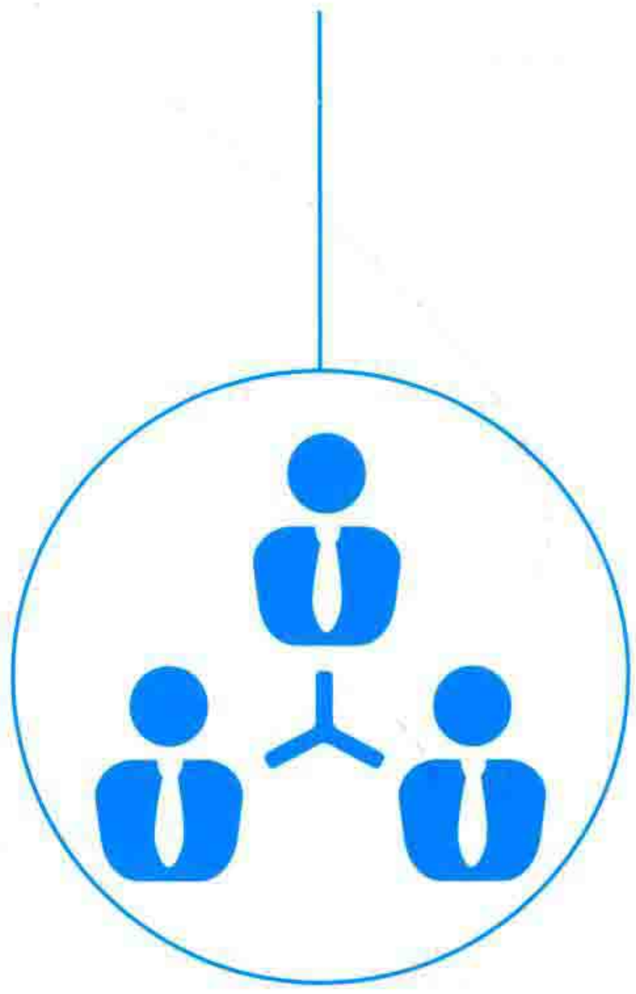
异步图书
www.epubit.com

程序设计领域学者**裘宗燕**教授潜心力作
程序员的 Python 入门和进阶指南

程序员学 Python

裘宗燕◎著



中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

程序员学Python

裘宗燕◎著

人民邮电出版社
北京

图书在版编目（CIP）数据

程序员学Python / 裘宗燕著. -- 北京 : 人民邮电出版社, 2018.8
ISBN 978-7-115-48262-4

Ⅰ. ①程… Ⅱ. ①裘… Ⅲ. ①软件工具－程序设计
Ⅳ. ①TP311.561

中国版本图书馆CIP数据核字(2018)第074189号

内 容 提 要

本书是面向学过编程、有一定编程经验的计算机专业人员，相关专业的大学生和教师的 Python 读物，也可作为以 Python 为第二门编程语言的高校课程教材或参考书。本书全面介绍了 Python 语言的各方面特征和应用技术，讨论了准确理解和正确使用 Python 语言所需要了解的深入概念和情况，还介绍了用 Python 开发较大型或较复杂程序时应该了解的一些高级功能，如程序的模块组织和导入系统，生成器、闭包和装饰器，基本的和高级的面向对象编程机制和技术，以及作为 Python 最新扩展的协程和异步编程等。

◆ 著　　裘宗燕
责任编辑　罗子超
责任印制　焦志炜
◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
涿州市京南印刷厂印刷
◆ 开本：800×1000　1/16
印张：26
字数：584 千字　　2018 年 8 月第 1 版
印数：1 – 2 400 册　　2018 年 8 月河北第 1 次印刷

定价：89.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

序

本书是针对程序员或其他学过用过至少一种编程语言、有一些编程经验的人们（如学过计算机基础课程的大学生）的 Python 编程著作。本书假定读者对于计算机的基础概念、程序和编程，都有一定的理解，但是没用过 Python。书中介绍了 Python 的基本特征，深入讨论了各方面的重要问题、高级机制和重要技术，目标是帮助读者深入理解 Python 语言，理解如何用好这门语言，理解如何用它写出真正能用的良好程序。

Python 是目前非常热门的一种编程语言，有关 Python 编程和应用的书籍，虽不能说是汗牛充栋、铺天盖地，也是林林总总、选择很多。本书与其他书籍有什么不同呢？基于上面提出的基本目标，本书的特点主要体现在如下几个方面：

根据对读者已有知识基础的考虑，书中对 Python 中与其他语言类似的基本编程机制的介绍相对精练，将更多篇幅集中于各种反映了 Python 特点的特征及相关编程和应用技术方面。例如，书中前两章详细讨论了函数定义的嵌套结构和作用域规则，丰富的形参/实参机制和形实参匹配规则，高阶函数的概念和应用，迭代器和可迭代对象的概念和定义，lambda 表达式（匿名函数）及其应用，标准组合对象的构造和使用，描述式的概念和应用等。书中还通过较大型的实例展示组合数据对象的应用和相关编程技术。

程序员学习 Python 不是为了写几个玩具程序，而是为了开发有用的系统。针对这种需求，本书深入讨论了许多与开发复杂和大型程序有关的问题，以及相关的 Python 特征和应用技术。书中内容包括程序的功能分解、信息局部化、模块化；Python 函数定义、类定义和模块机制的使用；Python 中面向对象机制的相关概念、结构和应用技术，以及一些特殊功能类的构造；程序的模块分解和管理，复杂模块结构的物理组织和导入技术等。

本书对 Python 语言中的各种高级机制和编程技术都有非常详细的介绍和讨论，如生成器函数的定义和使用，利用高阶函数的闭包技术，错误处理的概念和 Python 的异常处理功能及其应用，装饰器的概念、定义和应用，抽象基类、元类和各种高级面向对象机制和技术（属性操作、property 和描述器等），异步程序的概念和 Python 最新版本引进的协程等异步编程机制及其应用等。书中还给出了许多应用实例。正确运用这些结构和技术，可以提高程序的模块化、可读性和易维护性。如果读者希望用 Python 开发复杂的应用系统，准确理解和掌握这些高级特性，就更加重要了。

