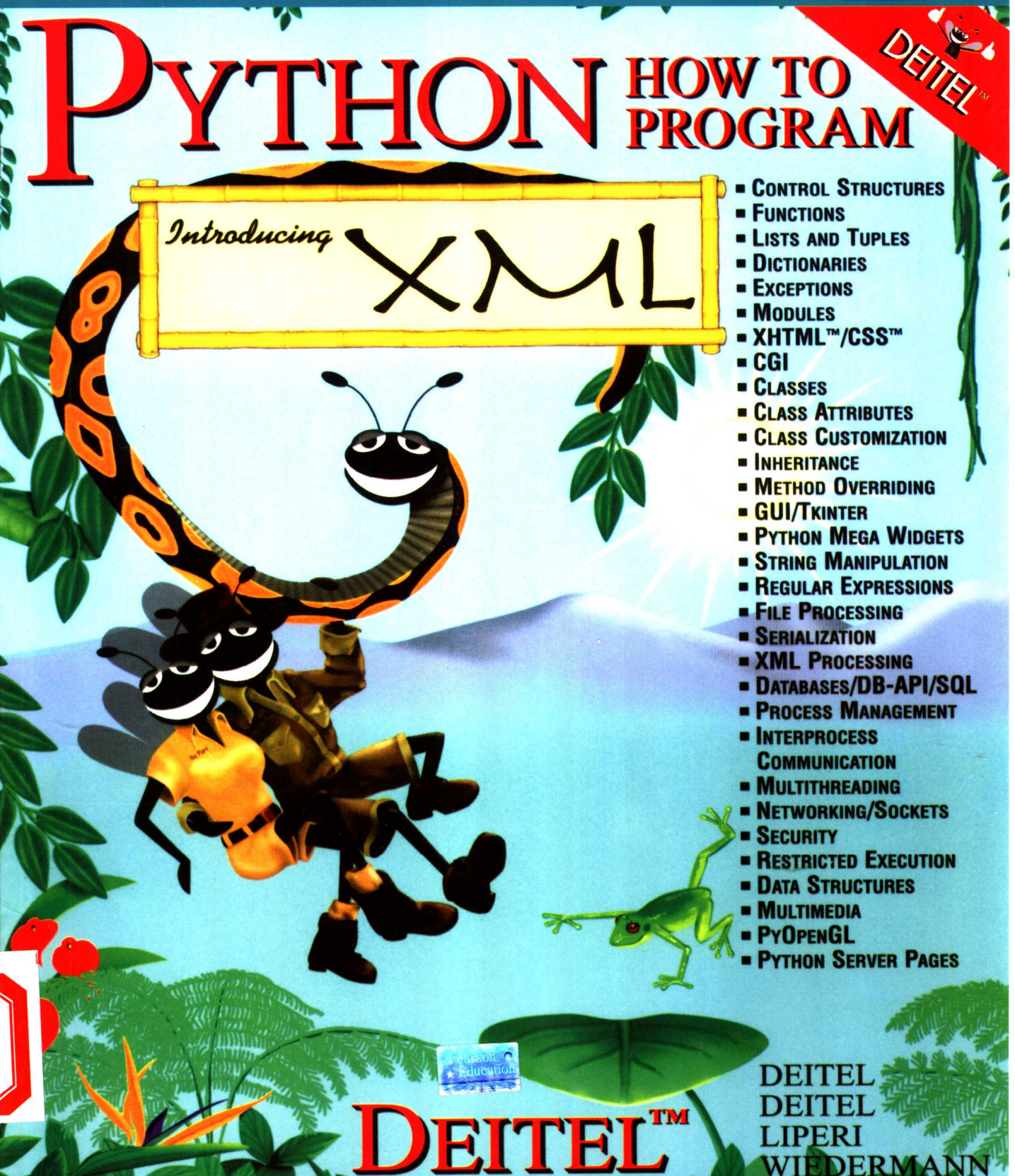
DEITEL™ 编程金典

# PYTHON 编程金典

PEARSON Prentice Hall

# PYTHON HOW TO PROGRAM

DEITEL™




- CONTROL STRUCTURES
- FUNCTIONS
- LISTS AND TUPLES
- DICTIONARIES
- EXCEPTIONS
- MODULES
- XHTML™/CSS™
- CGI
- CLASSES
- CLASS ATTRIBUTES
- CLASS CUSTOMIZATION
- INHERITANCE
- METHOD OVERRIDING
- GUI/TKINTER
- PYTHON MEGA WIDGETS
- STRING MANIPULATION
- REGULAR EXPRESSIONS
- FILE PROCESSING
- SERIALIZATION
- XML PROCESSING
- DATABASES/DB-API/SQL
- PROCESS MANAGEMENT
- INTERPROCESS COMMUNICATION
- MULTITHREADING
- NETWORKING/SOCKETS
- SECURITY
- RESTRICTED EXECUTION
- DATA STRUCTURES
- MULTIMEDIA
- PYOPENGL
- PYTHON SERVER PAGES

