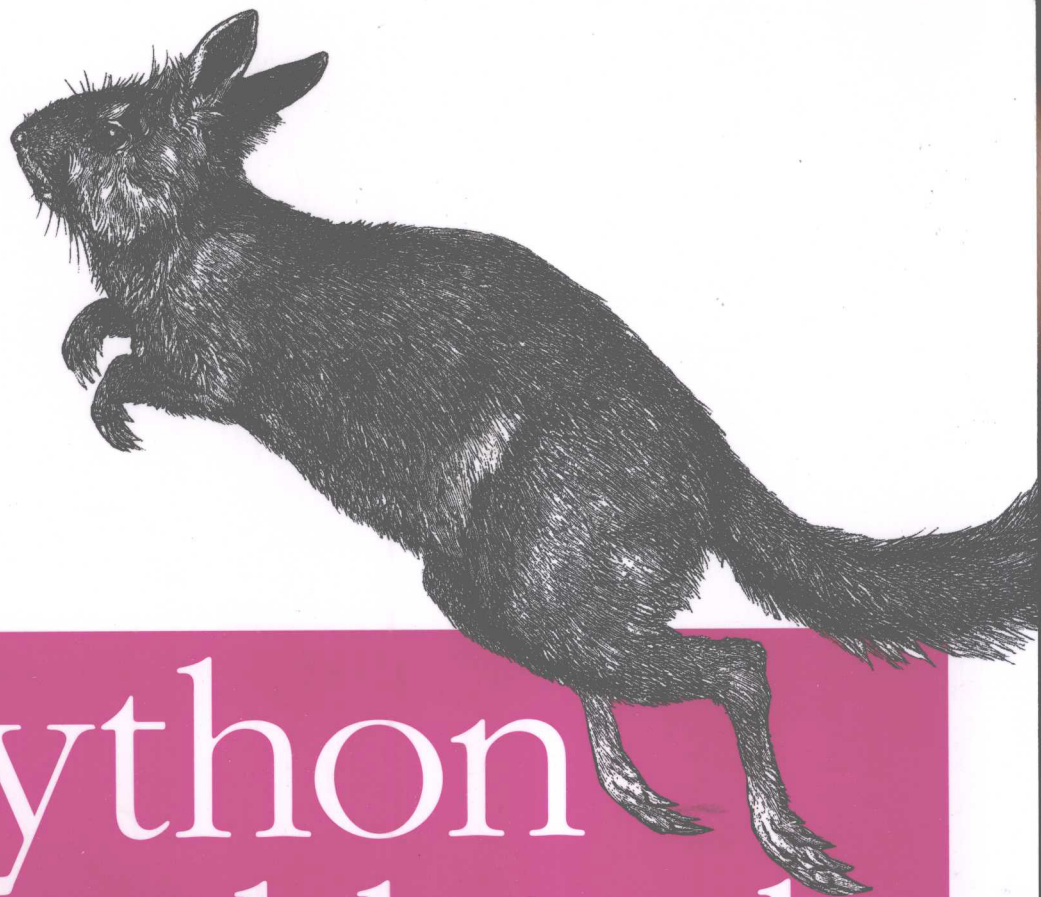


Python 社区集体智慧的结晶



Python Cookbook™

(第2版) 中文版

Alex Martelli

[美] *Anna Martelli Ravenscroft* 编

David Ascher

高铁军 译

O'REILLY®

 人民邮电出版社
POSTS & TELECOM PRESS

O'REILLY®

Python Cookbook™

(第2版) 中文版

[美] Alex Martelli Anna Martelli Ravenscroft David Ascher 编

高铁军 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Python Cookbook (第2版) 中文版 / (美) 马特利 (Martelli, A.), (美) 马特利 (Martelli, A.), (美) 阿舍尔 (Ascher, D.) 编; 高铁军译. -- 北京: 人民邮电出版社, 2010. 5
ISBN 978-7-115-22266-4

I. ①P… II. ①马… ②阿… ③高… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆CIP数据核字 (2010) 第022417号

版权声明

Copyright©2005 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2010. Authorized translation of the English edition, 2005 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.

本书中文简体版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

Python Cookbook (第2版) 中文版

-
- ◆ 编 [美] Alex Martelli Anna Martelli Ravenscroft
David Ascher
 - 译 高铁军
 - 责任编辑 刘映欣
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 787×1000 1/16
印张: 49.5
字数: 1031 千字 2010 年 5 月第 1 版
印数: 1-4 000 册 2010 年 5 月北京第 1 次印刷

著作权合同登记号 图字: 01-2009-1822 号

ISBN 978-7-115-22266-4

定价: 99.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223
反盗版热线: (010)67171154

