 /129

5.1 函数定义与调用 /129

5.2 形参与实参 /131

5.3 参数类型 /132

5.3.1 默认值参数 /132

5.3.2 关键参数 /134

5.3.3 可变长度参数 /135

5.3.4 参数传递时的序列解包 /136

5.4 return 语句 /136

5.5 变量作用域 /137

5.6 lambda 表达式 /139

5.7 案例精选 /140

5.8 高级话题 /144

本章小结 /147

习题 /148

第 6 章 面向对象程序设计 /149

6.1 类的定义与使用 /149

6.1.1 类定义语法 /149

6.1.2 self 参数 /150

6.1.3 类成员与实例成员 /150

6.1.4 私有成员与公有成员 /151

6.2 方法 /153

6.3 属性 /155

6.3.1 Python 2.x 中的属性 /155

6.3.2 Python 3.x 中的属性 /157

6.4 特殊方法与运算符重载 /159

6.4.1 常用特殊方法 /159

6.4.2	案例精选	/160
6.5	继承机制	/165
	本章小结	/168
	习题	/168
第7章 文件操作 /169		
7.1	文件对象	/169
7.2	文本文件操作案例精选	/171
7.3	二进制文件操作案例精选	/177
7.3.1	使用 pickle 模块	/177
7.3.2	使用 struct 模块	/178
7.4	文件级操作	/179
7.4.1	os 与 os.path 模块	/179
7.4.2	shutil 模块	/181
7.5	目录操作	/182
7.6	高级话题	/185
	本章小结	/189
	习题	/189
第8章 异常处理结构与程序调试 /191		
8.1	基本概念	/191
8.2	Python 异常类与自定义异常	/192
8.3	Python 中的异常处理结构	/195
8.3.1	try...except 结构	/195
8.3.2	try...except...else 结构	/196
8.3.3	带有多个 except 的 try 结构	/197
8.3.4	try...except...finally 结构	/198
8.4	断言与上下文管理	/200
8.4.1	断言	/200
8.4.2	上下文管理	/201

- 8.5 用 sys 模块回溯最后的异常 /201
- 8.6 使用 IDLE 调试代码 /202
- 8.7 使用 pdb 模块调试程序 /204
 - 8.7.1 pdb 模块常用命令 /204
 - 8.7.2 使用 pdb 模块调试 Python 程序 /206

本章小结 /208

习题 /209

第 9 章 GUI 编程 /210

- 9.1 Frame /210
- 9.2 Controls /214
 - 9.2.1 Button、StaticText、TextCtrl /214
 - 9.2.2 Menu /216
 - 9.2.3 ToolBar、StatusBar /217
 - 9.2.4 对话框 /218
 - 9.2.5 RadioButton、CheckBox /219
 - 9.2.6 ComboBox /221
 - 9.2.7 ListBox /222
 - 9.2.8 TreeCtrl /224

9.3 Boa-constructor /228

本章小结 /228

习题 /229

附录 A 将 Python 程序转换为 exe 程序 /230

附录 B 常用 Python 扩展库简介 /232

- B.1 图形图像编程模块 /232
- B.2 游戏编程模块 /232
- B.3 语音识别模块 /233
- B.4 网络编程模块 /233

B.5	多线程编程模块	/234
B.6	数据库编程模块	/234
B.7	Pywin32	/234
B.8	ctypes	/235
B.9	科学计算与可视化模块	/236
B.10	软件分析插件	/237
B.11	其他常用模块	/237
附录 C 安卓平台的 Python 编程		/239
参考文献		/242

第1章 基础知识

Python 是一门跨平台、开源、免费的解释型高级动态编程语言,同时也支持伪编译,即将 Python 源程序转换为字节码来优化程序和提高运行速度,并且支持使用 py2exe 工具将 Python 程序转换为扩展名为 exe 的可执行程序,可以在没有安装 Python 解释器和相关依赖包的 Windows 平台上运行;Python 支持命令式编程、函数式编程,完全支持面向对象程序设计,语法简洁清晰,并且拥有大量的几乎支持所有领域应用开发的成熟扩展库;Python 就像胶水一样,可以把多种不同语言编写的程序融合到一起实现无缝拼接,更好地发挥不同语言和工具的优势,满足不同应用领域的需求。