DEITEL™

DEITEL
DEITEL
LIPERI
WIEDERMANN

[美] H. M. Deitel, P. J. Deitel, J.P. Liperi, B. A. Wiedermann 著
周 靖 译

清华大学出版社

DEITEL 编程金典

# Python 编程金典

[美] H. M. Deitel，P. J. Deitel
J. P. Liperi，B. A. Wiedermann 著
周 靖 译

清华大学出版社
北京

## 内 容 简 介

本书由全球著名的编程培训专家 H. M. Deitel 博士领头编写，解释了如何将 Python 用做常规用途，编写基于 Internet 和 Web 的、数据密集型的、多层的客户端/服务器系统。书中采用作者独创的“活代码”教学方法，揭示了 Python 这一语言的强大功能。与此同时，本书还提供了大量的示例代码和编程技巧与提示，帮助读者搭建良好的知识结构，养成良好的编程习惯，指导读者如何避免常见的编程错误以及写出高效、可靠的应用程序。

本书适合对 Python 感兴趣的读者阅读和参考。

**EISBN: 0-13-092361-3**
**Python How To Program**


北京市版权局著作权合同登记号：图字 01-2002-4423 号

**图书在版编目（CIP）数据**

Python 编程金典/（美）迪特尔等著；周靖译.—北京：清华大学出版社，2003
(DEITEL 编程金典)
书名原文：Python How To Program
ISBN 7-302-06642-6
Ⅰ.P... Ⅱ.①迪...②周... Ⅲ.软件工具-程序-设计 Ⅳ.TP311.56
中国版本图书馆 CIP 数据核字（2003）第 036550 号

**出 版 者**：清华大学出版社(北京清华大学学研大厦,邮编 100084)
http://www.tup.com.cn
http://www.tup.tsinghua.edu.cn
**责任编辑**：文开棋
**印 刷 者**：清华大学印刷厂
**发 行 者**：新华书店总店北京发行所
**开　　本**：787×1092 1/16　**印张**：37.25　**字数**：1270 千字
**版　　次**：2003 年 6 月第 1 版　2003 年 6 月第 1 次印刷
**书　　号**：ISBN 7-302-06642-6/TP·4970
**印　　数**：0001～2000
**定　　价**：88.00 元

# 前 言

欢迎进入 Python 编程世界！Python 是一种强大的常规用途程序语言，尤其适合开发基于 Internet 和 Web 的、数据库密集型的、多层的客户机/服务器系统。本书讲解了大量先进的计算技术，它是我们的第二本有关开放源码程序语言的参考书。[1]

在我们编写本书时，Python 2.2 刚刚发布。为此，我们进行了艰苦的工作，以便将 Python 2.2 的功能合并到本书中。附录 B 将讲解 Python 2.2 的其他一些特性。

希望读者通过本书获取有用的信息，既能感到适度的挑战，又能从中获得无穷乐趣！本书写作过程令人愉快。Deitel & Associates 的团队长期致力于程序语言教科书和 e-Learning 素材的开发。我们涉足几乎每一种主流程序语言。在编写本书过程中，我们也注意到一些特别的地方。我们的开发者和撰稿人对 Python 给与高度评价。它强大的功能、高度的可靠性和编码的简洁性，给人留下深刻印象。他们喜欢 Python 能让他们运用自如。他们喜欢这种开发源码的软件开发世界，为 Python 而开发的模块正在与日俱增。

无论老师、学生、有经验的专业程序员还是新手，都能通过本书汲取有益的知识。Python 是一种出色的程序语言，也是开发具有工业强度的商业应用程序的优秀语言。对于学生和新手级程序员，通过前几章的学习，可打下良好的基础。我们讨论了许多程序开发模型，包括结构化编程、基于对象的编程、面向对象的编程以及事件驱动的编程等。对于专业开发人员，我们则选用 Python 真正强大的功能来创建实用的、进行了完全实现的系统。这部分的重点在于第 23 章的案例分析，它详细讲解了如何构建一个真正的网上书店。