人们常说 Python 语言简单，编写简单程序时好像也确实如此。但实际上 Python 绝不简

单，它也是一种很复杂的语言，其功能特征非常丰富，能支持多种编程风格，在几乎所有方面都能深度定制。要想用好 Python，用它解决复杂问题，开发功能正确的、效率高的程序，需要很好地理解上面说明的许多高级概念和特征，还需要理解这门语言的内在性质。本书的另一个特点就是深入分析了 Python 语言的各方面语义细节和重要性质。有关讨论随着各种语言特征的介绍散布全书，也有些基本问题集中在第 3 章专门讨论，那里特别关注了由引用语义带来的共享问题，由函数嵌套结构和高阶函数带来的复杂语义问题等。Python 的强大功能也容易误用，容易编写出意义正确但效率低下，以致根本不能实用的程序。第 3 章特别讨论了效率问题，分析了一些情况，提出了一些设计原则和技术。

Python 语言带有一个很大的、不断增长的标准库，包含数百个程序包。一些程序包是通用的，与具体应用领域无关，如提供访问底层系统的 API，或支持程序开发与调试的功能；有些包是专用的，如支持互联网应用或图形用户界面等。有些书籍用很大篇幅介绍若干程序包，示例也是围绕一些程序包的应用进行讲解。本书的考虑有所不同，这里集中关注 Python 语言本身的功能和特性。对于标准库，书中只涉及少量程序包，如核心的包（如 sys、io、types 等）或最常用的包（如 math、random 等），以及与某些语言功能关系密切、必须讨论的包（如 asyncio 等）。作者认为，学习一门语言，首要的也是最重要的，是全面准确地掌握这门语言的核心概念和重要性质。由于读者已有编程和使用编程语言的经验，理解了 Python 的基本特征和相关技术后，应该可以查阅有关文档，自己弄清有关的标准库功能。

总而言之，本书是一本全面介绍 Python 语言各方面特征和编程技术的著作，其内容涵盖了 Python 核心语言的所有方面，讨论的内容足以支持读者使用 Python 去开发复杂的大型 Python 程序。当然，为了应对具体的应用领域，读者可能还应该考察一些针对其目标应用领域的标准库包，或者第三方程序包，以更方便地完成自己的工作。

由于讨论如此宽泛和丰富，本书的内容绝不简单，不是几天就能读完的简单教程。阅读本书需要一些时间，还需要编程实践。严肃地学习任何一种新编程语言时，情况也都差不多。由于内容比较全面，本书还可以作为 Python 程序员手头的参考手册，供人们在用 Python 做应用开发的时候参考。附录 A 是一个简明的 Python 手册，总结了 Python 语言的各方面特征，并索引到书中有关章节，供读者查阅。

在撰写本书的过程中，作者最主要的参考材料就是 Python 的标准文档（语言手册和标准库手册），还参考了 Mark Lutz 的两本著作（见最后的“推荐阅读书目”）和互联网上的一些材料、报告和讨论（如 stackoverflow 上的讨论），在此一并向相关材料和书籍的作者表示感谢。人民邮电出版社的陈冀康编辑认真审阅了本书的草稿，发现了一些错误和不足之处，作者对其认真工作表示感谢。此外，作者也希望得到读者的反馈意见和建议，先在这里表示感谢。本书将在 http://www.math.pku.edu.cn/teachers/qiuzy/books.htm 有一个页面，维持一个勘误表和一些相关信息，供读者查看。

裘宗燕

2018 年 3 月，北京

前言

Python 已经成为目前最流行的编程语言之一，在各种语言排行榜中位居前列。人们用 Python 自学编程，用它教大学里的第一门计算机科学课程。Python 也被广泛用在互联网应用、数据处理和科学计算领域，以及各种应用系统的开发中。

本前言首先简单介绍 Python 语言的一些基本情况，包括其发展和使用的情况，而后简单介绍 Python 语言系统的安装和使用。

Python 语言简介

Python 语言是 CWI（荷兰国家数学和计算机研究中心）的程序员 Guido van Rossum 从 1989 年开始开发的一种高级语言，他的初始目标是希望能更方便地管理 CWI 的 Amoeba 操作系统，后来该语言由于各方面的优点而逐渐流行。今天，Python 已经发展成世界上使用最广泛的编程语言之一，在全世界（包括中国）形成了巩固的用户社群。人们已经用 Python 开发了大量实际应用系统，也积累了许多基础资源。

Python 语言的发展和应用

Python 语言目前由 Python 软件基金会（Python Software Foundation，PSF）主导开发和管理。PSF 是一个非营利性的国际组织。Python 的官方网址为 python.org，在那里可以找到有关 Python 语言和系统开发的最新信息，还有许多资源信息和链接。

Python 语言的开发经历了许多版本。2000 年发布的 Python 2.0 表明该语言进入了一个新阶段，也是国际上较广泛地接受它的标志性事件。Python 3.0 于 2008 年年底发布，设计中整合了有关语言发展的许多成熟想法，对语言做了全面清理，修正了许多重要缺陷，使整个语言的概念体系更加清晰，各方面的结构更具有统一性。

目前，Python 的发展和使用还处于 2.0 版与 3.0 版并存的阶段。PSF 早已宣告 Python 2.7 是 Python 2 的最后版本，今后只做有限完善，不再做大的版本升级，开发和研究力量将集中到 Python 3.0 的开发。经过几年发展，Python 3.5 于 2015 年 9 月发布，Python 3.6 于 2016 年 12 月发布。有统计显示，目前，Python 2.0 和 3.0 在实际开发中的使用比例大约各占一半

（2016 年下半年的情况），后者的使用比例正在不断上升。有消息说 PSF 和各重要 Python 库的开发者都已确定，在 2020 年以后不再支持 Python 2。

由于这些情况，本书选择 Python 3.0 作为工作语言，以适应发展需要。书中所有实例（及所附代码）都在 3.5 或 3.6 版本的系统中开发和测试，但这些代码并不特定于这些版本（除个别专门说明的例外），大都能在各种 Python 3.0 版本的系统上运行。

