

学习Python (影印版)

2nd Edition  
Covers Python 2.3

*Learning*

Python



O'REILLY®

東南大學出版社

Mark Lutz & David Ascher 著

**图书在版编目(CIP)数据**

学习 Python: 第2版 / (美) 卢茨 (Lutz, M.),  
(美) 阿舍 (Ascher, D.) 著. —影印本. —南京: 东南大  
学出版社, 2005. 6

书名原文: Learning Python, Second Edition

ISBN 7-5641-0047-8

I. 学... II. ①卢... ②阿... 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(2005)第 047638 号

江苏省版权局著作权合同登记

图字:10-2005-080 号

© 2003 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2005.  
Authorized reprint of the original English edition, 2003 O'Reilly Media, Inc., the owner of all rights to publish  
and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2003。

英文影印版由东南大学出版社出版 2005。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly  
Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名/ 学习 Python(影印版)

书 号/ ISBN 7-5641-0047-8/TP·8

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

地 址/ 南京四牌楼 2 号 邮编 210096

经 销/ 各地新华书店

印 刷/ 扬中市印刷有限公司

开 本/ 787 毫米×1092 毫米 16 开本 39 印张

版 次/ 2005 年 6 月第 1 版 2005 年 6 月第 1 次印刷

印 数/ 0001-1500 册

定 价/ 68.00 元(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 20 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 Media, 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。

第一批影印图书共 10 本,涉及 Java, Unix/Linux, Python 等方面:

- 《核心 Java》(影印版)
- 《Jakarta Commons 经典实例》(影印版)
- 《Weblogic 权威指南》(影印版)
- 《Java 网络编程 第三版》(影印版)
- 《Linux 设备驱动程序 第三版》(影印版)
- 《LPI Linux 认证权威指南》(影印版)
- 《GNU Make 项目管理 第三版》(影印版)
- 《游戏开发中的人工智能》(影印版)
- 《学习 Python 第二版》(影印版)
- 《精通正则表达式 第二版》(影印版)

*To the late Frank Willison, our mentor, friend,  
and first editor.*

---

# Preface

This book provides an introduction to the Python programming language. Python is a popular object-oriented language used for both standalone programs and scripting applications in a variety of domains. It is free, portable, powerful, and remarkably easy to use.

Whether you are new to programming or a professional developer, this book's goal is to bring you up to speed on the core Python language in a hurry.

## About This Second Edition

In the four years after the first edition of this book was published in late 1998, there have been substantial changes in both the Python language and in the topics presented by the authors in Python training sessions. Although we have attempted to retain as much of the original version as possible, this new edition reflects recent changes in both Python and Python training.

On the language front, this edition has been thoroughly updated to reflect Python 2.2, and all changes to the language since publication of the first edition. In addition, discussion of anticipated changes in the upcoming 2.3 release have been incorporated throughout. Some of the major language topics for which you'll find new or expanded coverage in this edition are:

- List comprehension (Chapter 14)
- Class exceptions (Chapter 25)
- String methods (Chapter 5)
- Augmented assignment (Chapter 8)
- Classic, true, and floor division (Chapter 4)
- Package imports (Chapter 17)
- Nested function scopes (Chapter 13)
- Generators and iterators (Chapter 14)
- Unicode strings (Chapter 5)

- Subclass types (Chapters 7 and 23)
- Static and class methods (Chapter 23)
- Pseudo-private class attributes (Chapter 23)
- Extended print and import statements (Chapters 8 and 18)
- New built-ins such as `zip` and `isinstance` (Chapters 7 and 10)
- New-style classes (Chapter 23)
- New configuration and launch options, and *.pth* files (Chapter 3 and Appendix A)
- New development tools such as IDLE, Psycho, Py2exe, and Installer (Chapters 2, 3, and 29)
- New testing and documentation tools such as PyDoc, PyUnit, and doctest (Chapter 26)