本书涉及了所有标准主题，包括数据类型、运算符、控制结构、算术运算、字符串、决策、算法开发、函数和随机数/模拟等等。

本书的特色之一是全面讲解了数据结构，书中首先介绍了 Python 的内建结构——列表、元组和字典。之后，还对包括队列、堆栈、链表和二叉树在内的传统数据结构进行了深入讲解。

本书强调了 Internet 和 Web 开发——我们首先介绍了 CGI，并在随后几章用它来构建基于 Web 的应用程序。我们用整章篇幅（第 25 章）介绍了 PSP（Python Server Pages，Python 服务器页），并利用它改编了第 16 章介绍的一个论坛案例。

本书用 3 章篇幅详细介绍了面向对象编程，涉及的主题包括类、封装、对象、属性、方法、构造函数、析构函数、自定义、运算符重载、继承、基类、派生类和多态性等等。

本书透彻讲解了如何用 Tkinter 进行 GUI（Graphical User Interface，图形用户界面）编程，涉及的主题包括事件驱动编程、标签、按钮、复选框、单选钮、鼠标事件处理、键盘事件处理、布局管理器以及一系列高级 GUI 功能，利用它们可创建和处理菜单和滚动组件。

我们讨论了如何利用异常处理使程序更“健壮”。Python 强大的字符串处理功能在此得到了深入讲解。至于正则表达式的主题，虽然它不易于理解，但由于它功能强大，所以我们也进行了详尽的解释。

我们讨论了文件处理、顺序访问文件、随机访问文件（以及 shelve 模块），同时还开发了一个事务处理程序，论述了对象序列化的问题。通过讨论文件处理，为后来的 Python 数据库编程奠定了良好的基础，后者通过 Python 的 DB-API（Database Application Programming Interface，数据库应用程序编程接口）来实现。我们讨论了关系数据库模型，并概述了 SQL（Structured Query Language，结构化查询语言）。

许多人都熟悉 HTML，但很少有人知道万维网协会（W3C）——HTML 技术的创建者——已声称 HTML 已成为“过去”，不会继续开发它。全世界正逐步过渡到 XML（eXtensible Markup Language，可扩展标记语言）。在这期间，Web 开发将采用一种名为 XHTML 的过渡技术。本书许多应用程序都将采用 XHTML。至于 XML 的常规主题，将采用一整章（即第 15 章）的篇幅来介绍它。对如今的 Web 应用

[1] 第一本是《Perl 编程金典》，清华大学出版社 2002 年出版，ISBN 7-302-05751-6。

程序开发者来说，这是必须掌握的。然后，我们将另起一章（第 16 章），专门讲解 Python 的 XML 处理技术，并提供一个详细的案例分析，运用 CGI 和 XML 来构建论坛。

计算机应用程序通常能很好地一次做一件事。今天，较高级的应用程序需要同时做许多事，在计算机领域内，我们更喜欢将其称为“并发性”。我们会分别用整章的篇幅来讲解进程管理（第 18 章）和多线程处理（第 19 章）。Python 程序员利用这些技术，可做到以前只有系统程序员在操作系统的级别上才能做到的事情。

我们讨论了联网问题，包括 Web 使用的 HTTP 协议，使用流套接字进行的客户机/服务器联网，使用数据文报进行的无连接的客户机/服务器交互，另外还使用多线程服务器，实现了一个客户机/服务器的 Tic-Tac-Toe（即三连棋）游戏。

我们全面讨论了常见的计算机安全问题，并讲解了 Python 特有的一些安全问题。讨论了如何使用模块 Bastion 在一个限制环境中执行恶意代码。另外，还演示了如何用模块 rotor 对文本进行加密。

作为本书的一个重点，第 23 章展示了一个详尽的案例分析，它使用前几章和本书附录讨论的许多技术来实现一个电子商务网上书店。我们介绍了 HTTP 会话和会话跟踪技术，并将这个书店构建成一个多层的客户机/服务器系统，它有能力处理大批量的客户，其中包括标准 Web 浏览器（使用 XHTML）和无线客户（使用 WML 和 XHTML Basic）。

我们用整章（第 24 章）的篇幅讲解多媒体所涉及的主题。使用 3D 图形例子介绍了 PyOpenGL，并介绍了 3D 环境 Alice，它提供的对象可通过 Python 脚本“动”起来。我们通过设计一个 CD 播放器，一个影片播放器和一个太空船游戏，演示了 pygame。

认识到服务器端开发的重要性之后，我们将展示 PSP（Python 服务器页），它可取代 CGI。另外，我们将论坛案例从 CGI 技术转换成了 PSP。

本书提供了两个附录，均与 Python 有关。其中，附录 A 介绍了 Python 开发环境，附录 B 介绍了 Python 2.2 的其他特点，其中还讨论了迭代器、生成器和嵌套作用域。