内 容 提 要

本书介绍了 Python 应用在各个领域中的一些使用技巧和方法，从最基本的字符、文件序列、字典和排序，到进阶的面向对象编程、数据库和数据持久化、XML 处理和 Web 编程，再到比较高级和抽象的描述符、装饰器、元类、迭代器和生成器，均有涉及。书中还介绍了一些第三方包和库的使用，包括 Twisted、GIL、PyWin32 等。本书覆盖了 Python 应用中的很多常见问题，并提出了通用的解决方案。书中的代码和方法具有很强的实用性，可以方便地应用到实际的项目中，并产生立竿见影的效果。尤为难得的是，本书的各位作者都具有丰富的业界实践经验，因此，本书不仅给出了对各种问题的解决方案，同时还体现了很多专家的思维方式和良好的编程习惯，与具体的细节性知识相比，这部分内容无疑是本书的精华。

本书适合具有一定 Python 基础的读者阅读参考。

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权人民邮电出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

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，同时也是联机出版的先锋。

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。

译者序

我认为 Python Cookbook 是一本很独特的书。很多 Python 的书籍，其内容都具有一定的相似性和重叠，比如 Python 的学习书籍和教材，我们会想起 Core Python Programming、Python Programming 等，说到 Python 的系统管理，会想起 Python for UnIX and Linux System Administration、Pro Python System Administration 等书，说起 Python 的 XML 处理，会想起 XML Processing with Python 和 Python & XML，而像 Python Cookbook 这样的书，则是独一无二的，我想不出另外一本和它相似的书。它的独特，体现在几个方面。首先，它不是由一个或几个人写的，而是由几百人贡献的心得和技巧综合而成，所以，它包含了各种思考的方式和看待问题的角度。这些作者中，既有普通的爱好者，也有一些名气极大的技术“牛人”们（如 Guido van Rossum、Tim Peters、Raymond Hettinger、Mark Hammond 等）。他们都有着极其丰富的工程实践经验，因此，他们给出的技巧心得都非常实用，具有极强的可操作性，读者可以轻易地修改书中的示例并应用于自己的项目。另一方面，这些作者对 Python 的理解非常深入，他们不仅仅提出了对问题的解决之道，同时也在字里行间注入了 Python 的思维方式，即解决问题只是一方面，更重要的是怎么用更 Pythonic 的方式解决问题，怎么赋予解决过程更多的美感。比起具体的各种问题的解决之法，我认为这才是本书最精华的部分。最后，此书各个部分并没有严密的逻辑联系，所以，看书的方式也可以很随意，读者既可以从头到尾循序渐进地逐章逐节阅读，也可以跳跃式阅读，甚至可以随机抽取章节阅读，一切悉听尊便，只要适合个人习惯即可。

这本书主要是针对 Python 2.4，同时也考虑到了对 Python 2.3 的支持。不过自从此书出版之后，Python 的发展也似乎得到了加速，2006 年 8 月，Python 2.5 发布，2008 年 10 月，Python 2.6 发布，仅仅几个月之后，Python 3.0 横空出世，现在最新的版本已经是 3.1 了。有人可能会担心，现在这本书的内容会不会太陈旧了。我认为其实无须担心，请参考 <http://www.python.org> 列出的各个版本的 What' New。从 Python 2.4 到 Python 2.5，一些新的模块被加入了，比如 xml.etree、sqlite 以及 ctypes，还包括了一些新的特性，比如条件表达式的新写法、with 语句、统一的 try/except/finally 等，此外还有几百个补丁以及 bug 的修正。至于从 Python 2.5 到 Python 2.6 的变化，根据 A.M. Kuchling 的说法，2.6 主要是为 Python 3.0 做铺垫的，Python 2.6 在保持对以前代码的兼容性的同时还包含了 3.0 的新特性和新语法。所以，至少 99% 的 Python 2.4 的代码在 Python 2.6 中仍可以通行无阻。虽然 Python 3.0 和 Python 3.1 不再考虑对 Python 2.x 代码的向后兼容性了，但绝大多数的 Python 2.x 的代码都可以略作修改继续运行。在 3.x 中，一些语法或

者内建函数和对象的行为略有调整，但核心的机制和特性并未改变，所以此书中的大多数技巧仍然有效。关于 Python 2.5、2.6 以及 3.0 和 3.1 的 What' New，请参看 <http://docs.python.org/dev/whatsnew/2.5.html>、<http://docs.python.org/dev/whatsnew/2.6.html>、<http://docs.python.org/dev/py3k/whatsnew/3.0.html> 和 <http://docs.python.org/dev/py3k/whatsnew/3.1.html>。

本书显然不是为程序的初学者准备的，甚至不是为有其他编程语言经验的初次接触 Python 的读者准备的。当然，阅读本书并不要求读者首先成为 Python 语言的专家，只要有 Python 编程语言的基础就可以了，如果读者已经拥有了其他语言的编程经验，或者有很多 Python 编程的经验，那本书就更适用了。