Smaller language changes (e.g., long integer promotion, module export lists) appear throughout the book. Besides such language changes, we augmented the core language parts of this edition (Parts I–VII) with new topics and examples presented in the Python training sessions Mark has held in recent years. For example, you’ll find:

- A new OOP introduction (Chapter 19)
- A new dynamic typing overview (Chapter 4)
- A new development tools summary (Chapter 26)
- New material on program architecture and execution (Chapters 2, 3, and 15)
- New coverage of documentation sources (Chapter 11)

Many core language part additions and changes were made with beginners in mind. You’ll also find that the coverage of many original core language topics has been substantially expanded in this edition, with new discussion and examples. Because this text has largely become the primary resource for learning the core Python language, we’ve taken liberties with making that coverage more complete than before, and added new use cases throughout. Likewise, we updated Part VIII to reflect recent Python application domains, and modern usage patterns.

In addition, this entire edition integrates a new set of Python tips and tricks, gleaned from both teaching classes over the last seven years, and using Python for real work over the last decade. The exercises have been updated and expanded to reflect current Python practice, new language features, and common beginner mistakes we’ve witnessed first-hand in recent years. Overall, this edition is bigger, both because Python is bigger, and because we’ve added context that has proved to be important in practice.

To accommodate the fact that this edition is more complete, we’ve split most of the original chapters into bite-sized chunks. That is, we’ve reorganized the core language section into many multichapter parts, to make the material easier to tackle. Types and statements, for instance, are now two top-level parts, with one chapter for

each major type and statement topic. This new structure is designed to allow us to say more, without intimidating readers. In the process, exercises and gotchas were moved from chapter ends to part ends; they now appear at the end of the last chapter in each part.

Despite all the new topics, this book is still oriented toward Python newcomers, and is designed to be a first Python text for programmers.\* It retains much of the first edition's material, structure, and focus. Where appropriate, we have expanded introductions for newcomers, and isolated the more advanced new topics from the main thread of discussion to avoid obscuring the fundamentals. Moreover, because it is largely based on time-tested training experience and materials, this edition, like the first, can still serve as a self-paced introductory Python class.

## Prerequisites

There are none to speak of, really. This book has been used successfully by both absolute beginners, and crusty programming veterans. In general, though, we have found that any exposure to programming or scripting before this text can be helpful, even if not required for every reader.

This book is designed to be an introductory level Python text for programmers. It may not be an ideal text for someone who has never touched a computer before (for instance, we're not going to spend any time explaining what a computer is), but we haven't made many assumptions about your programming background or education.

On the other hand, we won't insult readers by assuming they are "dummies" either, whatever that means; it's easy to do useful things in Python, and we hope to show you how. The text occasionally contrasts Python with languages such as C, C++, Java, and Pascal, but you can safely ignore these comparisons if you haven't used such languages in the past.

One thing we should probably mention up front: Python's creator, Guido van Rossum, named it after the BBC comedy series *Monty Python's Flying Circus*. This legacy has inevitably added a humorous flavor to many Python examples. For instance, the traditional "foo" and "bar" become "spam" and "eggs" in the Python world, and in some of the code you'll see in this book. The occasional "Brian," "Ni," and "shrubbery" likewise owe their appearances to this namesake. You don't need to be familiar with the series to make sense of such examples (symbols are symbols), but it can't hurt.

---

\* And by "programmer," we mean anyone who has written a single line of code in any programming or scripting language in the past. If you don't meet this test, you will probably find this book useful anyhow. But we'll spend more time teaching Python than programming fundamentals.



## This Book's Scope

Although this book covers all the essentials of the Python language, we've kept its scope narrow in the interest of speed and size. To keep things simple, this book focuses on core concepts, uses small and self-contained examples to illustrate points, and sometimes omits the small details that are readily available in reference manuals. Because of that, this book is probably best described as both an introduction and a stepping stone to more advanced and complete texts.