1.1 如何选择 Python 版本

众所周知,Python 官方网站目前同时发行 Python 2. x 和 Python 3. x 两个不同系列的版本,并且互相之间不兼容,除了输入输出方式有所不同,很多内置函数的实现和使用方式也有较大的区别,Python 3. x 对 Python 2. x 的标准库也进行了一定程度的重新拆分和整合。在本书开始编写的时候,最新版本分别为 Python 2. 7. 8 和 Python 3. 4. 2,本书编写完成时最新版本分别为 Python 2. 7. 10 和 Python 3. 4. 3,并且已发布 Python 3. 5. 0 的第三个测试版。对于很多初级用户而言,最纠结的一个问题很可能是自己到底应该选择哪个版本,是选择 Python 2. x 还是 Python 3. x,是选择 Python 2. 7. x 还是 Python 2. 6. x 呢? 对于 Python 的版本演化历史,这里不多解释,需要说明的是,并不是数字越大表示版本越新,例如 Python 2. 7. 9 就比 Python 3. 2. 6 晚几个月发行,并且 Python 3. 2. 6 比 Python 3. 4. 1 也晚几个月,类似的情况还有很多。另外,虽然同系列版本中高版本比低版本更加完善和成熟,但这并不意味着最新的才是最合适的。很多扩展库的发行总是滞后于 Python 发行的版本,甚至目前还有很多扩展库不支持 Python 3. x。因此,在选择 Python 的时候,一定要先考虑清楚自己学习 Python 的目的是什么,打算做哪方面的开发,有哪些扩展库可用,这些扩展库最高支持哪个版本的 Python。这些问题全部确定以后,再做出自己的选择,这样才能事半功倍,而不至于把太多时间浪费在 Python 以及各种扩展库的反复安装和卸载上。同时还应该注意,当较新的 Python 版本推出之后,不要急于更新和替换已安装版本,而是应该在确定自己必须使用的扩展库也推出了较新版本之后再一起进行更新。

尽管如此,以目前来看 Python 3. x 毕竟是大势所趋,如果你暂时还没想到要做什么行业领域的应用开发,或者仅仅是为了尝试一种新的、好玩的语言,那么请毫不犹豫地选择 Python 3. x 系列的最高版本(目前正式发行版最高版本是 Python 3. 4. 3)。我们相信,越来越多的扩展库将会在短时间内推出支持 Python 3. x 的版本。

安装好 Python 以后,在“开始”菜单中选择 IDLE(Python GUI)命令,即可启动

Python 解释器并可以看到当前安装的 Python 版本号,如图 1-1 和图 1-2 所示。当然,如果你喜欢,也可以启动 Python(command line)来开始美妙的 Python 之旅。在 IDLE (Python GUI)和 Python(command line)两种界面中,都以三个大于号 >>> 作为提示符,可以在提示符后面输入要执行的语句。在本书所有章节给出的示例代码中,>>> 符号都不需要输入,仅表示该代码是在交互模式下运行,而不带有该提示符的代码则表示是以脚本程序的方式运行的。本书主要使用 IDLE(Python GUI)环境来介绍 Python 程序的开发与应用。

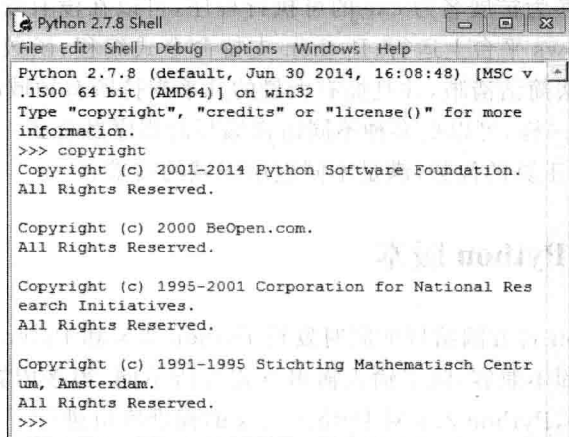


图 1-1 Python 2.7.8 主界面

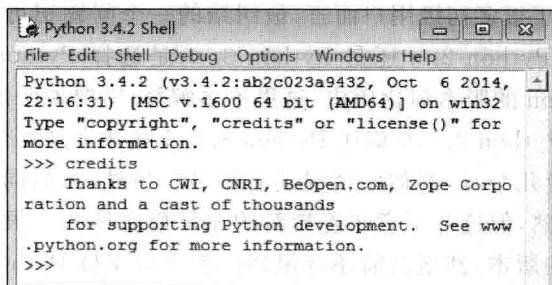


图 1-2 Python 3.4.2 主界面

除了在启动主界面上查看已安装的 Python 版本之外,还可以使用下面的命令随时进行查看。

```
>>> import sys
>>> sys.version
'3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:16:31) [MSC v.1600 64 bit (AMD64)]'
>>> sys.winver
'2.7'
>>> sys.version_info
sys.version_info(major=2, minor=7, micro=8, releaselevel='final', serial=0)
```