阅读本书的过程中，不管遇到什么问题，都请联系 deitel@deitel.com，来信必复。另外，请经常访问我们的网站 *www.deitel.com*，并订阅免费的 The Deitel BUZZ 电子刊物。我们会通过网站及电子刊物介绍最新的 Python 信息以及我们推出的其他产品和服务。

## 本书特色

本书具有许多特色，包括：

- **代码清洗**（code washing）。这是我们自己创造的一个术语，是指对程序进行全面格式化，为其精心添加注释，并使其具有一个开放布局。程序代码组合成小的、结构清晰的块，这大大增强了可读性——这正是我们要达到的一个重要目标，尤其是全书总共含有 14 930 行代码！
- **面向对象编程**。面向对象编程是目前广泛采用的一种编程技术，用于开发健壮的、可重用的软件。Python 被设计成一种面向对象语言，本书全面讨论了 Python 的各种面向对象特性。数据完整性是 Python 中尤其要关注的一个问题。所有 Python 类数据默认都是公共的，但几种技术可用于确保数据完整性。我们用 3 章篇幅详细讨论了这些问题以及其他面向对象的主题。其中，第 7 章介绍如何创建类，并讨论了公共、私有和 get/set 方法。第 8 章解释如何使创建的类具有自定义行为，比如运算符重载、字符串表示、列表和字典行为以及用于自定义属性访问的方法等。第 9 章对这些概念进行了进一步扩展，我们讨论了如何创建新类，令其“吸收”现有类的功能。通过这一章的学习，读者会熟悉一些关键概念，比如多态性、抽象类和具体类等，利用它们可更方便地处理一个继承层次结构中的对象。本章最后讨论了 Python 2.2 提供的其他面向对象能力，其中包括“属性”（Properties）。
- **数据库应用程序编程接口**。数据库存储着大量信息，个人和单位需要访问它们以处理各种事务。数据库管理系统（DBMS）供单位和个人用来操纵数据库——Python 提供了相应的数据

库应用程序编程接口（DB-API），以访问数据库管理系统。第 17 章详细介绍了这些功能，另外还介绍了用于查询 MySQL 数据库的结构化查询语言（SQL）。

- **XML**。可扩展标记语言（XML）在软件开发领域和电子商务社区获得了蓬勃发展。由于 XML 是一种与平台无关的技术，用于描述数据和创建标记语言，所以 XML 的数据移植性能与 Python 的可移植应用程序和服务较好地集成在一起。第 15 章介绍了 XML，我们讨论了基本的 XML 标记和技术，比如 DTD 和 Schema，它们用于校验 XML 文档内容。第 16 章则解释了如何用文档对象模型（Document Object Model，DOM）来处理 XML 文档，以及如何通过可扩展样式表语言转换（eXtensible Stylesheet Language Transformation，XSLT），将 XML 文档转换成其他文档类型。本章还介绍了 DOM 的一种替代物，名为 Simple API for XML（简称 SAX），它充当用于 XML 的一个基于事件的 API。
- **公共网关接口（CGI）和 Python 服务器页（PSP）**。因特网和万维网已深入人们的日常生活，交互式网站是商业成功的关键。第 6 章和第 25 章介绍了服务器端 Web 技术，开发者利用它们可创建交互式的、基于 Web 的应用程序。第 23 章提供了一个详细的案例分析，它综合运用 MySQL、XML、XHTML、XHTML Basic、层叠样式表（CSS）、XSLT、CGI 和无线标记语言（Wireless Markup Language，WML）来构建一个动态电子商务应用程序。本书演示了一个 XML 论坛的两种实现方式，用户可将自己的文章张贴到在线论坛。第 16 章使用的是 CGI，第 25 章使用的则是 PSP。
- **图形用户界面（GUI）**。Python 没有内建图形用户界面功能，但有许多模块可供使用，它们提供对现有的 GUI 软件的访问途径。第 10 章和第 11 章讨论了 Tkinter 模块（包括在 Python 标准库中），它允许 Python 程序员访问 Tool Command Language/Took Kit（Tcl/Tk）这一流行的 GUI 工具包。利用这些编程工具，开发者可快速、方便地创建图形程序。利用在这几章所学的知识，读者可为本书其余部分的程序开发 GUI。第 11 章还讨论了模块 Pmw，它利用 Tkinter 提供更复杂的 GUI 组件。
- **多媒体**。多媒体功能可生成具有丰富视听感受的强大应用程序，由此增强用户的体验。可利用几个 Python 模块创建令人印象深刻的多媒体应用程序。第 24 章探讨了 PyOpenGL 和 Alice 的功能，它们可创建 3D 图形，并让它“动”起来。同时还讨论了 pygame，它所包含的模块便于开发者访问强大的多媒体库。第 24 章使用 pygame 创建一个 CD 播放器、一个电脑游戏以及一个影片播放器。
- **多线程处理和进程管理**。计算机可并发执行大量任务，比如同时打印文档、从网络下载文件和在 Web 上冲浪等等。利用多线程处理技术，应用程序可执行并发性任务。Python 的多线程处理和进程管理功能尤其适合今天高度复杂的、多媒体密集型的、数据库密集型的、基于网络的、基于多处理器的以及分布式的应用程序。第 18 章讨论了并发性和进程间通信；第 19 章详细讨论了多线程处理的问题，其中详细解释了 Python 的 Global Interpreter Lock（即全局解释器锁，负责管理线程执行）。本章还通过几个例子，介绍了常见的线程同步机制。
- **文件处理和序列化**。大多数应用程序都要在磁盘上读写数据。Python 针对数据存储和获取提供了几项高级功能。第 14 章讨论了用于存储顺序数据的基本文件对象、用于存储随机访问数据的 shelve 对象，以及用于将整个对象序列化到磁盘的 cPickle 模块。

