

Foundations of Python Network Programming, Third Edition

Python网络编程

(第3版)

从应用开发角度介绍网络编程基本概念、模块以及第三方库

利用Python轻松快速打造网络应用程序

Python 3示例讲解

[美] Brandon Rhodes John Goerzen 著
诸豪文 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

Foundations of Python Network Programming, Third Edition

Python网络编程

(第3版)



[美] Brandon Rhodes John Goerzen 著
诸豪文 译

人民邮电出版社
北 京

图书在版编目(CIP)数据

Python网络编程：第3版 / (美) 布兰登·罗德
(Brandon Rhodes), (美) 约翰·格岑 (John Goerzen)
著；诸豪文译. -- 北京：人民邮电出版社，2016.9
(图灵程序设计丛书)
ISBN 978-7-115-43350-3

I. ①P… II. ①布… ②约… ③诸… 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(2016)第191199号

内 容 提 要

本书针对想要深入理解使用 Python 来解决网络相关问题或是构建网络应用程序的技术人员，结合实例讲解了网络协议、网络数据及错误、电子邮件、服务器架构和 HTTP 及 Web 应用程序等经典话题。具体内容包括：全面介绍 Python 3 中最新提供的 SSL 支持，异步 I/O 循环的编写，用 Flask 框架在 Python 代码中配置 URL，跨站脚本以及跨站请求伪造攻击网站的原理及保护方法，等等。

本书适合 Web 应用程序的开发者、系统集成者、系统管理员以及所有 Python 程序员。

-
- ◆ 著 [美] Brandon Rhodes John Goerzen
译 诸豪文
责任编辑 朱 巍
执行编辑 温 雪
责任印制 彭志环
- ◆ 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>
北京昌平百善印刷厂印刷
- ◆ 开本：800×1000 1/16
印张：22.5
字数：532千字 2016年9月第1版
印数：1-4 000册 2016年9月北京第1次印刷
- 著作权合同登记号 图字：01-2015-1056号
-

定价：79.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广字第 8052 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

版 权 声 明

Original English language edition, entitled *Foundations of Python Network Programming*, 3E by Brandon Rhodes, John Goerzen, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2014 by Brandon Rhodes. Simplified Chinese-language edition copyright © 2016 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

献词

献给我可爱的外甥女们：Avery、Savannah和Aila。

记得我们一起骑着自行车飞奔，她们总是在拐弯处勇猛地冲下山坡。

我希望无论她们将来会不会做网络编程的工作，都能永远保持这种对待生活的无畏。

引 言

经过20年来的严谨创新, Python在引入了诸如上下文管理器(context manager)、生成器(generator)以及推导式(comprehension)等特性的同时, 仍然保持了其语法及概念一贯的简洁。Python终于开始大放异彩, 这对于Python社区来说, 分外激动人心。

在有些人眼中, Python是一门只能被Google和NASA这种一流编程机构冒险使用的精品语言, 但事实恰好相反, Python正在被广泛使用。它不仅应用于传统的编程任务, 如Web应用程序设计; 也被大量“被赶鸭子上架的程序员”——科学家、数据专员以及工程师所采用, 他们编程并非出于兴趣, 而是必须靠编程才能在自己的领域中更进一步。我认为, 一门简单的编程语言为业余编程人员提供的便利是不容小觑的。

Python 3

自2008年问世以来, Python 3一直在不断修订和精简, 以期承担起Python 2的角色。如今, Python 3迎来了面世以来的第二个5年, 已然成为了Python社区内进行创新的首选平台。

Python 3提供给网络程序员的编程平台几乎在方方面面都有所改进, 无论是基础性的(如将Unicode文本设为Python 3的默认字符串类型), 还是特有的(如对SSL的正确支持、内置的用于异步编程的asyncio框架, 以及对标准库中大大小小的模块的细微调整)。这是一个显著的进步。要知道, Python 2就已经是程序员在现代互联网环境中用来快速高效工作的最佳语言之一了。