For example, we won't say much about Python/C integration—a complex topic, which is nevertheless central to many Python-based systems. We also won't talk much about Python's history or development processes. And popular Python applications such as GUIs, system tools, and network scripting get only a short survey, if they are mentioned at all. Naturally, this scope misses some of the big picture.

By and large, Python is about raising the quality bar a few notches in the scripting world. Some of its ideas require more context than can be provided here, and we'd be remiss if we didn't recommend further study after you finish this book. We hope that most readers of this book will eventually go on to gain a more complete understanding of application-level programming from other texts.

Because of its beginners' focus, *Learning Python* is designed to be naturally complemented by O'Reilly's other Python books. For instance, *Programming Python*, Second Edition provides larger and more advanced application-level examples, and was explicitly designed to be a follow-up text to the one you are reading now. Roughly, the second editions of *Learning Python* and *Programming Python* reflect the two halves of Mark's training materials—the core language and applications programming. In addition, O'Reilly's *Python Pocket Reference*, Second Edition, serves as a quick reference supplement for looking up the fine details we will largely skip here.

Other followup Python books can also help provide additional examples, explore specific Python domains, and serve as references. We recommend O'Reilly's *Python in a Nutshell* and New Riders' *Python Essential Reference* as references, and O'Reilly's *Python Cookbook* as an example library. Regardless of which books you choose, you should keep in mind that the rest of the Python story requires studying examples that are more realistic than there is space for here. There are roughly 40 English language Python books available today, along with a few dozen foreign language texts. Because books are a subjective experience, we invite you to browse all available texts to find one that suits your needs.

But despite its limited scope (and perhaps because of it), we think you'll find this book to be a good first text on Python. You'll learn everything you need to get started writing useful standalone Python programs and scripts. By the time you've finished this book, you will have learned not only the language itself, but also how to apply it to day-to-day tasks. And you'll be equipped to tackle more advanced topics and examples as they come your way.

# This Book's Style and Structure

Much of this book is based on training materials developed for a three-day hands-on Python course. You'll find exercises at the end of the last chapter of the core language parts, with solutions to all exercises in Appendix B. The exercises are designed to get you coding right away, and are usually one of the highlights of the course.

We strongly recommend working through the exercises along the way, not only to gain Python programming experience, but also because some exercises raise issues not covered elsewhere in the book. The solutions in Appendix B should help if you get stuck (and we encourage you to “cheat” as much and as often as you like). Naturally, you'll need to install Python to run the exercises.

Because this text is designed to introduce language basics quickly, we've organized the presentation by major language features, not examples. We'll take a bottom-up approach here: from built-in object types, to statements, to program units, and so on. Each chapter is fairly self-contained, but later chapters use ideas introduced in earlier ones (e.g., by the time we get to classes, we'll assume you know how to write functions), so a linear reading makes the most sense. From a broader perspective, this book is divided into the following functional areas, and their corresponding parts.

## Core Language

This portion of the book presents the Python language, in a bottom-up fashion. It is organized with one part per major language feature—types, functions, and so forth—and most of the examples are small and self-contained (some might also call the examples in this section artificial, but they illustrate the points we're out to make). This section represents the bulk of the text, which tells you something about the focus of the book. It is composed of the following parts:

### Part I, *Getting Started*

We begin with a general overview of Python, that answers commonly asked initial questions—why people use the language, what it's useful for, and so on. The first chapter introduces the major ideas underlying the technology, to give you some background context.

This part then begins the technical material of the book, by exploring the ways that both you and Python run programs. Its goal is to give you just enough information to be able to work along with later examples and exercises. If you need more help getting started, additional configuration details are available in Appendix A.

### Part II, *Types and Operations*

Next, we begin our tour of the Python language, by studying Python's major built-in object types in depth: numbers, lists, dictionaries, and so on. You can get a lot done in Python with these tools alone.