最后，由于本人才疏学浅，虽然竭力想做到译文准确流畅（优美就更不奢望了），但终究力所不逮，不妥和疏漏肯定是难以避免的。因此，我非常欢迎读者的批评指正和宝贵意见。

前言

这本书不是一本典型的 O'Reilly 风格的书，而是一本集合了多个作者的手稿的作品。实际上，这也是一种将开源开发的方式应用到书籍出版业的尝试。Python 社区有超过 300 个成员在本书中贡献了他们的心得和资料。在这里，我们作为编辑，想给你——本书的读者，介绍一些重要的背景资料，这些背景资料是关于此书是如何编著出来，以及这个过程和涉及的人，并提出一些关于这种崭新的风格的思考。

本书的构思

在 2000 年初，O'Reilly 的主编 Frank Willison 联系到我 (David Ascher)，问我是否想写一本书。Frank 曾是 *Learning Python* 一书的编辑，那本书是由我和 Mark Lutz 合著而成。由于当时我正在致力于 *Perl shop* (ActiveState) 一书的编写，实在没有精力再同时开始另一本书的工作。但我和 Frank 总是定期地通过 E-mail 或者聊天工具对该书的一些主题进行探讨。Frank 有个主意，他受到 Tom Christiansen 和 Nathan Torkington 编写的 *Perl Cookbook* 一书的启发，认为出一本类似的 *Python Cookbook* 将会很有趣。Frank 想重复 *Perl Cookbook* 一书的成功，但他也认为应该让更多的人参与到此书的编写中。他的想法是，作为一本真正的 cookbook，更多人的参与将会提供更多的思路、角度和品味。至于书籍的质量，他认为可以通过技术编辑的把关和 O'Reilly 严格的编辑流程来保障。

Frank 和 ActiveState 的 CEO Dick Hardt 意识到 Frank 的目标和 ActiveState 的目标其实是殊途同归的。ActiveState 的想法是为开源程序员创建一个网络社区，叫做 ActiveState Programmer's Network (ASPN)。ActiveState 有一个人气很高的网站，提供了很丰富和多样化的内容。ActiveState 充分认识到开源社区具有很强大的力量，它能够提供最最新和最准确的内容，无论多么生僻的方面均有涉及。

O'Reilly 和 ActiveState 很快认识到他们的目标上的相似性，并且意识到双方合作是最好的达成目标的方式。他们想要实现的包括：

- 创建一个由 Python 程序员提供素材的在线 Python Cookbook，为 Python 程序员服务；
- 出版一本包括了最好素材的书，该书由 Python 社区中的一些关键人物提供的材料和想法综合而成；
- 双方通过完成这本书亦可学习到一种不同的著书模式。

与此同时，有两件事情也需提及。首先，ActiveState 的一些人，包括 Paul Prescod，开始积极地寻找一些星级人物来加入 ActiveState 的开发小组中。其中一位候选者是著名的 Alex Martelli（但那时我们还没听说过他）。Alex 出名的原因是他在 Python 邮件列表中发表过大量透彻全面的文章，同时他也具有极好的耐心来解释 Python 的各种细微精巧之处，而且他非常高兴通过他自身的努力为 Python 带来了更多的新用户。我们不知道他的原因是，他住在意大利，而且对于 Python 社区来说他还算是一个新用户，Python 老手们也从来没见过他。但其实在 20 世纪 80 年代的时候，Alex 居住在美国，为 IBM 研发中心工作，并且很狂热地使用和推介其他的一些高阶语言（那时，大多是 IBM 的 Rexx）。

ActiveState 热切地期望 Alex 的到来，并试图劝说他搬到温哥华。我们已经是如此接近成功，可惜他的雇主给他带上了个金手铐，而且温哥华的天气明显无法与意大利的宜人气候竞争。所以，最后 Alex 仍然留在意大利，这让我很失望。即便如此，Alex 当时仍然在与 O'Reilly 接洽和商议写一本书。Alex 想写一本 cookbook 类型的书，但 *Python Cookbook* 当时已经签约。所以后来 Alex 和 O'Reilly 签了另外一本书 *Python in Nutshell* 的合同。

第二件同时进行的事就是创建 Python 软件基金会。由于各种各样的原因，这件事被当做最好的部分，在某个会议的结束酒会上被提出来，Python 社区的成员想要创建一个非盈利性的组织，作为各种各样 Python 作品的知识产权的拥有者，这样可以确保 Python 及其衍生的作品有更加坚实的法律支撑。当然，这样的组织需要经济上的支持和 Python 社区的支持才能成功。