本书的目的并非提供从Python 2迁移到Python 3的全面指南。本书不会讲述如何给老版本的print语句添加括号, 也不会介绍如何对导入的标准库模块进行重命名, 更不会讲解如何对Python 2中从字节字符串到Unicode字符串的自动转换(这一转换通常基于粗略的猜测) 这一危险特性引入的缺陷代码进行深度调试。关于如何从Python 2迁移到Python 3, 以及如何仔细编写能够同时支持这两个平台的代码, 已经有很多优质的资源提供了相应的指导。

本书的重点在于网络编程, 并且在所有示例脚本及代码片段中都使用Python 3来阐释。这些例子的目的是帮助读者全面了解使用这门语言提供的工具构建网络客户端、网络服务器以及网络工具的最佳实践。读者可以将这些例子与第2版中各章使用的脚本进行比较, 以此来学习从Python 2到Python 3的迁移。两个版本的代码都可以从<https://github.com/brandon-rhodes/fopnp/tree/m/>获取。这要感谢Apress出版社, 让我们能够从网络上获取源代码。接下来各个章节的目的就是介绍如何最大化地使用Python 3提供的功能来解决现代网络编程的问题。

本书直接把关注点放在了如何使用Python 3正确完成工作上, 希望以此帮助准备从头开始编写新应用程序的程序员和准备将老版本代码移植到新规范的程序员。他们都应该了解如何使用Python 3编

写出正确的网络程序代码，并清楚他们应该努力去编写的目标代码的样式和特点。

这一版的改进

这一版对以前的版本进行了更新，除了将目标语言向Python 3迁移，以及对过去5年中标准库和第三方Python模块进行了许多更新外，还在如下方面进行了改进。

- ❑ 这一版列出的每个Python程序都编写成了一个模块。换言之，每个程序都会导入其依赖的模块，定义其函数或类，然后通过一个if语句来确保所有导入行为。只有在模块__name__为特殊字符串值 '__main__' 时，该if语句对应的代码块才会执行。模块__name__为 '__main__'，表示该模块作为主程序执行。这是在本书之前版本中被彻底忽略的一个Python最佳实践，使得读者无法很方便地将示例代码应用到真实的代码库中解决问题。老版本的代码清单在左边空白区域列出了执行逻辑，而没有将其放在if语句中。这样做可能会少写一两行代码，但是初学Python的程序员在部署实际代码时，能从中得到的实践指导却少了很多。
- ❑ 老版本中的脚本临时使用原始的sys.argv字符串列表来解析命令行选项和参数，而这一版中的大多数脚本使用的则是标准库的argparse模块。这不仅阐明并记录了每个脚本被调用时表示的语义，还允许每个脚本的用户使用-h或者--help查询选项，在Windows或Unix的命令行中获取交互式的帮助文档。
- ❑ 这一版的程序通过在with控制语句中打开文件来进行合理的资源控制。with语句包含的代码块完成的时候，打开的文件会自动关闭。在之前的版本中，大多数程序并不会这样做，而是依赖于C Python的运行时服务。Python的官方网站提供的C Python通过严格的引用计数机制，通常可以确保打开的文件能够及时关闭。
- ❑ 大多数程序在进行字符串插值时已经转而使用现代的format()方法，以前则使用string % tuple的方法。后者在20世纪90年代有一定的意义，因为那时大多数程序员都通晓C语言。但对于现在进入这个领域的新人程序员来说，这种方法可读性较差，而且由于自定义的Python类不能对百分号格式符进行操作符重载，因此提供的功能也不够强大。
- ❑ 重写了关于HTTP和万维网的3章（第9章至第11章），侧重于更清晰地解释协议，并介绍Python所提供的大部分用于编写Web应用的现代工具。这一版在解释HTTP协议时使用Requests库进行客户端操作，它提供的API相当实用。第11章提供了Flask和Django框架的示例。
- ❑ Python 3大量改进了为编写安全的应用程序所提供的支持，所以这一版彻底重写了关于SSL/TLS（第6章）的内容。Python 2的ssl模块使用的是一个折中的方法。该方法功能较弱，甚至没有验证服务器的证书是否与Python连接的主机名对应。因此，Python 3的ssl模块提供了一个设计更严谨、功能更丰富的API，以便用户安全方便地使用其特性。

