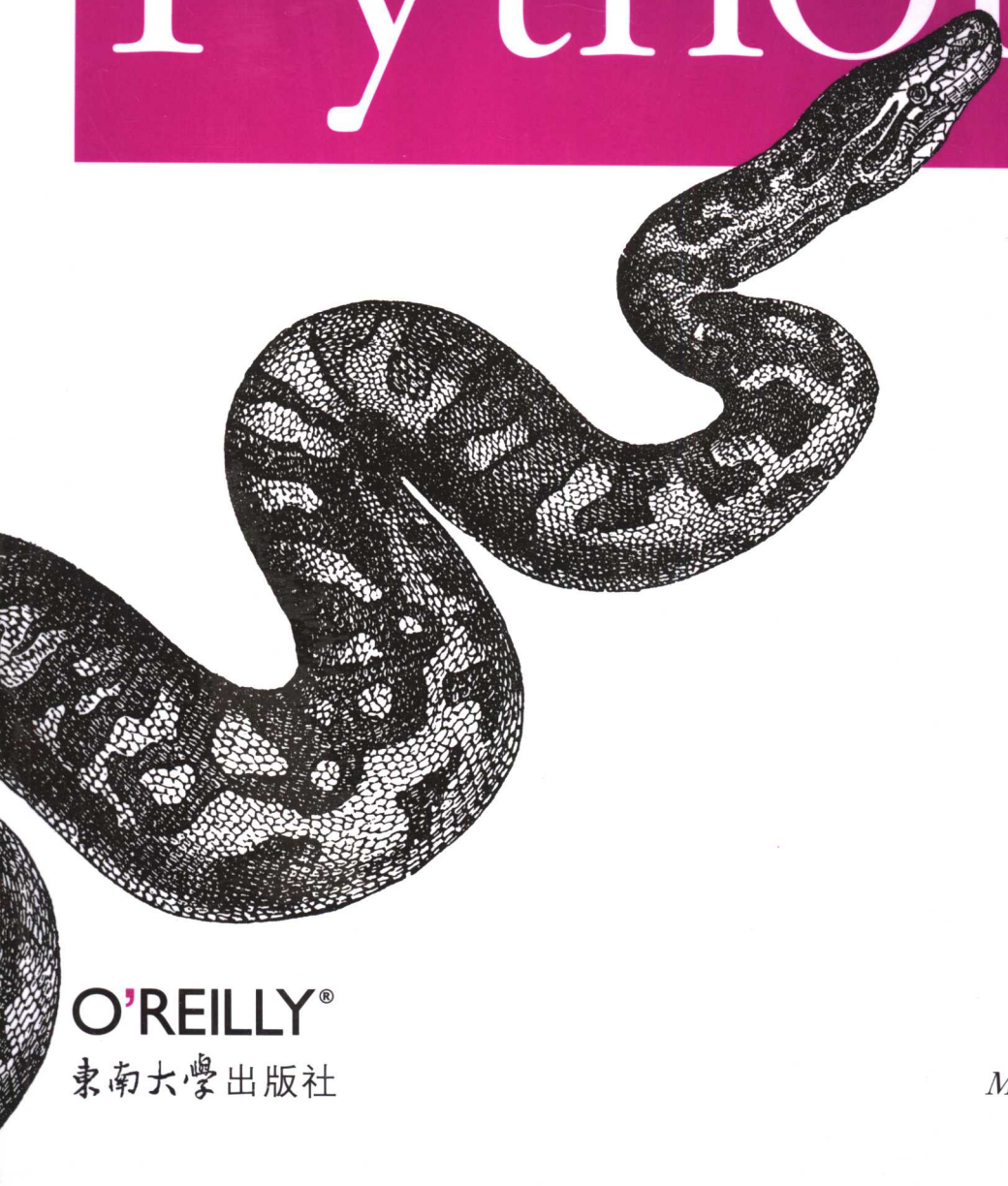
Python 编程（影印版）

3rd Edition
下册

Programming

Python



O'REILLY®

東南大學出版社

Mark Lutz 著

第三版

Python 编程(影印版)