在综合考虑当时的各种情况之后，各方达成了以下协议：

- ActiveState 将会创建一个在线 cookbook，提供一种机制让任何人可以提交技巧（比如，针对某个特定问题的一段 Python 代码，以及相关的一些探讨，用以阐明为何以及在什么条件下使用这个窍门）。为了激发作者的积极性和读者的互动性，这个网站还允许读者提出修改建议和问题。
- 作为我的 ActiveState 工作的一部分，我将编辑和确保这些材料的质量。Alex Martelli 也加入了这个项目，作为合作编辑，为那些将要出版的材料把关，另外还有 Anna Martelli Ravenscroft，我们一起作为此书第 2 版的主要编辑。
- O'Reilly 将那些最好的技巧和心得集中起来出版，形成本书。
- 作为本书作者版税，此书售卖的部分收入将捐赠给 Python 软件基金会。

本书的实现

在线 cookbook（在 <http://aspn.activestate.com/ASPN/Cookbook/Python/>）是所有的技巧和心得材料的一个入口。读者可以通过填写表格和一些相关信息来注册免费的账户。那

些材料现在已经变成了本书的一部分。成千上万的用户阅读过那些材料，并且给出了相应的评论，因此，在本书的出版发行过程中，相关的材料已经变得越来越成熟和丰富。同时，作者的名字被印在书中也会吸引更多的人去访问在线 `cookbook`。随着每月新的技巧和心得的持续增加，以及邮件列表中越来越多的人频繁地引用在线 `cookbook`，可以期望这个在线 `cookbook` 也会取得极大的成功，这将再次证明作者的工作带给读者的巨大价值。

既然已经拥有了网站中的大量素材，那么实现这本书的工作就变成了挑选、合并、排序以及一些编辑工作了。这部分工作的一些细节将会在前言的“组织方式”部分介绍。

使用本书中的代码

本书的目的是为了帮助你更好地完成工作。一般而言，可以随意在自己的程序或者文档中使用书中代码。不需要联系我们来获得许可，除非使用了相当大一部分代码作为产品的一部分。比如，使用书中的代码写几个程序，无须获得许可。但是贩卖含有本书的代码的光盘则需要获得许可。回答问题时引用本书中的代码和文字不需要获得许可。而从本书中择取一定数量的代码用于你自己产品的文档则需要获得许可。我们感激但不强求使用者给出出处。一个完整的出处通常包括了标题、作者、出版商以及 ISBN。举个例子：“*Python Cookbook*, 2nd ed., by Alex Martelli, Anna Martelli Ravenscroft, and David Ascher (O’Reilly Media, 2005) 0-596-00797-3”。如果你觉得使用代码的情况可能符合需要获取许可的条件，请通过 permissions@oreilly.com 与我们联系。

本书的读者

我们希望至少能够懂一点 Python。本书并不是一本循序渐进教授 Python 语言的书。实际上，它针对某些特定的任务和问题，提出一些特别的技术和概念（有时是花招）来应对。如果想找一本完整的介绍 Python 语言的书，可以考虑前言中“参考资料”部分里面提到的书。不过也不需要完全了解 Python 就能从本书获益。一些章节展示了完成一些基本和通用任务的最佳技术，而另一些章节则展示了更加复杂和特殊的技术。我们在书中加了很多边栏来提示一些书中提到的概念，以便将它们和你听到的一些概念区分开来，但你也或许仍会感到混淆。所以，这绝对不是一本给初学者看的书。本书针对的最大的读者群应该是 Python 社区的人，他们中有很多很棒的程序员，他们既不是刚上手的新人，也不是 Python 用得入神入化的高手。如果你已经对 Python 有了很多了解，本书仍可能给你一个惊喜，因为我们也把一些最新的和不为人所知的领域的高级技巧收进了本书。你会学到一些新的东西，就像我们一样。无论你在 Python 技术方面处于哪个级别，我们都相信你从本书获得一些有价值的东西。

如果你已经有了本书的第 1 版，可能会问，我还需要购买第 2 版吗？我们认为，答案是“是的”。第 1 版有 245 节；我们只保留了其中的 146 个（而且绝大多数都重新编辑过），并增加了 192 个新的，这样第 2 版中总共有 338 节。所以，本书中有超过半数内

容是全新的，而且所有的例子都被更新和修改过，以便适用于 Python 2.3 和 2.4。这也是本书比第 1 版多了约 100 节却仍能保持页数大致相同的主要原因。第 1 版覆盖的 Python 版本包括了 1.5.2（相当的老的版本）到 2.2；本书则集中于 2.3 和 2.4。这应当归功于如今 Python 的强大表现力，而且我们也避免和删除了一些为了兼容 5 年甚至更久之前的 Python 的“历史包袱”般的内容，因此我们能够用大致相同的篇幅，提供更多有价值 and 更实用的内容。