因此，对于渴望不断学习的程序员来说，且不论Python 3本身所做出的改进，单论代码清单以及示例的构建，这一版相较以前的版本也绝对是更好的资源。

网络实验环境

本书给出程序的源代码可在网上获取，因此本书所有者及想要阅读本书的读者均可访问网络资源

进行学习。读者可以从如下网址找到本书各章的代码目录：

<https://github.com/brandon-rhodes/fopnp/tree/m/py3>

然而对于对网络编程充满好奇的学生来说，本书给出的代码清单所能提供的帮助仍然是有限的。如果只在单机上进行实验，那么网络编程的很多特性都无法得到体现。因此，本书的程序库也提供了一个由12台机器组成的网络实验环境，每台机器通过一个Docker容器实现。程序库同样包含了一个安装脚本，用来构建、启动以及连接各容器的镜像。读者可从如下地址找到该安装脚本以及容器镜像：

<https://github.com/brandon-rhodes/fopnp/tree/m/playground>

图0-1展示了这12台机器以及它们的连接架构。该网络的设计目的是模拟一个小型的互联网。

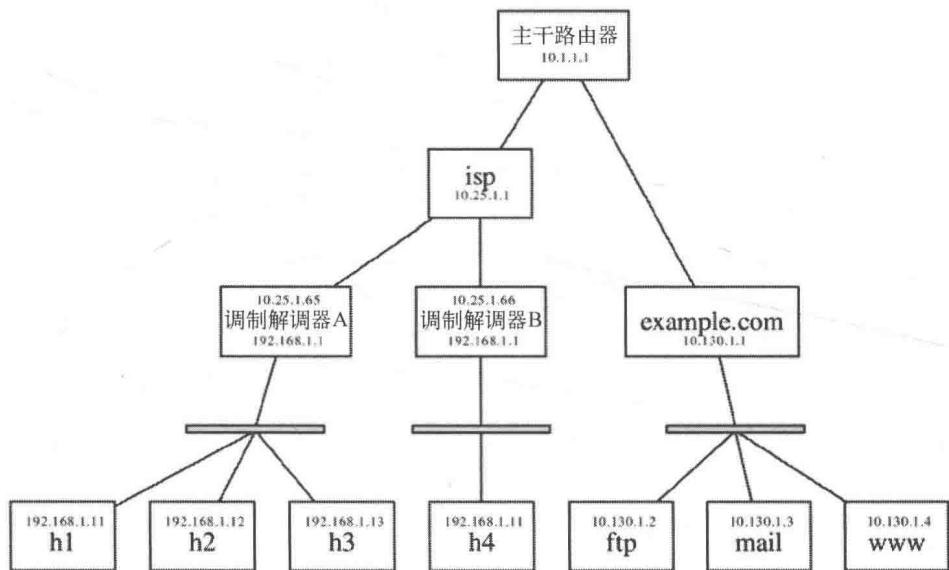


图0-1 网络实验环境的拓扑结构

- 调制解调器A和B下面的客户机（h1到h4）表示典型的客户端场景，如家庭或咖啡店。这些客户机不向互联网提供服务。事实上，它们对外部网络是完全不可见的。它们与同一家庭或咖啡店环境内的其他主机共享子网，只拥有在这些子网内才有意义的本地IP地址。当这些客户机连接外网时，其实都是通过调制解调器的IP地址进行连接的。
- 调制解调器可通过一个ISP网关与更广域的网络直接相连。实验环境中的广域网由一个主干路由器表示，该路由器负责将数据包发送至与之相连的网络。
- example.com及与之相连接的机器表示一组简单的面向服务的机房配置。这里没有进行任何网络地址转换或伪装。互联网上的各个客户端可随意访问example.com后的3个服务器提供的服务端口。
- ftp服务器、mail服务器和www服务器中的任意一个都运行着正确配置的守护进程。因此，本书中的Python脚本也可以运行在其他机器上，并成功连接到上述服务。
- 所有服务器均已成功安装TLS证书（见第6章），而所有客户机则有example.com的签名认证，并安装了受信证书。这意味着要求TLS认证的Python脚本可以成功获取认证。