有时候可能需要同时安装多个不同的版本,例如,同时安装 Python 2.7.8 和 Python 3.4.2,并根据不同的开发需求在两个版本之间进行切换。多版本并存一般不影响在 IDLE 环境中直接运行程序,只需要启动相应版本的 IDLE 即可。在命令提示符环境中运行 Python 程序时,如果无法正确运行,可以尝试在调用 Python 主程序时指定其完整路径,或者通过修改系统 Path 变量来实现不同版本之间的切换。在 Windows 7 系统下修改系统 Path 变量的步骤如下:单击“开始”菜单,右击“计算机”并执行“属性”命令,在弹出的对话框中单击“高级系统设置”选项,切换至“高级”选项卡,单击“环境变量”按钮,然后修改系统 Path 变量中的 Python 安装路径,如图 1-3 所示。

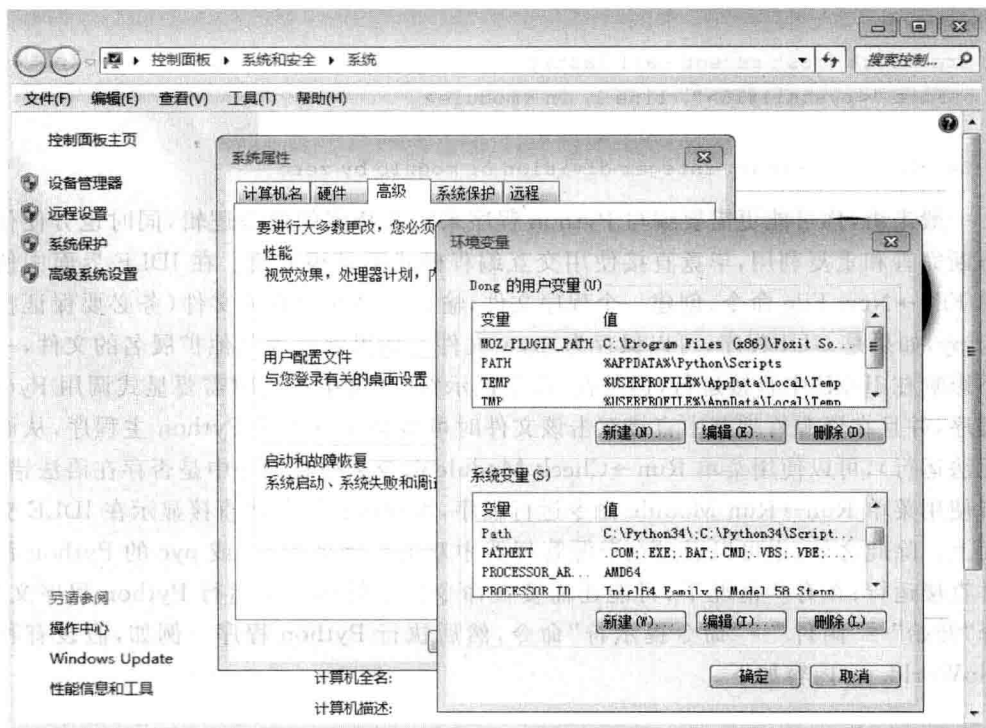


图 1-3 Windows 7 环境中系统 Path 变量修改方法

1.2 Python 安装与简单使用

为节约篇幅,这里不再详述 Python 的安装步骤,与大多数软件的安装并没什么明显的不同,打开 Python 官方主页 <https://www.python.org/> 后选择适合自己的版本下载并安装即可。如果使用的是 Linux 系统,例如 Ubuntu,那么很可能已经预装了某个版本的 Python,请根据需要进行升级。若未经特别说明,本书所有示例均在 Windows 7 平台上使用 Python 3.4.2 和 Python 2.7.8 进行开发和演示。

安装好以后,默认以 IDLE 为开发环境,当然也可以安装使用其他的开发环境,例如 PythonWin。本书均以 IDLE 为例,如果使用交互式编程模式,那么直接在 IDLE 提示符