此外，本书还讨论了其他许多主题。要详细了解每章的特点，请参阅 1.6 节。

## Python 2.2 的特性

本书出版时，喜闻 Python 2.2 正式版刚刚发布。然而，本书所有示范代码都通过了 Python 2.2b2（即 Beta 2）和 Release Candidate 1 的测试。测试平台包括 Windows 和 Linux 操作系统。我们在各章尽可能介

绍 Python 2.2 的特性和功能。[1]本书要介绍 Python 2.2 的以下特性。

**Floor 除法和 True 除法**：Python 2.2 引入新运算符（//）进行 Floor（整数）除法。在此之前的 Python 版本中，除法运算符（/）的默认行为是 Floor 除法；在 2.2 以后的版本，默认行为变成 True（浮点）除法。通过定义两个除法运算符，Python 新版本可在同时使用了整数及浮点除法的程序中，避免出现类型混淆的问题。本书 2.6 节讨论了这两种除法的区别，并解释了程序如何更改除法运算符（/）的默认行为，令其执行 True 除法。

**嵌套作用域**：Python 2.2 引入了嵌套作用域的概念，它意味着嵌套的类、方法和函数现在可访问其封闭作用域中定义的变量。这种行为尤其适合编写 lambda 表达式。第 4 章讨论了 Python 的基本作用域规则，并提供了一系列万维网资源，便于读者更深入地了解嵌套作用域。如程序员在一种功能性编程模型中使用 Python，作用域的嵌套就显得非常重要。附录 B 更详细地讨论了嵌套作用域。由于本书强调的主要是面向对象的编程风格，所以只指出了使用嵌套作用域的一个高级动机，并推荐了可进一步参考的资源，便于读者在需要时了解 Python 中的嵌套作用域和功能性编程。

**更多的面向对象功能**：Python 2.2 的大多数新特性是为语言添加更多的面向对象功能。第 8 章和第 9 章介绍了其中的一些新特性。第 8 章讨论了如何重载一个由程序员定义的类，以便为运算符（包括新运算符//）定义行为。介绍了一个字典方法，它也是 Python 2.2 版本新增的，便于程序用 if/in 语句检测字典中是否包含一个特定的键。第 9 章讨论开发者们期待以久的新特性——允许从内建类型继承程序员定义的类。本章展示了一个实例，它继承自内建类型 list，目的是实现由程序员定义一个列表，其中只包含惟一性的元素。还讨论了其他面向对象特性，其中包括静态方法，__slots__（用于定义类中可能包含的一个对象的属性），方法__getattribute__（客户每次访问一个对象的属性时执行），以及属性（允许类定义 get/set 方法，以便在客户访问一个属性时执行）。

**迭代器**：附录 B 介绍的其他 Python 2.2 特点中，有的未在正文中详细讲解。附录 B 首先全面地探讨了迭代器——用于遍历一系列值的特殊对象。B.2 节提供了两个例子，它们展示了由程序员定义的迭代器类，并演示该类的一个客户如何使用迭代器从一个序列中获取值。第一个例子展示了如何定义一个类，使它的对象支持迭代器；第二个例子展示了一个计算机猜谜游戏，它展示如何利用迭代器处理长度不确定的序列。Python 2.2 采用了新的迭代器机制后，性能比以前的版本有了显著改进。另外，软件设计也因为程序员能分离迭代行为和随机访问行为而得以改进。