### Part III, *Statements and Syntax*

The next part moves on to introduce Python's statements—the code you type to create and process objects in Python. It also presents Python's general syntax model.

### Part IV, *Functions*

This part begins our look at Python's higher-level program structure tools. Functions turn out to be a simple way to package code for reuse.

### Part V, *Modules*

Python modules let you organize statements and functions into larger components, and this part illustrates how to create, use, and reload modules.

### Part VI, *Classes and OOP*

Here we explore Python's object-oriented programming (OOP) tool, the class. As you'll see, OOP in Python is mostly about looking up names in linked objects.

### Part VII, *Exceptions and Tools*

We wrap up the core language coverage of this book section with a look at Python's exception handling model and statements, and a brief overview of development tools. This comes last, because exceptions can be classes if you want them to be.

## Outer Layers

Part VIII samples Python's built-in tools, and puts them to use in a collection of small example programs. Common tasks are demonstrated in Python to give you some real-world context, using both the language itself, and its standard libraries and tools.

### Chapter 27, *Common Tasks in Python*

This chapter presents a selection of the modules and functions that are included in the default Python installation. By definition, they comprise the minimum set of modules you can reasonably expect any Python user to have access to. Knowing the contents of this standard toolset will likely save you weeks of work.

### Chapter 28, *Frameworks*

This chapter presents a few real applications. By building on the language core explained in earlier parts and the built-in tools described in Chapter 27, we present many small but useful programs that show how to put it all together. We cover three areas that are of interest to most Python users: basic tasks, text processing, and system interfaces. We close with a brief discussion of Jython, the Java port of Python, and a substantial Jython program.

### Chapter 29, *Python Resources*

This chapter discusses the layers of the Python community and specialized libraries that are either part of the standard Python distribution or freely available from third parties.

## Appendixes

The book ends with appendixes that give platform-specific tips for using Python on various platforms (Appendix A), and provide solutions to exercises that appear at the end of the last chapter in each part (Appendix B). Note that the index and table of contents can be used to hunt for details, but there are no reference appendixes in this book. As mentioned earlier, the *Python Pocket Reference*, Second Edition (O'Reilly), as well as other books and the free Python reference manuals maintained at <http://www.python.org>, fill in syntax and built-in tool details.

## Book Updates

Improvements happen (and so do mis<sup>^</sup>H<sup>^</sup>H<sup>^</sup>H typos). Updates, supplements, and corrections for this book will be maintained (or referenced) on the Web at one of the following sites:

- <http://www.oreilly.com> (O'Reilly's site)
- <http://www.rmi.net/~lutz> (Mark's site)
- <http://starship.python.net/~da> (David's site)
- <http://www.python.org> (Python's main site)
- <http://www.rmi.net/~lutz/about-lp.html> (book's web page)

If we could be more clairvoyant, we would, but the Web changes faster than printed books.

## Font Conventions

This book uses the following typographical conventions:

### *Italic*

For email addresses, filenames, URLs, for emphasizing new terms when first introduced, and for some comments within code sections

### Constant width

Shows the contents of files or the output from commands and to designate modules, methods, statements, and commands

### Constant width bold

In code sections to show commands or text that would be typed

### *Constant width italic*

Shows replaceables in code sections

### <Constant width>

Represents syntactic units that you replace with real code



Indicates a tip, suggestion, or general note relating to the nearby text.



Indicates a warning or caution relating to the nearby text.

In our examples, the `%` character at the start of a system command line stands for the system's prompt, whatever that may be on your machine (e.g., `C:\Python22>` in a DOS window). Don't type the `%` character yourself! Similarly, in interpreter interaction listings, do not type the `>>>` and `...` characters shown at the start of lines—these are prompts that Python displays. Type just the text after these prompts. To help you remember this, user inputs are shown in bold font in this book. Also, you normally don't need to type text that starts with a `#` in listings; as we'll explain later, these are comments, not executable code.

