

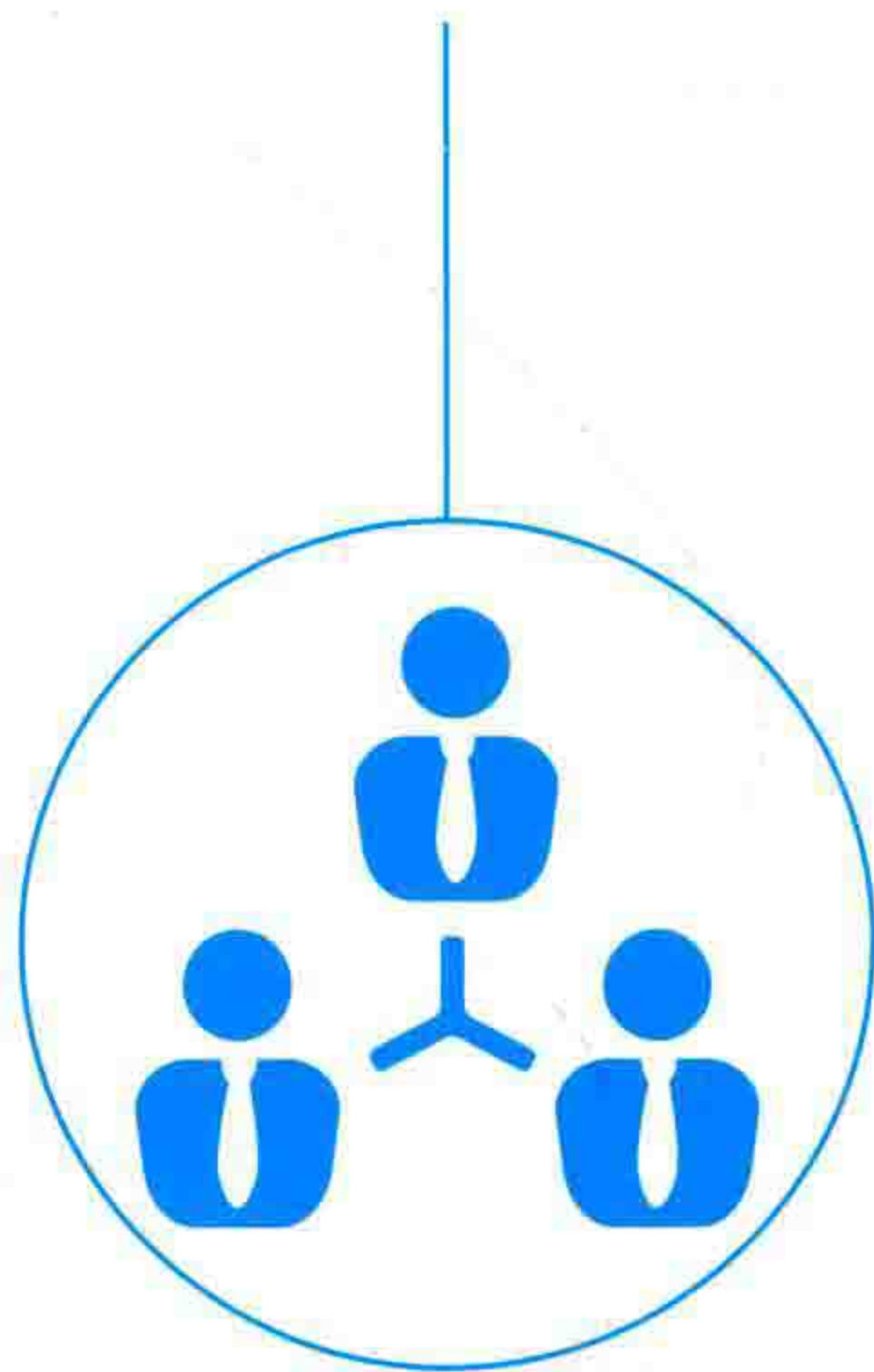


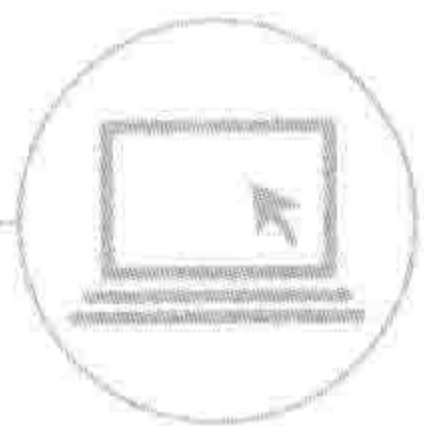
程序员学



Python

裘宗燕◎著





程序员学



Python

裘宗燕◎著

人民邮电出版社

北京

图书在版编目 (C I P) 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

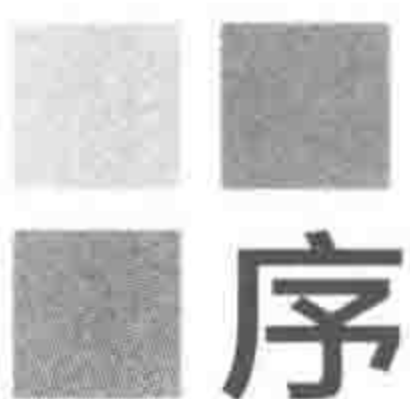
内 容 提 要

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
印数: 1-2 400 册
- 2018 年 8 月第 1 版
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 最新版本引入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 `>>>`，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。