Programming Python

下册

Mark Lutz

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

Python编程：第3版 /（美）卢茨（Lutz, M.）著.—影印本.— 南京：东南大学出版社，2006.11

书名原文：Programming Python, Third Edition

ISBN 7-5641-0570-4

Ⅰ.P... Ⅱ.卢 Ⅲ.软件工具－程序设计－英文 Ⅳ.TP311.56

中国版本图书馆CIP数据核字（2006）第115484号

江苏省版权局著作权合同登记

图字：10-2006-257号

书　　名/ Python编程 第三版（影印版）

书　　号/ ISBN 7-5641-0570-4

责任编辑/ 张烨

封面设计/ Edie Freedman，张健

出版发行/ 东南大学出版社（press.seu.edu.cn）

地　　址/ 南京四牌楼2号（邮政编码210096）

印　　刷/ 扬中市印刷有限公司

开　　本/ 787毫米×980毫米　16开本　100.75印张

版　　次/ 2006年11月第1版　2006年11月第1次印刷

印　　数/ 0001-2500册

全套定价/ 138.00元（上下册）

O'Reilly Media, Inc.介绍

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为O'Reilly Media, Inc.紧密地与计算机业界联系着，所以O'Reilly Media, Inc.知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用，人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而，计算机领域的技术更新速度之快也是众所周知的，为了帮助国内技术人员在第一时间了解国外最新的技术，东南大学出版社和美国 O'Reilly Meida, Inc.达成协议，将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作，以影印版或者简体中文版的形式呈献给读者。其中，影印版书籍力求与国外图书“同步”出版，并且“原汁原味”展现给读者。

我们真诚地希望，所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助，对国内计算机技术的发展有所促进。也衷心期望读者提出宝贵的意见和建议。

最新出版的一批影印版图书，包括：

- 《深入浅出 Ajax》（影印版）
- 《Ajax Hacks》（影印版）
- 《深入理解 Linux 网络内幕》（影印版）
- 《Web 设计技术手册 第三版》（影印版）
- 《软件预构艺术》（影印版）
- 《Ruby on Rails: Up and Running》（影印版）
- 《Ruby Cookbook》（影印版）
- 《Python 编程 第三版》（影印版）
- 《Python 技术手册 第二版》（影印版）
- 《Ajax 设计模式》（影印版）
- 《实用软件项目管理》（影印版）
- 《用户界面设计模式》（影印版）

About the Author

Mark Lutz is the world leader in Python training, the author of Python's earliest and best-selling texts, and a pioneering figure in the Python community.

Mark is also the author of the O'Reilly book *Python Pocket Reference*, and coauthor of *Learning Python*, all currently in second or third editions. Involved with Python since 1992, he started writing Python books in 1995 and began teaching Python classes in 1997. As of mid-2006, he has instructed more than 170 Python training sessions.

In addition, he holds B.S. and M.S. degrees in computer science from the University of Wisconsin, and over the last two decades has worked on compilers, programming tools, scripting applications, and assorted client/server systems.

Whenever Mark gets a break from spreading the Python word, he leads an ordinary, average life in Colorado. Mark can be reached by email at *lutz@rmi.net*, or on the Web at *http://www.rmi.net/~lutz*.

Colophon

The animal on the cover of *Programming Python* is an African rock python, one of approximately 18 species of python. Pythons are nonvenomous constrictor snakes that live in tropical regions of Africa, Asia, Australia, and some Pacific Islands. Pythons live mainly on the ground, but they are also excellent swimmers and climbers. Both male and female pythons retain vestiges of their ancestral hind legs. The male python uses these vestiges, or spurs, when courting a female.

The python kills its prey by suffocation. While the snake's sharp teeth grip and hold the prey in place, the python's long body coils around its victim's chest, constricting tighter each time it breathes out. They feed primarily on mammals and birds. Python attacks on humans are extremely rare.

The cover image is a 19th-century engraving from the *Dover Pictorial Archive*. The cover font is Adobe ITC Garamond. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSans Mono Condensed.

Table of Contents

上册

Part I. The Beginning

Part II. System Programming

下册

Part IV. Internet Programming

Part VI. Integration

Part VII. The End

CHAPTER 13

Network Scripting

"Tune In, Log On, and Drop Out"

Over the last decade, the Internet has virtually exploded onto the mainstream stage. It has rapidly grown from a simple communication device used primarily by academics and researchers into a medium that is now nearly as pervasive as the television and telephone. Social observers have likened the Internet's cultural impact to that of the printing press, and technical observers have suggested that all new software development of interest occurs only on the Internet. Naturally, time will be the final arbiter for such claims, but there is little doubt that the Internet is a major force in society, and one of the main application contexts for modern software systems.

The Internet also happens to be one of the primary application domains for the Python programming language. It has been a decade since the first edition of this book was written as well, and in that time the Internet's growth has strongly influenced Python's tool set and roles. Given Python and a computer with a socket-based Internet connection today, we can write Python scripts to read and send email around the world, fetch web pages from remote sites, transfer files by FTP, program interactive web sites, parse HTML and XML files, and much more, simply by using the Internet modules that ship with Python as standard tools.