## About the Programs in This Book

This book, and all the program examples in it, are based on Python Version 2.2, and reflect the upcoming 2.3 release. But since we'll stick to the core language, you can be fairly sure that most of what we have to say won't change very much in later releases of Python. Most of this book applies to earlier Python versions too, except when it does not; naturally, if you try using extensions added after the release you've got, all bets are off. As a rule of thumb, the latest Python is the best Python. Because this book focuses on the core language, most of it also applies to Jython, the Java-based Python language implementation, as well other Python implementations, described in Chapter 2.

Source code for the book's examples, as well as exercise solutions, can be fetched from the book's web site at <http://www.oreilly.com/catalog/lpython2/>. So how do you run the examples? We'll get into startup details in Chapter 3, so please stay tuned for the details on this front.

## Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example

code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*ActionScript: The Definitive Guide*, Second Edition, by Colin Moock. Copyright 2001 O'Reilly & Associates, Inc., 0-596-00369-X.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).

## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly & Associates, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
(800) 998-9938 (in the United States or Canada)  
(707) 829-0515 (international or local)  
(707) 829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

<http://www.oreilly.com/catalog/lpython2>

To comment or ask technical questions about this book, send email to:

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

<http://www.oreilly.com>

Mark and David are also both happy to answer book questions from readers, though you're more likely to get a response by sending “Core Language” questions to Mark, and “Outer Layer” queries to David, the two area's respective primary authors. You can find both of the authors' email addresses at the book's web site.

(Throughout this book, we normally use “we” to refer to both authors, but occasionally slip into a specific author's name for personal anecdotes—usually Mark in the Core Language parts, and David in the Outer Layers parts, reflecting the primary author of each part. Although this book was a joint effort of many, each author sometimes steps out of the collective.)

# Acknowledgments

We'd like to express our gratitude to all the people who played a part in developing this book. First of all, we'd like to thank the editors that worked on this project: Laura Lewin, Paula Ferguson, and finally Linda Mui. We'd also like to thank O'Reilly in general, for supporting another Python book project. We're glad to be a part of what is now a full and growing Python product line at O'Reilly.

Thanks also to everyone who took part in the early review of this book—Guido van Rossum, Alex Martelli, Anna Ravenscroft, Sue Giller, and Paul Prescod.

And for creating such an enjoyable and useful language, we owe an especially large debt to Guido, and the rest of the Python community; like most open source systems, Python is the product of many heroic efforts.

We'd also like to give a special thanks to our original editor on this book, the late Frank Willison. Frank had a profound impact on both the Python world, and our own personal careers. It is not an overstatement to say that Frank was responsible for much of the fun and success of Python's early days. In fact, this very book was his idea. In recognition of his vision and friendship, we dedicate this update to him. Hack on, Frank.

## Mark Also Says:

When I first met Python in 1992, I had no idea what an impact it would have on the next decade of my life. After writing the first edition of *Programming Python* in 1995, I began traveling around the country and world teaching Python to both beginners and experts. Since finishing the first edition of this book in 1999, I've been a full-time, independent Python trainer and writer, thanks largely to Python's exponentially growing popularity.

As I write these words in early 2003, I've taught roughly 90 Python training sessions, in the U.S., Europe, Canada, and Mexico, and met over one thousand students along the way. Besides racking up frequent flyer miles, these classes helped me refine my contributions to this book, especially the core language material. These parts of the book mostly come straight from my current course materials.

I'd like to thank all the students who have participated in my courses over the last seven years. Together with recent Python changes, your feedback played a huge role in shaping my contributions to this text. There's nothing quite as instructive as watching one thousand students repeat the same beginners' mistakes! The core language section of this second edition owes its changes primarily to classes held after 1999; I'd like to single out Hewlett-Packard, Intel, and Seagate for multiple sessions held in this timeframe. And I'd especially like to thank the clients who hosted classes in Dublin, Mexico City, Barcelona, and Puerto Rico; better perks would be hard to imagine.