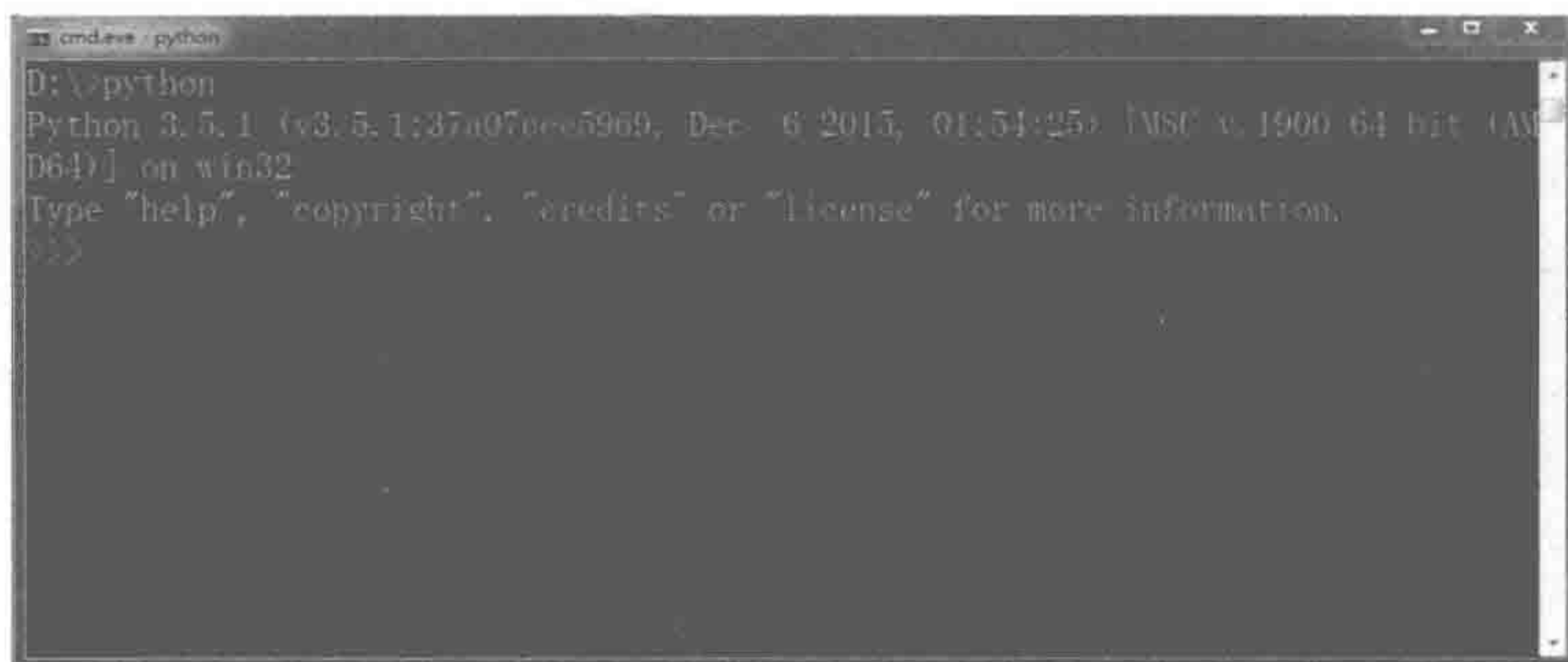


图 0.1 启动 Python 解释器后显示的命令行窗口

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。

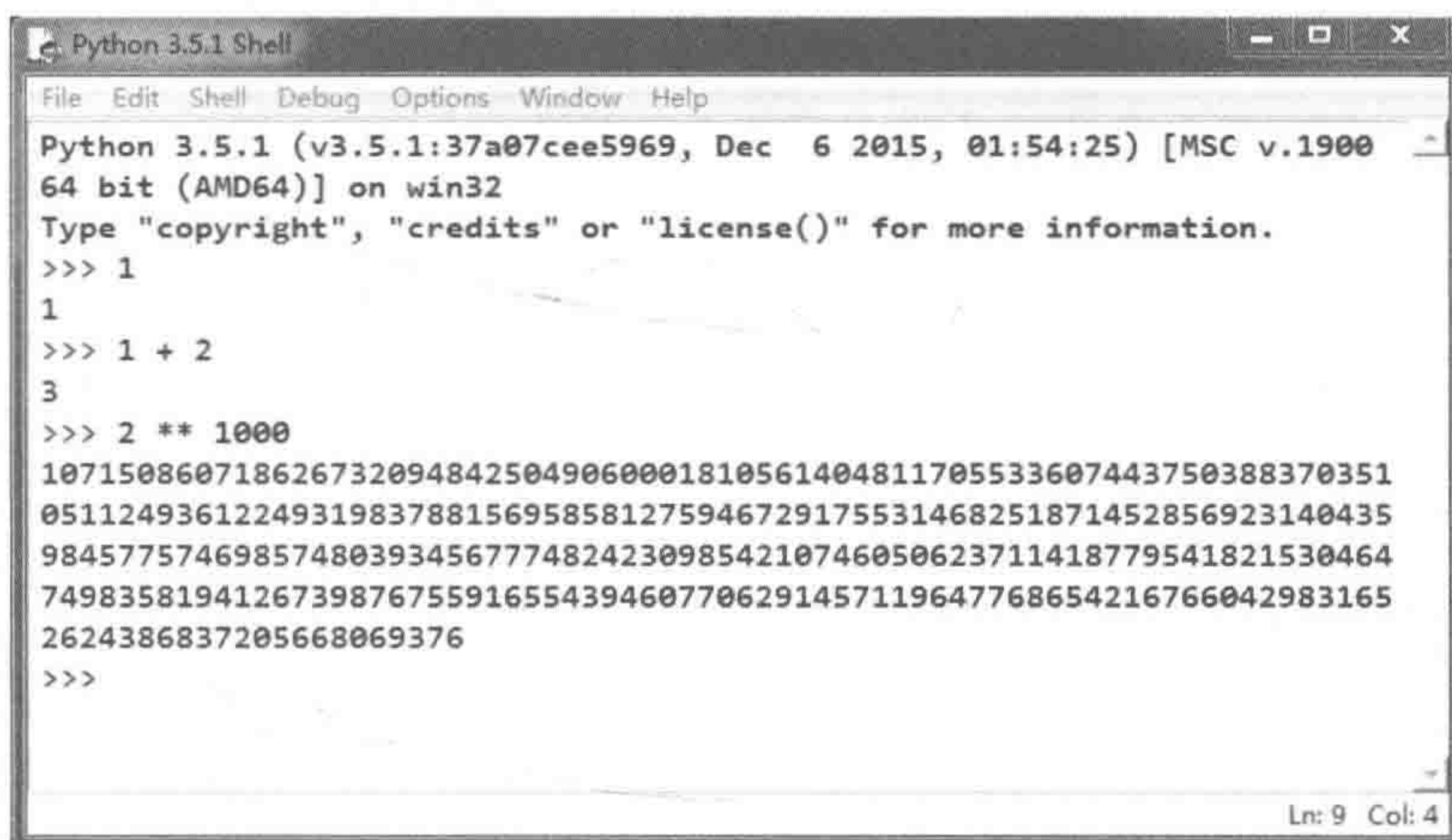


图 0.2 启动 IDLE 执行窗口 (Shell) 的情况

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。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.com>)

文档名称：《程序员学Python》裘宗燕 著.pdf

请登录 <https://shgis.com/post/3385.html> 下载完整文档。

手机端请扫码查看：