随着Python和Docker的不断优化，我们也会持续维护网络实验环境。关于如何下载并在本机上运行该网络实验环境的使用说明会不断更新，也会根据用户的使用报告进行微调，以保证读者可以在Linux、Mac OS X以及Windows机器上运行该实验环境的虚拟机。

由于可以在网络实验环境的任意一台机器上连接并运行命令，读者可对网络中的任意一个点进行数据包追踪，查看客户端和服务端之间的网络数据传输情况。通过文档中的示例代码以及本书中的例子和使用说明，读者可以深刻、生动地理解客户端与服务端是如何通过网络进行通信的。

致 谢

2009年，当我开始修订本书第1版，重新编写第2版时，我就对自2004年出版第1版以来的5年之内Python语言发生的巨大变化震惊不已。CherryPy和Django这样全新的Web框架彻底改变了程序员编写Web应用的方式，而mechanize和lxml这样的库也大大简化了从HTTP服务器获取信息的步骤。

当我完成第3版的初稿时，Python社区又完成了一次革新！大大小小的改进让我觉得有必要对一些文字进行修改，并且彻底重写某些章节。这个新版首要体现了整个Python语言社区的辛勤工作，无论是编写Python 3的语言核心开发者，还是Flask、Tornado等让编写HTTP服务更方便更安全的新Web框架的作者，又或者是维护标准库和第三方库的程序员；说真的，他们对整个生态系统进行了优化，加入了大量工具和库，帮助我们更好地编写互联网程序。

自第2版出版以来，许多读者联系了我，提出了各种问题和想法，或是指出某个代码清单无法成功运行或是不再正确。我希望他们知道，这些反馈对于新版本的编写是至关重要的，我想他们将在本书中看到基于这些反馈所做出的重大改进。

非常感谢Apress的编辑和审稿人，他们阅读了我的初稿（有些地方写得特别挫），发现代码清单中的bug以及错误，在编写到完成的过程中给予了帮助及指导。我要特别感谢文字编辑，我慢慢地从他们那里学到不要在每句话里都说very和actually。也要感谢他们，让我在人生中慢慢学会如何区分在连接从句时该使用that还是which。

最后，感谢在我编写本书时所有等着我回邮件、改bug或是和我见面的人，特别感谢我的妻子Jackie一直给我鼓励。

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

目 录

第 1 章 客户端/服务器网络编程简介	1	3.2 何时使用 TCP	38
1.1 基础：协议栈与库	1	3.3 TCP 套接字的含义	39
1.2 应用层	4	3.4 一个简单的 TCP 客户端和服务端	40
1.3 协议的使用	5	3.4.1 每个会话使用一个套接字	43
1.4 一个原始的网络会话	6	3.4.2 地址已被占用	44
1.5 层层深入	8	3.5 绑定接口	46
1.6 编码与解码	9	3.6 死锁	46
1.7 网际协议	10	3.7 已关闭连接，半开连接	51
1.8 IP 地址	11	3.8 像使用文件一样使用 TCP 流	52
1.9 路由	12	3.9 小结	53
1.10 数据包分组	13	第 4 章 套接字名与 DNS	54
1.11 进一步学习 IP	14	4.1 主机名与套接字	54
1.12 小结	15	4.1.1 套接字的 5 个坐标	55
第 2 章 UDP	16	4.1.2 IPv6	56
2.1 端口号	17	4.2 现代地址解析	57
2.2 套接字	18	4.2.1 使用 getaddrinfo() 为服务器 绑定端口	58
2.2.1 混杂客户端与垃圾回复	22	4.2.2 使用 getaddrinfo() 连接服务	59
2.2.2 不可靠性、退避、阻塞和超时	23	4.2.3 使用 getaddrinfo() 请求规范 主机名	60
2.2.3 连接 UDP 套接字	27	4.2.4 其他 getaddrinfo() 标记	61
2.2.4 请求 ID：好主意	28	4.2.5 原始的名称服务程序	62
2.3 绑定接口	29	4.2.6 在代码中使用 getsockaddr()	62
2.4 UDP 分组	31	4.3 DNS 协议	64
2.5 套接字选项	33	4.3.1 为何不使用原始 DNS	66
2.6 广播	33	4.3.2 使用 Python 进行 DNS 查询	66
2.7 小结	35	4.3.3 解析邮箱域名	68
第 3 章 TCP	37	4.4 小结	70
3.1 TCP 工作原理	37		