Python 语言的特点

Python 的一个重要设计目标是让程序简单、清晰和优雅，坚持一套整齐划一的设计风格。Python 程序具有易写、易读、易维护的特点，受到广大程序员欢迎。这些特质也是导致 Python 的使用越来越广泛的原因。21 世纪以来，Python 已发展为世界上最受欢迎的编程语言之一，其使用非常广泛。国际上一些公司做过（或一直在做）各种编程语言使用情况的调查，统计结果中 Python 都位于前四五名之内。它还被 TIOBE 编程语言排行榜（最有影响力的语言排行榜之一）评为 2010 年的年度语言。

Python 被广泛认为是一种容易入门的语言。实际上，Python 语言机制的跨度比较大，从完成最简单计算的表达式开始，一直延伸到许多当前最先进的编程概念，如面向对象的程序设计、数据抽象、迭代器、异步编程等。这些情况有利于学习者在一个语言里逐步深入地学习许多编程概念和技术。Python 用正文缩进形式表现程序的结构，具有较好的可读性。

Python 是一种比较高级的编程语言。除了最基本的编程机制外，它还提供了使用方便的数据功能，可以很方便地组织和管理大批数据。Python 的所有编程机制和结构都围绕着**对象**的概念，程序里定义和操作的各种实体都是对象，不仅所有数据都是对象，函数和类等也是对象。它也能很好支持**面向对象编程**的理念和相关技术。

由于其基本设计的一些特点，Python 代码和部件比较容易重用，已开发的程序容易修改和扩充，有利于软件的升级改造，可以减轻软件开发者的工作负担，提高程序开发的效率。此外，Python 语言的设计也为开发大规模软件系统提供了很好支持。这些是许多 IT 公司乐于选择和使用 Python 作为其主要开发语言的重要原因。

在用 Python 开发程序时，可以采用交互式的执行方式，随时把代码发送给系统，立刻看到执行效果。这种方式使人更容易在编程中做各种试验，可以提高工作效率。一个 Python 程序文件（称为模块）的内容就是一系列简单或复杂的命令的序列。人们也把这样的语言称为**脚本语言**（script language），其程序就像一个工作脚本。

实际上，Python 并不是简单的脚本语言，而是一个能支持大规模软件开发的通用编程语言，其实现具有较高的执行效率。PSF 的 Python 系统带有一个很大的标准库，提供了很多在实际开发中非常有用的功能。此外，全世界的开发者已经为 Python 开发了面向各种应用领域的大量专用程序包，例如面向图形用户界面的设计和编程，面向网络应用、数值计算、数据统计和处理、图形图像处理、可视化等。针对所有重要应用领域，都可以找到相关的程序包，

大大方便了人们用 Python 开发领域应用软件和综合性软件的工作。

Python 语言和标准库的设计特别考虑了可扩充性，提供了丰富的接口和工具，使有经验的程序员比较容易使用其他语言，例如 C、C++、CPython（一种专门用于扩充 Python 的 C 语言工具）等编写 Python 模块，然后能像 Python 标准库包一样方便地使用。这种情况也使一些大公司把 Python 用作高级的**粘接语言**（glue language），用一些较低级的语言实现一批性能要求较高的完成具体工作任务的模块，而后用 Python 实现整个系统的高层控制和调度。这样做，既能获得很好的开发效率，也有利于修改和扩充。

Python 基金会提供最新版本的 Python 语言系统和基本开发环境，任何人都可以免费获取。该系统可以在各种主流计算机和软件平台上运行，包含了丰富的标准程序库和完整文档。此外，也存在另外一些商业的或非商业的 Python 系统可供选择。经过多年使用，全世界的 Python 开发者和使用者已经形成了一个活跃的专业社群，活跃在世界各地（包括中国），探讨、交流学习和使用 Python 的经验。互联网有很多与 Python 有关的信息，有许多 Python 讨论组。这些都促进了 Python 语言的学习和传播。

当然，Python 也不是完美无缺的（完美的语言并不存在），也有些缺点。还有一些使用需要注意的问题。后面讨论中也会提到一些这方面的情况。

Python 的应用情况

Python 已经有了非常广泛的实际使用。国际上的许多知名 IT 公司和机构以其作为主要开发语言，如美国的 Google、Yahoo!、Dropbox 等大公司，CERN（欧洲原子能研究中心）、NASA（美国国家航空航天局）等重要机构，还有大量较小的公司和机构。国内企业的应用正在发展，有较大影响的豆瓣网就是用 Python 开发的。

此外，全世界 Python 社区一直在努力，开发了许多适合各领域需要的 Python 包，这些工作也大大推动了 Python 的应用。例如，Python 的科学计算专用扩展库，包括 NumPy（高效的数组数据处理）、SciPy（高性能数值运算）和 matplotlib（数学绘图库）等。大量面向数据处理和计算的开源包也为 Python 使用提供了接口（可作为库调用，支持 Python 应用开发），如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK 等。Python 语言与这些库结合，构成的开发环境很适合工程技术人员和科研人员处理实验数据、制作图表，以及开发科学和工程计算方面的应用程序。在应用系统领域，Python 社群开发了一批支持网络应用开发的 Python 库和其他方面的库及编程框架，这些工作和后续经验的积累，已经使 Python 成为目前使用最多的应用系统开发语言之一。

Python 还被广泛用于复杂的和大规模的数据处理，成为目前人们在研究、开发大数据和人工智能等热门发展方向时使用最多的语言之一。

Python 语言参考材料

Python 软件基金会通过 python.org 提供了很多与 Python 语言和编程有关的材料，其 Python 系统（称为 CPython 实现，详见下一节）包含一套文档，主要内容包括如下。

- The Python Tutorial（Python 教程），其内容是 Python 各方面基本情况的介绍，基本使用规则，以及一些简单的程序示例。
- The Python Language Reference（Python 语言手册），详细介绍 Python 语言的整体情况和各种特征。学习和使用中应经常查阅这个手册。
- The Python Standard Library（Python 标准库手册），介绍 Python 的所有内置常量、内置函数和内置类型，以及标准库的一大批程序包。这些程序包提供了许多重要功能，包括一些系统功能，以及许多支持应用开发的功能。
- 其他内容，包括 CPython 系统的情况，典型编程问题的常见处理方式（HOWTO），一些常见问题（FAQ）、术语和解释等。

近年来，由于 Python 语言的发展和普及，国内外出版了不少有关 Python 编程的书籍。国外出版的许多相关书籍有中文译本，也有一些国内作者撰写的书籍。本书最后的“推荐阅读书目”列出了几本，供读者参考。