组织方式

本书有 20 章。每章都集中提供某一特定类型的技巧和方法，比如有关算法的、文本处理的以及数据库操作的等。第 1 版有 17 章。不过 Python 也在不断进步，无论是语言本身还是相关的库都在不断变化，另外 Python 社区发表在 cookbook 在线的文章也在不停地更新和增加，所以我们加入了全新的 3 个章节：在 Python 2.3 中引入的迭代器(iterator)和生成器(generator)；关于 Python 对时间和财务的操作，新旧并呈；Python 2.2 中导入了新的工具和类型（自定义描述符(custom descriptor)，装饰器(decorator)，元类(metaclass))。每章包括一个总体介绍，由这方面的一个专家执笔，然后是从在线 cookbook 中撷取的技巧和方法（其中有大约 5% 的内容是专为本书撰写的），并被编辑和修改，以符合全书的整体格式与风格。Alex（在 Anna 的帮助下）主要负责对第 1 版的内容进行筛选，确定哪些需要保留，哪些需要更新，并从网站的近 1000 个材料中挑选新的内容，再进行合并或修改（所以，如果你发现在线 cookbook 提交的内容和印刷出来的版本不一样，找 Alex 去吧，一定是这家伙干的）。他也需要确定涵盖哪些主题，但有时他可能难以找到符合主题的好材料，所以某些内容并没有被选入，或者最后选用了一些不是从在线 cookbook 中找来的材料，这也不完全是他的错。

当材料的选择工作完成之后，就进入了编辑及合并的工作，有时还需要把在线 cookbook 上的一些重要的相关评论融合到内容中。这的确是个挑战，就像此书的第 1 版一样，甚至更具挑战性。那些技巧和方法涵盖范围极广，复杂度、完整度和组织方式也相应变化。此书涉及了超过了 300 名作者，他们每个人的声音和风格都包含在书中。我们必须设法让本书能够容纳多种风格，以反映出本书最本质的特点——由整个 Python 社区集体完成。当然，我们也必须小心翼翼地编辑工作，以确保内容尽可能地易于使用和查阅，同时还要维持着全书统一的结构和呈现方式。绝大多数材料，包括本书第 1 版和在线 cookbook 中的材料，都必须先根据当前情况更新，有的甚至还要重写。不过，通过借助新工具和更好的方法，这个工作完成得很顺利。然后，我们要仔细考虑各章节的排列顺序、每个章节中的布局以及相关技巧和方法的排列顺序。我们做这种排序的工作主要是为了让本书更加易于使用，无论是对刚上手的新人还是 Python 老手都是如此，同时，对于各种不同的阅读习惯，无论是按照逐字逐页的顺序阅读本书，还是为了重点加强某个领域的知识而跳跃性地阅读，甚至是漫无目的地随机阅读，我们都希望我们的排序和组织方式能够对读者有所帮助。

虽然本书很适于“跳跃性”阅读，我们仍然相信，花一些时间循序渐进地一览全书，会为你节约更多的时间。在这次初览中，可以跳过那些你觉得难的，或者不感兴趣的章节。虽然你跳过了一些内容，但是仍可以粗略地领略到本书的构建方式以及各个主题是如何覆盖和解释的，这将为你以后更深层次的精读打好基础，如果愿意，随机的跳跃性的阅读当然也没问题。为了帮助你更好地了解本书的内容，我们给出了各个章节的主要内容介绍，以及为各个章节写下引言的 Python 专家的介绍。

第 1 章，引言由 Fred L. Drake, Jr 撰写。

本章包含了操纵文本的一系列方法和窍门，包括了合并、过滤、格式化字符串以及在整个文本中对部分字符串的替换，还有 Unicode 的处理。

Fred Drake 是 PythonLabs group 的成员，一直致力于 Python 的开发。他是 3 个孩子的父亲，Fred 在 Python 社区很有名气，因为他一人独立完成了整个官方文档的维护。Fred 还是 *Python & XML* 一书的合著者。

第 2 章，引言由 Mark Lutz 撰写。

本章呈现了操作文件中的数据以及操纵文件系统下的文件和目录的一些技术，还包括了处理特定的文件格式以及存档格式的内容，比如 tar 和 zip 格式。

Mark Lutz 以著作多而闻名，他的书包括了 *Programming Python*、*Python Pocket Reference* 以及 *Learning Python*（都由 O'Reilly 出版），最后一本由他和 David Ascher 合作完成。Mark 同时还是一流的 Python 培训教师，为 Python 在全世界的传播和流行尽心尽力。

第 3 章，引言由 Gustavo Niemeyer 和 Facundo Batista 撰写。

本章（第 2 版中的新章节）介绍了处理日期、时间、十进制数以及一些和财务相关的问题的工具和技术。

Gustavo Niemeyer 是第三方模块 *dateutil* 的作者，他参与了很多 Python 的扩展和项目的开发。Gustavo 住在巴西。Facundo Batista 是 *Decimal* PEP 327 的作者，同时也是标准库模块 *decimal* 的作者，由于这个模块，Python 2.4 才支持浮点十进制计算。他住在阿根廷。作为本书的编辑，我们非常高兴能够请到他们俩为本章撰写引言。