**生成器**：这是一种“可恢复函数”，能记住两次调用期间的状态。生成器也有利于性能和设计的改进。通常，程序中可以写一个生成器，采用一种简单、直观方式定义如何生成一个序列的元素。生成器还有利于执行重复性任务，或要求复杂逻辑和状态信息才能完成的任务。B.2 节围绕上述问题讨论了生成器，并定义了两个版本的生成器来计算斐波拉契序列。第一个版本不确定地生成序列中的下一个值；第二个生成所有序列值，直到包括用户自定义的第 *n* 个值。

## 万维网访问

本书（以及我们的其他出版物）所有示例代码都可从以下网站下载：

*www.deitel.com*
*www.prenhall.com/deitel*

注册过程非常简单。建议下载所有例子，并在阅读时运行相应的程序。修改例子，可马上看到修改效果——这是提升编程水平的有效方式之一。上述网站还解释了如何安装本书用到的各种软件（比如 Apache Web Server）。网站还提供其他 Web 服务器和软件的安装指南（注意，它们是有版权的。学习时可任意使用，但未经 Prentice Hall 和作者的书面许可，不得采取任何方式重新出版它的任何部分）。

[1] 阅读本书前，先从 *python.org* 下载最新 Python 版本。新版本发布后，我们会对本书代码进行测试，并在 *www.deitel.com* 进行相应的更新。阅读每一章之前，最好能访问我们的网站查看这些更新。

# 目 录

# 第1章 绪 论

**学习目标**

- 了解开放源码软件
- 熟悉 Python 程序语言的历史
- 本书导读

## 1.1 简介

欢迎进入 Python 的世界！希望通过我们艰苦的努力，带给读者一本内涵丰富、寓教于乐的计算机参考书。为此，我们采用了多种方法，最终写成了一本与众不同的 Python 参考书。例如，我们很早便讲解了 Python 如何与公共网关接口（CGI）配合，以便编写基于 Web 的应用程序。这样一来，便可在本书剩余部分，更好地演示大量动态的、基于 Web 的应用程序。本书介绍了大量重要主题，包括面向对象编程（OOP）、Python 数据库应用程序编程接口（DB-API）、图形、可扩展标记语言（XML）以及安全性。不管您是一名新手还是有经验的程序员，本书提供的信息量、趣味性和挑战性，都会让您称心如意。

本书面向所有层次的读者，从正式的程序员，一直到很少或根本没有编程经验的自学者。同一本书，怎么可能同时适用于高低两个层次的读者呢？其中的关键在于，我们以成熟的“结构化编程”以及“基于对象的编程”技术为准，始终都在强调如何编写“思路清晰”的程序。非程序员出身的人可以学会基本的技能，为将来进行良好编程奠定基础。有经验的程序员将获得对语言的严谨解释，而且书中的内容有助于改进他们的编程风格。为帮助刚入门的程序员，我们采用一种清晰的、平铺直叙的方式，其间穿插大量插图。另外，更重要的是，本书提供了数百个功能完整的 Python 程序。本书所有示例程序都可从我们的网站（*www.deitel.com*）下载。

大多数人或多或少都熟悉计算机令人激动的功能。使用本书，可学会如何指挥计算机，亲自实现那些功能。毕竟，是“软件”（要求计算机采取行动和做出决策的指令）在控制着计算机（通常称为“硬件”）。

今天，几乎每个领域都越来越依赖计算机。在其他成本都在稳定、缓慢提升的同时，计算成本却呈显著下降态势——这完全归功于硬件和软件技术的快速发展。25～30 年前，需要几个大房间才能摆下的大型计算机，以及那些动辄数百万美元的“巨无霸”，现在只需指甲大小的硅芯片即可搞定。成本也降至每片几美元左右。不过，具有讽刺意味的是，“硅”是我们这个地球上不值钱的东西之一。在海边随手抓一把沙子，里面含的绝大多数元素便是“硅”。硅芯片技术的问世，使得计算成本变得异常低廉，也直接促成了如今数亿台计算机广泛应用于各行各业。在商业、工业、政府以及我们的个人生活方面，计算机都能提供强有力的帮助。而且再过几年，这个数字还可以轻松翻一倍。