『 Python 系统和编程环境 』

本节简单介绍 PSF 主导开发的 CPython 系统及其附带的编程环境。对初学者而言，使用这个系统及其所带的程序包就足够了。一些开源社团或软件厂商开发了更强大的开发环境，利用 CPython 的功能或其他 Python 实现。鉴于本书的基本设想和目标读者群，这里不准备涉及任何超出 CPython 系统的内容。有兴趣的读者可以自己学习。

Python 是一种高级语言，具有易读易用的形式。为了运行 Python 程序，需要有一个 Python 解释器来填补 Python 源程序和计算机之间的鸿沟。PSF 的 Python 系统（CPython，以下说 Python 系统时专指这个系统）的主要部分就是一个解释器[①]。

下面以 Windows 系统中安装 Python 的情况为例，在其他系统里的安装情况类似。从 PSF 网站或其他地方下载 Python 安装文件，在所用计算机环境成功安装后，通常可以看到快捷启动方式。Python 系统各部分的安装位置、系统的启动方式、启动后窗口显示的情况，在不同环境里可能有些不同，但在功能上没有本质差别。

以命令行方式启动 Python 解释器，启动后的情况如图 0.1 所示。解释器显示版本等信息，最后一行是**提示符**串>>>，可以在这里输入要求执行的命令（程序）。

---

① Python 官网介绍了一些其他 Python 系统，本书不涉及。

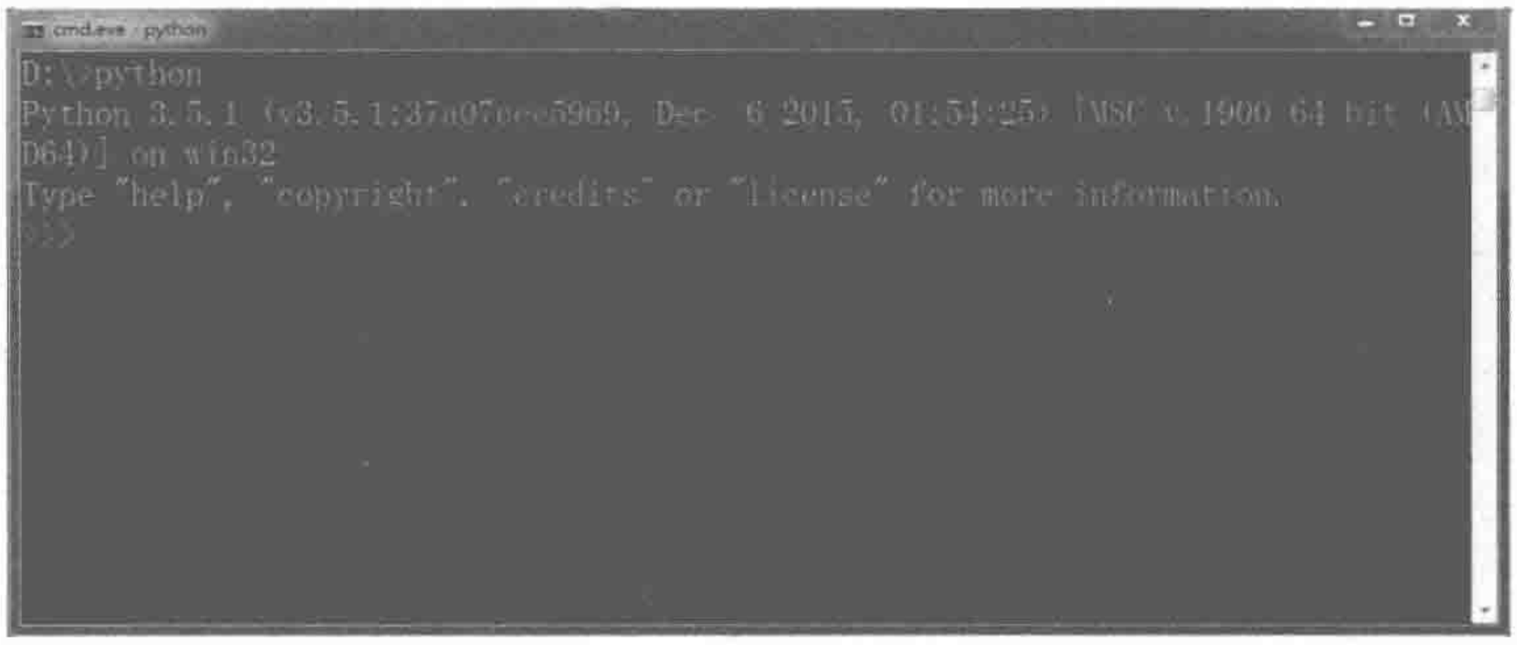


图 0.1　启动 Python 解释器后显示的命令行窗口

Python 解释器采用解释方式工作。一旦得到一个完整的程序单元，它就执行该单元并输出结果，然后重复。后面还会介绍解释器工作方式的一些细节。

CPython 提供了一个程序开发环境 IDLE，使程序员可以方便地编辑程序并随时运行。启动 IDLE 将看到一个窗口，顶部有标准的菜单条。图 0.2 显示了 IDLE 的解释执行窗口的一个情况[①]，可以看到解释器的提示符。输入一个程序单元（表达式或语句）后换行，解释器就会执行它并显示结果。这里显示的是执行 3 个表达式后的情况。第一个表达式要求计算 1 的值，解释器给出 1；第二个表达式要求计算 1+2；第 3 个表达式要求计算 2 的 1000 次幂，得到的大整数输出了几行。