第 4 章，引言由 David Ascher 撰写。

本章包括了一些很通用的，可以应用到各处的技术，这些内容难于归结到某个类别。

David Ascher 是本书的编辑之一。David 的背景颇为斑驳，他做过物理研究、视觉研究、科学可视化、计算机图形学，还摆弄过一堆编程语言，他还是 *Learning Python* 一书的合著者，也做过 Python 的教学工作，最近，他又接手了一些非技术性的任务，比如管理 *ActiveState* 的团队。David 还定期地组织 Python 会议。

第 5 章，引言由 Tim Peters 撰写。

本章覆盖了 Python 中的搜索和排序技术。很多例子展示了将稳定快速的 `list.sort` 和 `decorate-sort-undecorate` (DSU) (在 Python 2.4 中新导入的能力) 结合在一起的创造性的应用，其余例子还展示了 `heapq`、`bisect` 的威力，并介绍了 Python 中其他的搜索和排序工具。

Tim Peters 以 `tim-bot` 的诨号为人所知，他是 Python 界中的一个传奇性的人物。他犹如一个导师或先知，在 Guido van Rossum 抽不开身的时候，他能够化身为另一个 Guido van Rossum 来指引方向；当有人略微提及有关 IEEE 标准的问题时，他又能够化身为 IEEE-754 浮点标准委员会，引经据典，滔滔不绝；当有人试图鼓吹对 Python 进行某些激进的改革时，他还能够适时地表现出中流砥柱的一面，稳健而坚定。Tim 也是 PythonLabs 团队中的一员。

第 6 章，引言由 Alex Martelli 撰写。

本章的内容展示了在 Python 中应用面向对象编程模式的威力，包括了一些很重要的技术，比如通过一些特殊方法来托管和控制属性，用一些中级的技术来实现不同的设计模式，还有一些对于高级概念的简单应用，比如自定义元类，关于元类的有关内容在第 20 章会有更深入的探讨。

Alex Martelli，外号 `martelli-bot`，是本书的编辑之一。他在 IBM 研发中心工作过约十年，然后又在 `think3, inc.` 工作了更久的时间。Alex 现在是自由职业的顾问，最近在为 AB Strakt 工作，那是个以 Python 为业务中心的瑞典公司。他有时也编辑 Python 文章或者著书，比如 *Python in a Nutshell*，偶尔，他还会研究一下合约桥牌。

第 7 章，引言由 Aaron Watters 撰写。

本章着重介绍了 Python 的持久化技术，包括序列化的方法以及和不同的数据库交互的方式。

Aaron Watters 是最早的 Python 倡导者之一，也是数据库专家。他作为第一本 Python 书 (*Internet Programming with Python*, M&T Books, 现在已经绝版) 的主要作者而闻名遐迩，他也是很多广为流传的 Python 扩展的作者，比如 `kjBuckets` 和 `kwParsing`。Aaron 目前也是自由职业的顾问。

第 8 章，引言由 Mark Hammond 撰写。

本章介绍了一系列有关调试和测试的方法和窍门，包括了可定制错误日志和回溯信息，以及使用可定制模块 (`unittest` 和 `doctest`) 进行单元测试。

为了使 Windows 平台支持 Python，Mark Hammond 做了很多工作并因此而出名。他和 Greg Stein 创建了一个包含很多模块的强大的库，这个库将 Python 完全地映射到了 Windows 的一系列 API、库甚至组件模型，如 COM。他是个专家级的设计者，还是很

多开发工具的作者，其中最为人所知的工具就是 Pythonwin 和 Komodo。最后，Mark 还是调试高手，在 Komodo 开发中，常常面对各种棘手的状况，他有时也被邀请去调试涉及三种语言 (Python、C++和 Javascript) 的、多线程、多进程的系统。Mark 和 Andy Robinson 合著了 *Python Programming on Win32* 一书。

第 9 章，引言由 Greg Wilson 撰写。

本章涵盖了一系列同步编程技术，包括了线程、队列以及多进程。

Greg Wilson 写过儿童书籍，也写过并行编程和数据处理的书。当他不写书的时候，他是 Doctor Dobb's Journal 的得力编辑，多伦多大学计算机科学系的助理教授，同时还是个自由职业的软件开发者。Greg 是 Software Carpentry 项目的最初发起者和推动者，最近他又得到 Python 软件基金会的认可，为需要进行科学计算的科学家和工程师开发一系列具有 Python 特色的课程材料。

第 10 章，引言由 Donn Cave 撰写。

本章介绍了利用 Python 进行一些通用的系统管理任务的内容，包括生成密码和操作 Windows 的注册表，以及处理信箱和 Web 服务器的问题。

Donn Cave 是华盛顿大学计算中心的软件工程师。多年以来，Donn 已经在 comp.lang.python 充分证明了他在有关系统调用、UNIX、系统管理、文件、信号等方面丰富的知识和技能。