In fact, companies all over the world do: Google, Yahoo!, Walt Disney, Hewlett-Packard, JPL, and many others rely on Python's standard tools to power their web sites. For example, the Google search engine—widely credited with making the web usable—makes extensive use of Python code. And the BitTorrent peer-to-peer file transfer system—written in Python and already downloaded by tens of millions of users—leverages Python's networking skills to share files among clients and remove server bottlenecks.

Many also build and manage their sites with the Zope web application server, which is itself written and customizable in Python. Others build sites with the Plone content management system, which is built upon Zope and delegates site content to its users. Still others use Python to script Java web applications with Jython (formerly

known as JPython)—a system that compiles Python programs to Java bytecode, exports Java libraries for use in Python scripts, and allows Python code to serve as web applets downloaded and run in a browser.

More recently, XML-RPC and SOAP interfaces for Python, such as xmlrpclib and SOAPy, have enabled web service programming; frameworks such as CherryPy, Webware, TurboGears, and Django have emerged as convenient tools for constructing web sites; the new XML package in Python's standard library provides a suite of XML processing tools; and the new IronPython implementation promises to provide seamless .NET/Mono integration for Python code.

As the Internet has grown, so too has Python's role as an Internet tool. Python has proven to be well suited to Internet scripting for some of the very same reasons that make it ideal in other domains. Its modular design and rapid turnaround mix well with the intense demands of Internet development. In this part of the book, we'll find that Python does more than simply support Internet scripts; it also fosters qualities such as productivity and maintainability that are essential to Internet projects of all shapes and sizes.

Internet Scripting Topics

Internet programming entails many topics, so to make the presentation easier to digest, I've split this subject over the next six chapters of this book. This chapter introduces Internet fundamentals and explores sockets, the underlying communications mechanism of the Internet. From there, later chapters move on to discuss the client, the server, web site construction, and more advanced topics.

Each chapter assumes you've read the previous one, but you can generally skip around, especially if you have any experience in the Internet domain. Since these chapters represent a substantial portion of this book at large, the following sections go into a few more details about what we'll be studying.

What we will cover

In conceptual terms, the Internet can roughly be thought of as being composed of multiple functional layers:

Low-level networking layers
: Mechanisms such as the TCP/IP transport mechanism, which deal with transferring bytes between machines, but don't care what they mean

Sockets
: The programmer's interface to the network, which runs on top of physical networking layers like TCP/IP

Higher-level protocols
: Structured communication schemes such as FTP and email, which run on top of sockets and define message formats and standard addresses

Server-side web scripting (CGI)
: Higher-level client/server communication protocols between web browsers and web servers, which also run on top of sockets

Higher-level frameworks and tools
: Third-party systems such as Zope and Jython, which address larger problem domains

In this chapter and in Chapter 14, our main focus is on programming the second and third layers: sockets and higher-level protocols. We'll start this chapter at the bottom, learning about the socket model of network programming. Sockets aren't strictly tied to Internet scripting, but they are presented here because this is their primary role. As we'll see, most of what happens on the Internet happens through sockets, whether you notice or not.

After introducing sockets, the next two chapters make their way up to Python's client-side interfaces to higher-level protocols—things like email and FTP transfers, which run on top of sockets. It turns out that a lot can be done with Python on the client alone, and Chapters 14 and 15 will sample the flavor of Python client-side scripting. The next two chapters then go on to present server-side scripting—programs that run on a server computer and are usually invoked by a web browser. Finally, the last chapter in this part, Chapter 18, briefly introduces even higher-level tools such as Jython and Zope.

Along the way, we will also put to work some of the operating-system and GUI interfaces we studied earlier (e.g., processes, threads, signals, and Tkinter), and we'll investigate some of the design choices and challenges that the Internet presents.

That last statement merits a few more words. Internet scripting, like GUIs, is one of the sexier application domains for Python. As in GUI work, there is an intangible but instant gratification in seeing a Python Internet program ship information all over the world. On the other hand, by its very nature, network programming imposes speed overheads and user interface limitations. Though it may not be a fashionable stance these days, some applications are still better off not being deployed on the Net. In this part of the book, we will take an honest look at the Net's trade-offs as they arise.

The Internet is also considered by many to be something of an ultimate proof of concept for open source tools. Indeed, much of the Net runs on top of a large number of such tools, such as Python, Perl, the Apache web server, the sendmail program, MySQL, and Linux.* Moreover, new tools and technologies for programming the Web sometimes seem to appear faster than developers can absorb them.

* In fact, there is even a common acronym for this today: LAMP, for the Linux operating system, the Apache web server, the MySQL database system, and the Python, Perl, and PHP scripting languages. It's possible, and even very common, to put together an entire enterprise-level web server with open source tools. Python users would probably also like to include systems like Zope, Plone, Webware, and CherryPy in this list, but the resulting acronym might be a bit of a stretch.