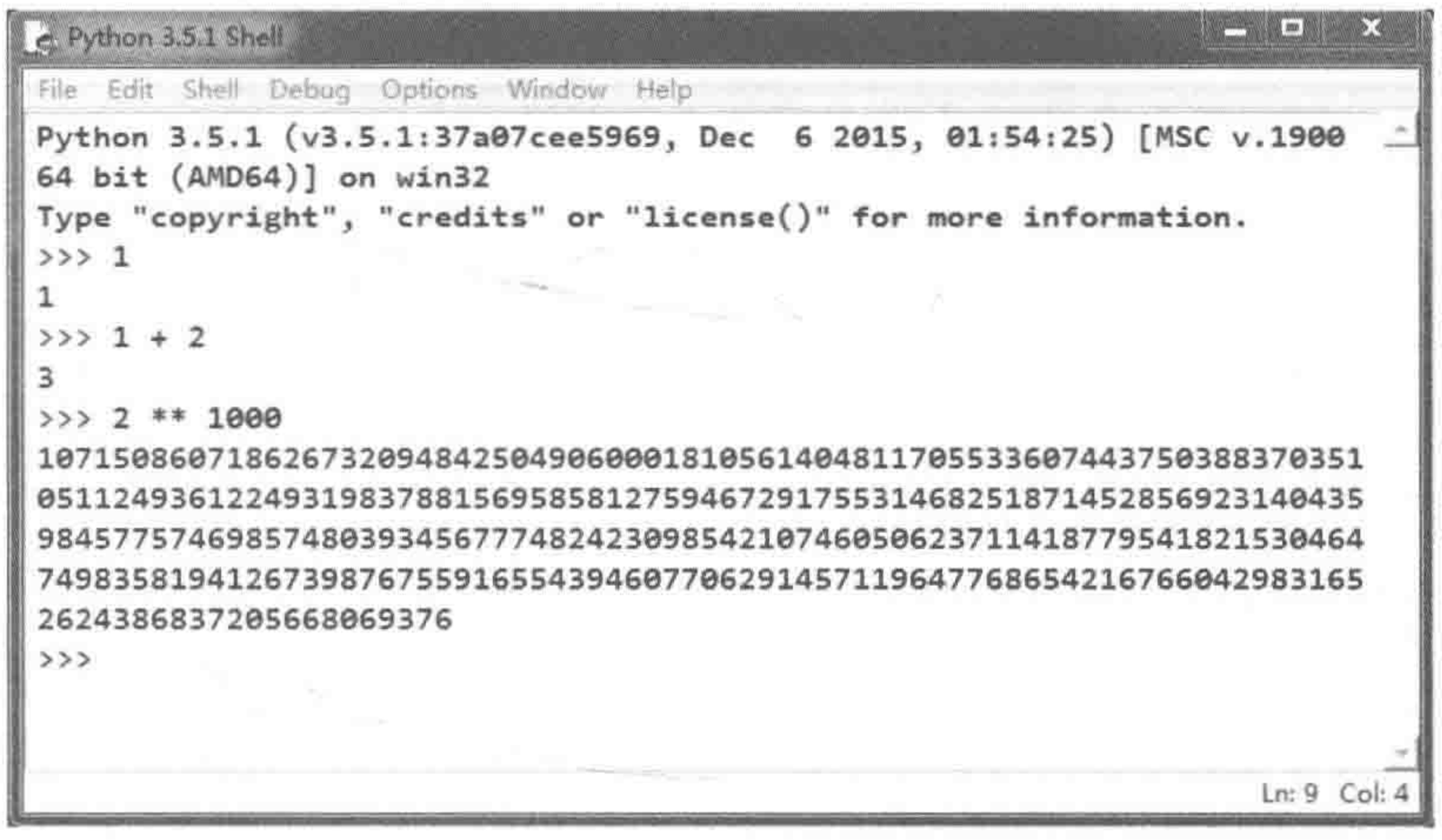


图 0.2　启动 IDLE 执行窗口（Shell）的情况

IDLE 的另一种窗口是编辑器，如图 0.3 所示。在这里编辑的程序可以随时运行。一个代码单元称为一个**模块**，执行前需要保存为文件。Python 术语中**模块**和**程序文件**基本是同义词，程序文件应该以 py 作为扩展名。