第 11 章，引言由 Fredrik Lundh 撰写。

本章包括了一些通用的界面任务，主要使用 Tkinter，但也涉及了一些 wxPython、Qt、图像处理，以及和 Jython (用于 JVMJava 虚拟机) 相关的一些特定的界面处理，还有 Mac OS X 和 IronPython (用于 dotNet)。

Fredrik Lundh 也以 eff-bot 之名著称，他是 Secret Labs AB 的 CTO，该公司主要提供一些基于 Python 的产品和技术。Fredrik 是 Tkinter (Python 最流行的界面工具包) 领域的一流专家，也是 Python 图形库 (Python Image Library, PIL) 的主要作者。他还是 *Python Standard Library* 一书的作者，该书对 Python 的标准库进行了全面的介绍，可以说是本书非常有益的补充。他在 comp.lang.python 发文极多，极大地帮助了广大的 Python 新手，甚至很多老手和专家也从他的文章中获益良多。

第 12 章，引言由 Paul Prescod 撰写。

本章介绍了如何使用一系列 Python 工具来分析、处理以及生成 XML。

Paul Rescod 在三种技术方面完全可以称得上是专家：Python，在这个方面他已经无须向世人来证明什么了；XML，他在 XML 的应用上的功力也毋庸置疑 (Paul 和 Charles Goldfarb 合作，完成了 *XML handbook* 一书)；Unicode，不像前面两种技术，Unicode 总是让广大程序员痛苦而迷惘。Paul 目前是 Blast Radius 的产品经理。

第 13 章，引言由 Guido van Rossum 撰写。

本章包括了各种网络编程技术，从基本的 TCP 客户端和服务端，一直到操纵 MIME 消息的技术。

Guido 创造了 Python，并在 Python 的整个幼儿期精心照顾它，现在 Python 的发展方向还完全由他控制着。对于这位 Python 之父，我们实在没有什么可说的了。

第 14 章，引言由 Andy McKay 撰写。

本章主要介绍了 Web 相关的技术，包括 CGI 脚本、在 Jython 中运行 Java servlet 以及访问 web 页面的内容。

Andy McKay 是 Enfold Systems 的副总裁和创始人之一。过去几年来，Andy 从快乐的 Perl 用户转变成了狂热的 Python、Zope 以及 Plone 专家。他写了 *Definitive Guide to Plone* 一书，并且还运营着一个热门的 Zope 讨论网站，<http://www.zopezen.org>。

第 15 章，引言由 Jeremy Hylton 撰写。

本章介绍了将 Python 用于简单的分布式系统的一些技术，包括了 XML-RPC、CORBA 以及 Twisted 的 Perspective Broker。

Jeremy Hylton 在 Google 工作。除了他年幼的双胞胎孩子，他还喜欢计算机编程理论、解析器等。作为他为 CNRI 工作的一部分，Jeremy 在许多分布式系统上做过工作。

第 16 章，引言由 Paul F. Dubois 撰写。

本章涉及了程序内省、currying 机制、动态导入以及程序发布、词法分析和解析。

Paul Dubois 在 Lawrence Livermore 国家实验室工作过多年，从天气模型到原子模拟器，他为那里的科学家开发过各种各样的软件系统。他在有关科学计算方面有着无与伦比的经验，同时还精于程序设计和高级的面向对象编程技术。

第 17 章，引言由 David Beazley 撰写。

本章提供了一些技巧和方法，可以帮助读者在 Python 扩展的开发中事半功倍。

David Beazley 最主要的名声来自于 SWIG，这是一个精巧强大的工具，可以快速地将 C 程序和相关的库打包，再由 Python、Tcl、Perl 等其他语言来调用。这个看似语言无关的中立的工具实际上是以支持 Python 为第一要务的，关于此点，在他的著作 *Python Essential Reference* 中可见端倪。David Beazley 是相当变态（这里没有贬义）的家伙，他总是能让我们相信他的脑子里随时会产生出更多惊人而强大的工具。他现在正在芝加哥大学向计算机科学系的学生灌输他的幽默感。

第 18 章，引言由 Tim Peters 撰写。

本章提供了一些用 Python 实现的、优美而实用的算法和数据结构。

Tim Peters 的相关信息见前面第 5 章内容介绍部分。

第 19 章，引言由 Raymond Hettinger 撰写。

本章（第 2 版中新加入的章节）介绍和展示了迭代器和生成器的强大威力，它们让你的循环结构变得更简单、更快速，同时也更具有复用性。

Raymond Hettinger 是 `itertools` 包的开发者，也是生成器表达式方案的提出者。他已经成为了 Python 发展中重要的贡献者。如果你不知道是谁构思和实现了 Python 2.3 和 2.4 中的一些新颖而重要的优化，我们建议你打赌 Raymond 就是那个家伙。