I'd like to thank O'Reilly for giving me a chance to work on now six book projects; it's been net fun (and only feels a little like the movie *Groundhog Day*). I also want to thank coauthor David Ascher, for his work and patience on this project. Besides this book and his day job developing Python tools at ActiveState, David also donates his time to organizing conferences, editing other books, and much more.

Finally, a few personal notes of thanks. To all the people I worked with at various companies earlier in my career. To the Boulder public library, in which I hid while writing parts of this edition. To the late Carl Sagan, for inspiration in youth. To Jimmy Buffet, for perspective at the dawn of middle age. To a woman from New Mexico on a flight from Oklahoma, for reminding me of the importance of having a dream. To the Denver Broncos, for winning the big one (twice). To Sharp and Sony, for making such sweet machines. And most of all, to my children, Michael, Samantha, and Roxanne, for making me a truly rich man.

*Longmont and Boulder, Colorado  
July 2003*

## **David Also Says:**

In addition to the previous thanks, I'd like to extend special thanks to the following.

First, thanks to Mark Lutz for inviting me to work with him on this book and for supporting my efforts as a Python trainer. Additional thanks to the impressive array of Python folks who encouraged me in my early days understanding the language, especially Guido, Tim Peters, Don Beaudry, and Andrew Mullhaupt. It's amazing how a little encouragement at the right time can have long-lasting impact.

I used to teach Python and Jython, much like Mark still does. The students in these courses have helped me identify the parts of Python that are the trickiest to learn, as well as remind me of the aspects of the language that make it so pleasant to use, and I thank them for their feedback and encouragement. I would also like to thank those who gave me the chance to develop these courses: Jim Anderson (Brown University), Cliff Dutton (then at Distributed Data Systems), Geoffrey Philbrick (then at Hibbitt, Karlsson & Sorensen, Inc.), Paul Dubois (Lawrence Livermore National Labs), and Ken Swisz (KLA-Tencor). While I'm no longer regularly teaching Python, that experience is one I rely on still when coaching novices.

Thanks to my scientific advisors, Jim Anderson, Leslie Welch, and Norberto Grzywacz, who have all kindly supported my efforts with Python in general and this book in particular, not necessarily understanding why I was doing it but trusting me nonetheless. Any of the lessons they taught me are still relevant daily.

The first victims of my Python evangelization efforts deserve gold stars for tolerating my most enthusiastic (some might say fanatical) early days: Thanassi Protopapas, Gary Strangman, and Steven Finney. Thanassi also gave his typically useful feedback on an early draft of the book. Several Activators (known to civilians as "ActiveState



employees”) have been tremendous colleagues and friends these last three years—I’ll single out Mark Hammond, Trent Mick, Shane Caraveo, and Paul Prescod. Each of them have taught me much, about Python and otherwise. ActiveState as a whole has provided me with an amazing environment in which to build a career, learn new things from fascinating people every day, and still program in Python.

Thanks to my family: my parents JacSue and Philippe for always encouraging me to do what I want to do; my brother Ivan for reminding me of some of my early encounters with programming texts (after hours of effort, realizing that a program listing in Byte magazine was buggy had a 13-year-old boy crying out of frustration); my wife Emily for her constant support and utter faith that writing a book was something I could do; our children, Hugo and Sylvia, for sharing the computers with me—they approach computers with such ease that I can’t wait to see what their generation comes up with.

Finally, thinking about this edition in particular, I want to thank everyone who has contributed to the Python community. It is striking to compare Python now with Python five years ago—the language has changed a little, while the world in which it lives is so much broader and richer. It bubbles with enthusiasm, code, and ideas (from amazingly bright people and crackpots alike), while remaining respectful and cheerful. Let’s keep on doing that.

*Vancouver, British Columbia, Canada  
November 2003*