从现在起，您将开始一段美妙的、令人激动的、充满挑战的以及令人回味无穷的学习之旅。在这个旅程中，如果您碰到什么问题，不妨发信给 *deitel@deitel.com*，或浏览我们的网站（*www.deitel.com*、*www.prenhall.com/deitel* 和 *www.InformIT.com/deitel*）。祝您学习顺利。

## 1.2 开放源码软件的革命

如果程序的源代码免费提供给任何开发人员进行修改，传播，以及用作其他软件的基础，就称为“开放源码软件”或“开源软件”。[1]相反，“封闭源码软件”则禁止其他开发者以之为基础开发其他软件。

开放源码并不是一个新概念。早在 20 世纪 60 年代，开源技术就是现代计算工业得以迅速发展的一项主要推动力。特别是美国政府出资兴建了今天 Internet 的前身，并鼓励计算机科学家开发各种各样的

[1] 真正的开源（Open-Source）软件要符合 9 条要求。详情参见 *www.opensource.org/docs/definition.html*。

技术，以实现在各种计算机平台上的分布式计算。这演变成了今天的一系列重要技术，比如用于实现 Internet 通信的协议。Internet 建好之后，封闭源码技术及软件才逐渐在软件工业中流行起来，开放源码的思想在 20 世纪 80 年代和 90 年代初一度受到抑制。但在这之后，为了反击大多数商业软件的“封闭”本质，并因为封闭源码厂商冷淡的态度，开放源码软件再度流行。如今，Python 已成为快速成长的开源软件社区的一部分，其他还有 Linux 操作系统、Perl 脚本编程语言、Apache Web 服务器以及成百上千的其他软件项目。

计算机行业的一些人将开放源码视为 Free（免费）软件。大多数情况下，这种说法是成立的。不过，在谈到开放源码软件“Free”时，一种更合适的说法是“Freedom”（自由）——任何开发者都可自由修改源码，交流编程思想，参与软件开发过程，以及基于现有的开源软件开发出新的软件程序。大多数开源软件实际都是有版权的，要使用它们，需要获得相应的许可证。开放源码许可协议所规定的条款是各不相同的；有的只有极少限制（比如 Artistic License），另一些可能有许多限制，详细规定软件的修改和使用方式。通常，软件的版权由个人开发者或者一个组织所持有。若要查看许可证/许可协议的一个例子，可访问 *www.python.org/2.2/license.html*，仔细阅读 Python 的使用许可协议。

通常，可通过 Internet 下载开源产品的源代码。这使开发者能够学习、检查和修改源码，以满足他们自己的需求。由于有整个开发者社区作为后盾，有许多人审查代码，所以相较于封闭源码软件开发，性能和安全问题能更快得到检测和解决。另外，更大的开发者社区可为一个软件提供更多的特性。通常，代码修正数小时内便可推出，开源软件新版本的推出频率也要比封闭源软件高出许多。在开源软件的使用许可协议中，通常要求开发者发表他们所做的任何改进，这使得开源社区不断地发展壮大，不断地改进现有的产品。例如，Python 开发者喜欢加入 comp.lang.python 新闻组，交流有关 Python 开发的心得体会。Python 开发者还可为自己的修改编写文档，并通过 Python Enhancement Proposals（PEPS）提交给 Python Software Foundation（Python 软件基金会）。这样一来，Python 开发团体就能对提议的修改进行评估，并在未来的版本中集成好的改进。

许多公司（比如 IBM，Red Hat 和 Sun）都支持开源开发者及项目。有时，这些公司还会取得开源应用程序，并通过商业途径销售它们（这取决于软件的使用许可协议）。为获得赢利，他们还为开源软件提供一系列服务，比如技术支持、为客户定做软件和培训等等。开发者可作为顾问或培训者提供服务，帮助用户实现软件。欲知开源软件的详情，请访问 Open Source Initiative 网站（*www.opensource.org*）。

## 1.3 Python 的历史

Python 起源于 1989 年末。当时，CWI（阿姆斯特丹国家数学和计算机科学研究所）的研究员 Guido van Rossum 需要一种高级脚本编程语言，为其研究小组的 Amoeba 分布式操作系统执行管理任务。为创建新语言，他从高级教学语言 ABC（All Basic Code）汲取了大量语法，并从系统编程语言 Modula-3 借鉴了错误处理机制。然而，ABC 的一个重大缺点是扩展性不足；语言不是开放式的，不利于改进或扩展。因此，Van Rossum 决定在新语言中合成来自现有语言的许多元素，但要求必须能通过类和编程接口进行扩展。他将这种新语言命名为 Python（原意为“大蟒蛇”）——来源于 BBC 当时正在热播的喜剧片连续剧“Monty Python”。