第 20 章，引言由 Raymond Hettinger 撰写。

本章（第 2 版中新加入的章节）从深层次审视了 Python 语言的一些基础构架和元素，正是这些部分使得 Python 的面向对象编程技术更加强大而平滑。无论是为了找乐子还是应用于实际，你都可以自行发掘和定制这些元素。从创建属性的惯用法到重命名和缓存属性，到装饰器（decorator），再到一个自定义元类工厂（a factory of custom metaclasses），这些方法允许你操纵字节码从而优化程序功能，还允许你避免元类型冲突。本章表明，即使这些部分看起来很复杂、很吓人，只要你静下心来认真研究，你的 Python 应用能力会获得极大的提高。

Raymond Hettinger 的信息见前面第 19 章内容介绍部分。

参考资料

从一般的介绍性文章到非常正式的语言描述文档，有很多文本内容可以帮助你学习 Python 或者帮助你加深对某些技术的理解。

对于一般学习而言，我们推荐下列书目（所有的这些书都至少涵盖了 Python 2.2，除非特别注明）：

- *Python Programming for the Absolute Beginner*, Michael Dawson 著 (Thomson Course Technology)，对于从未编写过程序的人而言，这是一本非常实用的、简洁易读的 Python 介绍性读物。
- *Learning Python*, Mark Lutz, David Ascher 著 (O'Reilly)，这是一本非常全面的关于 Python 基础知识的介绍性读物。
- *Practical Python*, Magnus Lie Hetland 著 (APress)，此书介绍了 Python，同时也在书中开发了很多实用而重要的程序，细节描述多，完成度也高。
- *Dive into Python*, Mark Pilgrim 著 (APress)，这是一本针对有经验的程序员的快速入门读物，也可以免费在线阅读或者下载 (<http://diveintopython.org>)。
- *Python Standard Library*, Fredrik Lundh 著 (O'Reilly)，此书为 Python 发行版的庞大的标准库中的每个模块提供了一个有用的实例（目前此书的第 1 版只支持到 Python 2.0）。

- *Programming Python*, Mark Lutz 著 (O'Reilly), 这是一本有关 Python 编程技术的全面的纲要性读物 (目前此书第 2 版只支持到 Python 2.0)。
- *Python Essential Reference*, David Beazley 著 (New Riders), 这是一本针对 Python 语言本身和核心的 Python 库的快速查阅参考书 (目前此书第 2 版只支持到 Python 2.1)。
- *Python in a Nutshell*, Alex Martelli 著 (O'Reilly), 这是一本广为使用的、可以快速查阅 Python 语言和一些关键库的参考书。

另外, 其他一些具有特殊用途的书也可以帮助你了解 Python 编程在特定领域的应用。至于是否喜欢那些书, 则要看你感兴趣的领域是什么。根据个人经验, 编辑推荐下列书目:

- *Python and XML*, Christopher A. Jones, Fred L. Drake, Jr. 著 (O'Reilly), 介绍了关于 Python 读取、处理和转换 XML 的各种技术。
- *Jython Essential*, Samuele Pedroni, Noel Rappin 著 (O'Reilly), 这是一本关于 Jython 的权威书籍, 也是 Python 到 JVM 的一个桥梁。如果读者已经懂一些 Java, 此书的用处就更大。
- *Game Programming with Python*, Sean Riley 著 (Charles River Media), 此书涵盖了用 Python 进行游戏编程的方方面面, 包括从高级图形操作到不错的人工智能。
- *Python Web Programming*, Steve Holden 著 (New Riders), 涵盖了用 Python 构建网络系统的技术, 并介绍了其他一些相关内容 (数据库、HTTP、HTML 等)。对于在这些领域没有经验或者经验很少的读者来说, 这是一本很实用的书。

如果你在阅读本书时遇到问题, 除了上面这些书, 还有一些重要的资源可以帮助你找到答案。我们将在每个小节的“更多资料”的条目下给出一些信息和指引。通常, 我们可以参考和查阅标准的 Python 文档: 最常用的是“Library Reference”, 有时查看“Reference Manual”, 偶尔也会看看“Tutorial”。这些文档都可以以多种形式免费获得。

- python.org 网站 (<http://www.python.org/doc/>), 总是提供最新的关于 Python 的文档。
- pydoc.org 网站 (<http://pydoc.org/>), 提供标准库中每个模块的文档, 这些文档由 pydoc 这个强大工具自动生成。
- Python 本身。最近的 Python 版本提供了一个漂亮的在线帮助系统, 如果没用过可以去试试看。只要在交互的 Python 解释器提示符下键入 help(), 就可以启动帮助系统。
- 你的 Python 安装版本中所含的部分在线文档。ActivePython 的安装程序, 包括了一个可以搜索的 Windows 帮助文件。虽然当前标准的 Python 发布包只是提供