---

① 通过 Options 菜单可以选择打开的是执行窗口或编辑窗口，默认为打开执行窗口。

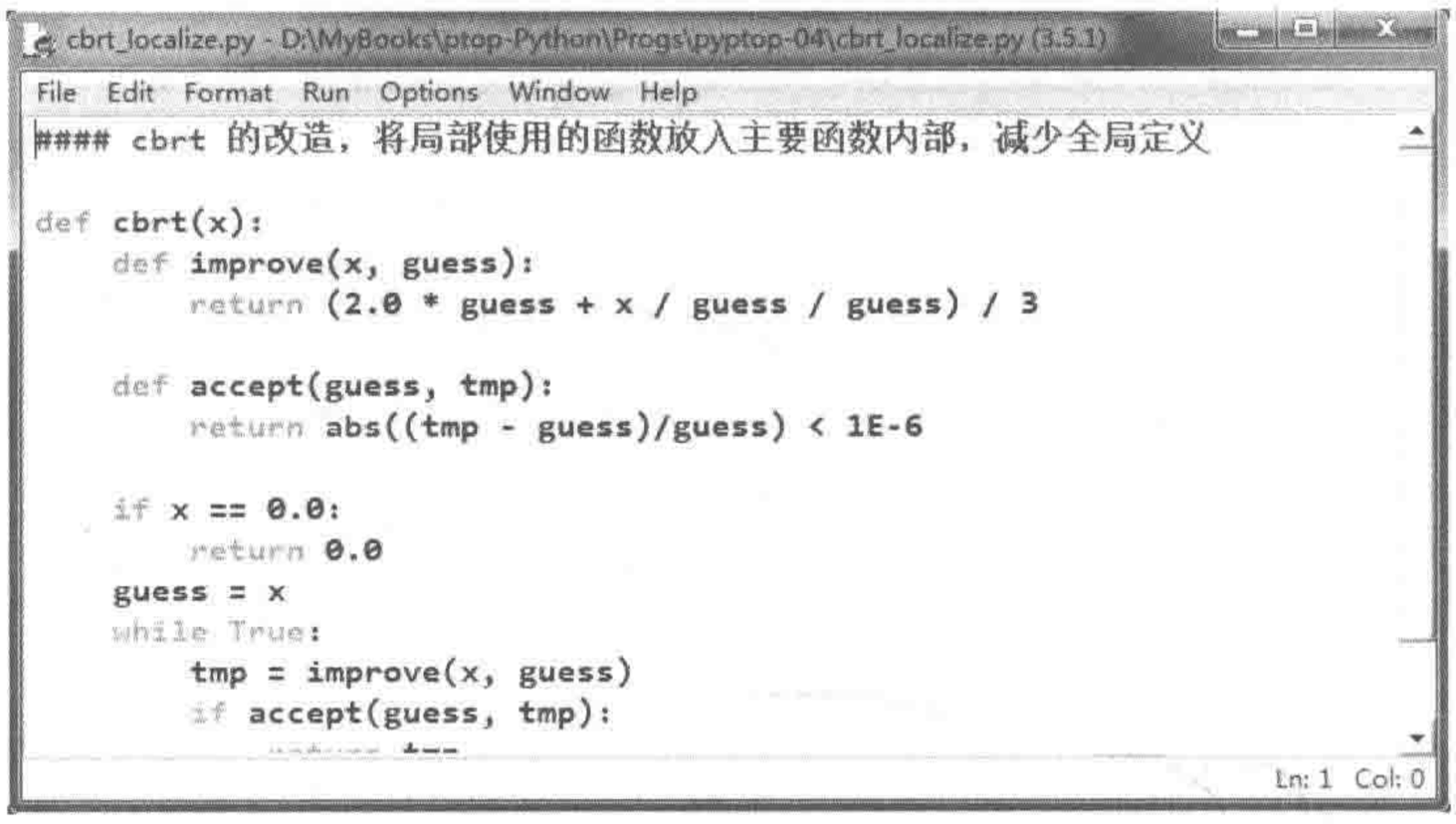


图 0.3　IDLE 编辑器窗口

IDLE 窗口支持常规的编辑命令。与执行窗口相比，这里多了 Format 和 Run 两个菜单。Format 里的命令用于修改被编辑程序的格式。Run 菜单用于启动模块执行，包括启动（或激活）关联执行窗口，调用解释器检查代码的语法，从空环境开始执行所编代码。运行时，解释器逐一执行其中语句，代码的标准输入和输出（常见的是用标准函数 input 和 print）通过关联的执行窗口实现。

IDLE 的执行窗口还有一个 Debug 菜单，其中命令服务于程序调试，需要与编辑窗口配合使用。有关功能将在“附录 C”介绍。此外，运行 IDLE 时按 F1 功能键，或者通过 Help 菜单的相应选项，都能打开 Python 系统自带的语言文档。

用 IDLE（或其他编辑器）开发的 Python 程序（模块）可以脱离编辑器，直接交给 Python 解释器执行。假设要执行的文件是 prog.py，只需在命令行窗口中键入：

```
python prog.py
```

就能启动 Python 执行该程序了［假设 Python 解释器已在操作系统（OS）的命令路径上］。

IDLE 是一个简单的开发环境，在开发复杂的应用系统时可以考虑用其他开发环境。有些通用开发环境带有配合使用 CPython 的插件，例如 Eclipse，可以在安装插件后用于 Python 程序开发。JetBrains 公司的 PyCharm 是目前比较流行的一个专业开发环境，提供了很好的开发支持。由于 Python 程序文件的内容就是普通文本，完全可以用任何文本编辑器编辑开发。CPython 的标准库还提供了一些支持调试的包。

前面说过，CPython 系统带有一个标准库，包含一大批程序包，系统文档中包含了标准库包的文档。标准库包的情况丰富多彩，有些包提供一些基础功能，如数学函数、文件操作、文件输入输出、随机数生成等。另一些包提供通用的编程服务功能，例如字符串处理、正则表达式、数据持久性、图形用户界面编程、并发编程、程序源文件组织等。一些包提供了某些特殊功能，如支持 Web 应用程序、媒体处理、加密解密等；还有一些支持程序开发、调试

等。如果开发中需要某些功能，但语言没提供，可以到标准库中找找。

本书主要关注 Python 语言本身的编程问题，下面的讨论将不涉及工具的使用。本书也不准备作为标准库的使用手册，对标准库的介绍将限于书中讨论 Python 语言基本功能和编程技术的需要。读者可以查看 CPython 的自带文档或其他材料，找到更多信息。

除了基本的 Python 系统和标准库程序包，一些个人、组织或公司也开发了许多第三方库，或对一些有用的库做了 Python 定制。有些库已被广泛使用，如支持图形用户界面开发的 wxPython 和 PyQt。CPython 推荐用自带的库安装工具 pip 安装其他库和程序包，Python 参考手册中有说明，另见本书 5.1 节。

本书结构和各章简介

本书假定读者学过某种编程语言，例如 C 语言等，有一定的编程经验，对程序语言的基本概念有些理解，对如何针对问题去开发程序有一定的认识。本书希望帮助这类读者学习 Python 语言和程序设计。基于上述基本假设，对于 Python 中最基本的和常规的特征，本书将只给出简短介绍，不过多解释，而把注意力集中到 Python 比较特殊的方面，讨论各种重要语言特征和相关的重要技术，书中还仔细地解释了 Python 的许多深入问题。总之，我们的目的是帮助读者深入理解 Python，理解怎样用它开发正确而高效的程序，如何利用其优势组织程序结构，使之更清晰，更容易理解，而且容易修改、维护和扩充。

本书比较全面地介绍了 Python 语言的各方面机制，各章的基本内容如下。

第 1 章介绍 Python 语言的基本编程特征，包括类型和表达式、基本类型、基本操作和控制结构、函数定义和输入输出等。本章也讨论了一些具有 Python 特色的概念和问题，如高阶函数、lambda 表达式、函数的嵌套组织等。这里还特别讨论了函数定义和使用的相互配合、变量的作用域问题，以及一些特殊的函数参数机制。

第 2 章讨论 Python 的数据功能。首先介绍了序列的概念和各种标准的序列类型（表、元组）及其操作，还介绍了字典和集合类型的性质和操作、字符串操作和格式化、文件的概念和使用等。这里还讨论了计算机信息处理的一些重要概念，包括数据持久性问题和 Python 系统支持数据持久性的 pickle 标准库包。