自 1991 年初公开发行后，Python 开发者和用户社区逐渐壮大，Python 语言逐渐演变成一种成熟的、并获得良好支持的程序语言。Python 被用来开发大量应用程序，从创建网上电子邮件程序到控制水下交通工具，以及配置操作系统和创建动画片等等。2001 年，核心 Python 开发团队移师 Digital Creations 公司，后者是 Zope（用 Python 编写的一个 Web 应用程序服务器）的创始人。预计 Python 会继续成长与发展，进入一个全新的领域。

## 1.4 Python 模块

Python 是一种模块化的可扩展语言，它可随时集成新“模块”（Modules）——一种可重用的软件组件。任何 Python 开发人员都能编写新模块来扩充 Python 的功能。Python 源代码、模块和文档资料的主要“集散地”是 Python 网站（*www.python.org*），该网站还计划建立一个专门维护 Python 模块的网站。

## 1.5 Python 和本书的一般注意事项

Python 经过了良好的设计，无论新手还是有经验的程序员都能快速学习和理解这种语言，并轻松上手。和其前身不同，Python 具有良好的移植和扩展能力。Python 的语法和设计有利于养成良好的编程习惯，并可在不牺牲程序扩展性与维护性的同时，显著缩短开发时间。

Python 相当简单，新手程序员可轻松上手；但它同时具有强大的功能，对专家也有足够的吸引力。本书通过丰富、完整且实际有效的例子和讨论，介绍了大量编程概念。随着学习的深入，读者可通过我们创建的实际应用程序，探索更加复杂的主题。贯穿全书，我们始终在强调良好的编程习惯，并给出大量移植性提示以及解释如何防范常见的编程错误。

Python 是当前移植能力最强的程序语言之一。最初，它是在 UNIX 上实现的。但之后扩展到了其他许多平台，其中包括 Microsoft Windows 和 Apple Mac OS X。Python 程序通常可直接从一种操作系统移植到另一种操作系统，无需任何更改，而且能确保正确执行。

## 1.6 本书导读

本节简要介绍全书讲解的各个主题。有的章末附有“因特网和万维网资源”小节，提供有关 Python 编程的更多信息。

**第 1 章——绪论**：介绍开放源码革命，讲述 Python 程序语言的起源，并概括本书其余各章主要内容。

**第 2 章——Python 编程概述**：介绍一个典型的 Python 编程环境，并解释了 Python 程序的基本语法。我们讨论了如何从命令行运行 Python。除解释器之外，Python 还可在一个交互模式中执行语句，即输入一个语句，立即执行一个。在本章及全书，我们会展示许多交互会话，强调各种平常容易忽视的编程要点。本章讨论了变量，介绍了算术运算、赋值、相等、关系和字符串运算符。我们介绍了决策和算术运算。字符串是一种基本的、功能强大的内建数据类型。我们介绍了一些标准的输出格式化技术，并讨论了“对象”和“变量”的概念。对象是值的容器，而变量是用于引用对象的名称。结束本章的学习后，读者将理解如何编写简单而又完整的 Python 程序。

**第 3 章——控制结构**：介绍用于解决问题的“算法”（过程）。解释了高效率使用控制结构的重要性：控制结构可使程序易于理解、调试和维护，而且有助于首次试运行即告成功。本章介绍了选择结构（if，if/else 和 if/elif/else）和重复结构（while 和 for）。我们详尽解释了如何进行重复，并对比了计数器控制的循环和哨兵值控制的循环。同时还解释了“自上而下求精法”为什么有助于生成结构正确的程序，有助于建立一种流行的程序设计辅助手段——伪代码。本章提供的案例分析演示了如何快速方便地将伪代码算法转换成能实际工作的 Python 代码。本章还解释了用于改变控制流程的 break 和 continue 语句。另外，我们展示了怎样用逻辑运算符 and，or 和 not 在程序中做出复杂的决策。本章的几个交互式会话演示了如何创建一个 for 结构，以及如何避免结构化编程中的一些常见编程错误。本章最后对结构化编程进行了总结。利用本章介绍的技术，能在任何程序语言中有效使用控制结构，而非局限于 Python。本章有助于读者培养良好的编程习惯，为进行本书后面更高级的编程任务打好基础。

**第 4 章——函数**：讨论了“函数”的设计与构建。在 Python 中，和函数相关的功能包括内建函数、程序员定义的函数和递归等。本章介绍的技术是创建具有正确结构的程序的基础（尤其是系统程序员和