第 3 章包含两部分内容，主要是讨论了一些与 Python 的基本性质有关的问题，另外还介绍了若干不适合放在前两章的编程特征和相关技术。为了支持方便灵活的数据对象创建和自动存储管理，Python 的变量（和对象属性）都采用引用语义，变量的值是独立存在的对象，这种情况与 C 语言完全不同，带来了复杂的对象共享问题。另一方面，Python 支持函数的嵌套定义和高阶函数，还支持生成器、迭代器等重要编程概念。这些功能都非常有用，但也蕴涵着较为复杂的语义问题。本章详细讨论了这些语义问题及其对编程的各种影响，以帮助读者准确理解 Python 程序中各种操作的意义，理解应如何正确使用这些操作。此外，Python

使用方便，但也容易写出看起来简单但却极端低效的程序。3.5 节专门讨论了 Python 程序的效率问题，特别是与复杂数据对象的构造和使用有关的问题，提出了一些实现高效程序的准则和技术。此外，本章还讨论了生成器函数的定义，通过高阶函数实现的功能强大的闭包技术，以及程序中的错误处理和 Python 异常处理机制的使用。

第 4 章集中关注 Python 的面向对象特征和相关编程技术。人们说 Python 是一种面向对象的语言，最重要的就是它有一套面向对象编程的特征，支持重要的面向对象编程技术。实际上，Python 的面向对象特征与 C++或 Java 都不同，有关机制完全是动态的，并允许深度定制。本章主要关注如何按比较规范的方式做面向对象编程，讨论了类定义的方法、几类不同方法的定义技术、实例的生成和使用、通过继承定义派生类的技术，以及抽象基类的概念和基于接口的编程技术等。这里还介绍了几种重要特殊类的定义，包括迭代器类、容器类、上下文管理器类等，以及 Python 中特殊方法名的概念和使用等。

第 5 章讨论了一些与开发大型和复杂 Python 程序有关的问题，以及一些高级的 Python 编程机制和技术。这里首先总结了 Python 模块和程序的概念，介绍了一些与程序有关的基本问题，而后详细讨论了 Python 的导入系统和基于它的程序组织技术，还介绍了动态编译的概念等。5.2 节专门讨论装饰器的概念和技术，这个概念也是 Python 中许多机制的基础（例如类定义中的类方法和静态方法）。在这一章里，我们还介绍了与面向对象编程有关的一些高级概念和技术，包括类的创建和元类、属性的管理和一些相关机制，还有最基础的描述器机制和相关应用技术。本章最后一节介绍了 Python 最新和最重要的发展：协程和它所支持的异步编程概念和技术，以及另一些相关的异步特征，如异步迭代器、异步循环、异步生成器、异步描述式等。协程和其他异步编程机制是 Python 3.5 引入的新特征，Python 3.6 又做了一些重要扩充。基于这种机制，我们可以开发出一类超轻量级的并发程序，满足许多复杂应用的需要，特别是与互联网有关的应用等。

本书中每章都有一节“总结和补遗”，其中通常会介绍一些与该章内容有关的语言细节，也对该章里讨论的主要内容做一点简单总结。

本书最后有 4 个附录：“附录 A”是 Python 语言的一个浓缩手册，其中罗列了该语言的各方面特征，并给出了与之相关的章节索引；“附录 B”列出了 Python 的所有标准函数；“附录 C”简单介绍了 IDLE 开发环境，特别是它所支持的调试功能；“附录 D”列出了本书中使用（并有所介绍）的几个标准库包。最后给出了推荐阅读的相关图书。

资源与支持

本书由异步社区出品，社区（https://www.epubit.com/）为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

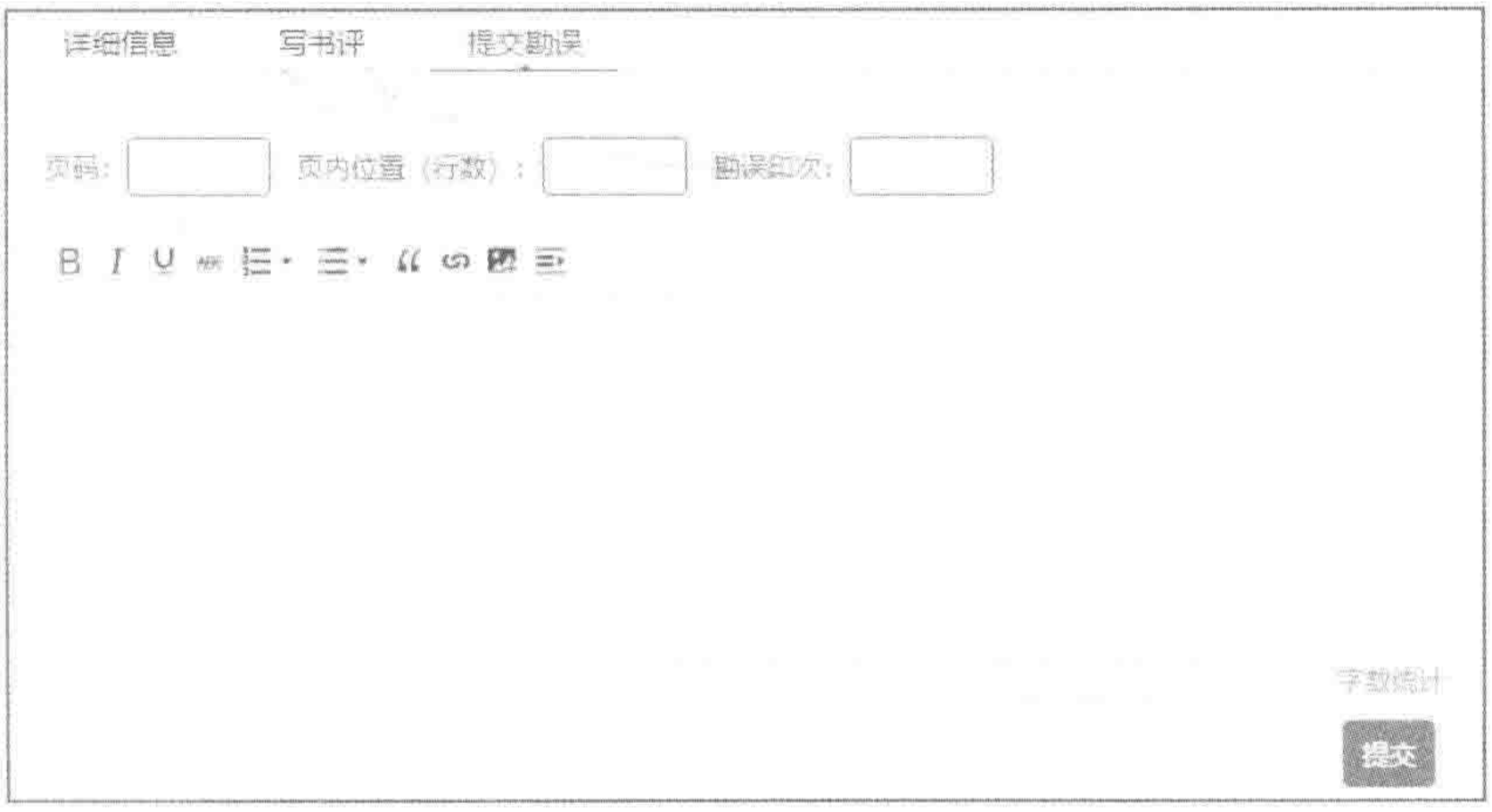
- 本书源代码。

要获得以上配套资源，请在异步社区本书页面中点击 配套资源，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。

与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 https://www.epubit.com。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。

异步社区

微信服务号

目录

第1章 Python 基础 …… 1
1.1 表达式和计算 …… 1
1.1.1 数值计算 …… 1
1.1.2 标准函数和数学函数包 …… 5
1.1.3 字符串 …… 7
1.2 变量和赋值 …… 10
1.2.1 名字、变量和赋值 …… 10
1.2.2 简单脚本程序 …… 12
1.2.3 若干情况 …… 13
1.3 逻辑和控制 …… 14
1.3.1 条件判断和条件语句 …… 15
1.3.2 循环语句 …… 18
1.4 定义函数 …… 20
1.4.1 计算的抽象：函数 …… 21
1.4.2 递归定义的函数 …… 25
1.4.3 比较复杂的递归问题 …… 32
1.5 函数定义的若干问题 …… 34
1.5.1 函数的意义 …… 34
1.5.2 函数分解：定义和调用 …… 36
1.5.3 程序框架和函数的函数参数 …… 40
1.5.4 匿名函数和 lambda 表达式 …… 44
1.5.5 作用域，嵌套的函数定义 … 48
1.5.6 带默认值形参和关键字实参 …… 53
1.6 总结和补遗 …… 55
1.6.1 整数的位运算 …… 55
1.6.2 基本字符集和一些词法规则 …… 56
1.6.3 循环语句的 else 段 …… 57
1.6.4 总结 …… 58
第2章 数据的构造和组织 …… 60
2.1 表和元组 …… 60
2.1.1 表（list） …… 60
2.1.2 表的使用和处理 …… 64
2.1.3 元组（tuple） …… 71
2.1.4 有理数程序包 …… 75
2.2 序列和序列操作 …… 79
2.2.1 序列和序列操作 …… 79
2.2.2 描述式 …… 83
2.2.3 一些程序实例 …… 86
2.2.4 几个序列类型 …… 89
2.3 字符串和格式化 …… 91
2.3.1 字符串操作 …… 91
2.3.2 字符串的格式化 …… 95
2.4 文件 …… 99
2.4.1 文件和输入/输出 …… 99
2.4.2 Python 的文件功能 …… 99
2.4.3 文件处理程序实例 …… 104
2.5 字典（dict） …… 106
2.5.1 概念和操作 …… 107
2.5.2 字典的应用实例 …… 109
2.5.3 字典与函数参数 …… 111

2.6 集合（set 和 frozenset）……112
2.6.1 概念和构造……112
2.6.2 集合操作……114
2.7 程序和数据……116
2.7.1 文本处理……117
2.7.2 数据记录和信息管理…… 122
2.7.3 数据持久性…… 127
2.8 总结和补遗…… 129
2.8.1 函数形参和实参…… 129
2.8.2 拆分与组合对象描述…… 130
2.8.3 总结…… 131
第 3 章 深入理解 Python…… 133
3.1 基本语义问题…… 133
3.1.1 变量和对象…… 133
3.1.2 函数和参数的语义…… 141
3.1.3 逻辑判断…… 144
3.1.4 几个问题…… 149
3.2 程序的语义实现…… 152
3.2.1 环境和状态…… 152
3.2.2 程序执行中的环境和状态变化 …… 155
3.2.3 函数定义结构和函数调用 …… 159
3.2.4 函数的若干问题…… 160
3.3 生成器函数和闭包…… 163
3.3.1 提取文件数据的函数…… 163
3.3.2 生成器函数…… 166
3.3.3 闭包技术和原理…… 170
3.3.4 编程实例…… 175
3.4 异常和异常处理…… 178
3.4.1 运行中的错误…… 178
3.4.2 Python 异常处理和 try 结构 …… 180
3.4.3 异常处理的结构和技术… 183
3.4.4 预定义异常…… 187
3.4.5 异常作为控制机制…… 189
3.5 效率…… 192
3.5.1 基础…… 192
3.5.2 一个例子…… 198
3.5.3 标准组合类型的实现和操作效率…… 199
3.6 总结和补遗…… 204
3.6.1 异常处理机制补遗…… 204
3.6.2 生成器函数进阶…… 206
3.6.3 总结…… 210
第 4 章 面向对象编程…… 213
4.1 数据抽象、类和自定义类型…… 213
4.2 Python 的类和对象…… 215
4.2.1 类的定义和使用…… 215
4.2.2 几个问题…… 221
4.2.3 简单实例…… 225
4.2.4 Python 类、对象和方法…229
4.3 继承…… 230
4.3.1 继承、基类和派生类…… 230
4.3.2 几个简单实例…… 237
4.3.3 多继承…… 241
4.3.4 异常和类…… 244
4.4 特殊方法名和特殊的类…… 245
4.4.1 容器类和迭代器…… 246
4.4.2 上下文管理…… 248
4.4.3 一些特殊方法名和标准函数…… 251
4.5 实例：链接表…… 255
4.5.1 基本考虑…… 255
4.5.2 简单单链表…… 257
4.5.3 带尾结点指针的单链表…264
4.5.4 双链表…… 266
4.5.5 讨论…… 269
4.6 总结和补遗…… 269
4.6.1 对象的定义和使用…… 269
4.6.2 面向对象的技术和方法…… 273