第 5 章 网络数据与网络错误	71	8.2 散列与分区	137
5.1 字节与字符串	71	8.3 消息队列	140
5.1.1 字符串	72	8.4 小结	145
5.1.2 二进制数与网络字节顺序	75	第 9 章 HTTP 客户端	147
5.2 封帧与引用	77	9.1 Python 客户端库	147
5.3 pickle 与自定义定界符的格式	82	9.2 端口、加密与封帧	149
5.4 XML 与 JSON	83	9.3 方法	151
5.5 压缩	84	9.4 路径与主机	152
5.6 网络异常	85	9.5 状态码	152
5.6.1 抛出更具体的异常	87	9.6 缓存与验证	155
5.6.2 捕捉与报告网络异常	87	9.7 传输编码	157
5.7 小结	88	9.8 内容协商	158
第 6 章 TLS/SSL	90	9.9 内容类型	160
6.1 TLS 无法保护的信息	90	9.10 HTTP 认证	160
6.2 可能出问题的地方	91	9.11 cookie	162
6.3 生成证书	93	9.12 连接、Keep-Alive 和 httplib	163
6.4 TLS 负载移除	95	9.13 小结	164
6.5 Python 3.4 默认上下文	96	第 10 章 HTTP 服务器	166
6.6 手动选择加密算法与完美前向安全	102	10.1 WSGI	166
6.7 支持 TLS 的协议	104	10.2 异步服务器与框架	168
6.8 了解细节	105	10.3 前向代理与反向代理	169
6.9 小结	111	10.4 4 种架构	170
第 7 章 服务器架构	112	10.4.1 在 Apache 下运行 Python	171
7.1 浅谈部署	112	10.4.2 纯粹的 Python HTTP 服务器的兴起	172
7.2 一个简单的协议	114	10.4.3 反向代理的优势	172
7.3 单线程服务器	117	10.5 平台即服务	173
7.4 多线程与多进程服务器	120	10.6 GET 与 POST 模式和 REST 的问题	174
7.5 异步服务器	122	10.7 不使用 Web 框架编写 WSGI 可调用 对象	176
7.5.1 回调风格的 asyncio	126	10.8 小结	180
7.5.2 协程风格的 asyncio	127	第 11 章 万维网	181
7.5.3 遗留模块 asyncore	129	11.1 超媒体与 URL	181
7.5.4 两全其美的方法	130	11.1.1 解析与构造 URL	182
7.6 在 inetd 下运行	131	11.1.2 相对 URL	184
7.7 小结	133	11.2 超文本标记语言	186
第 8 章 缓存与消息队列	134		
8.1 使用 Memcached	134		

11.3 读写数据库	189	13.2.2 邮件头与信封接收者	245
11.4 一个糟糕的 Web 应用程序 (使用 Flask)	190	13.2.3 多跳	246
11.5 表单和 HTTP 方法	195	13.3 SMTP 库简介	247
11.5.1 表单使用了错误方法的 情况	197	13.4 错误处理与会话调试	248
11.5.2 安全的 cookie 与不安全 的 cookie	198	13.5 从 EHLO 获取信息	251
11.5.3 非持久型跨站脚本	200	13.6 使用安全套接层和传输层安全 协议	253
11.5.4 持久型跨站脚本	201	13.7 认证的 SMTP	255
11.5.5 跨站请求伪造	202	13.8 关于 SMTP 的小贴士	257
11.5.6 改进的应用程序	203	13.9 小结	257
11.6 使用 Django 编写的账单应用程序	205	第 14 章 POP	258
11.7 选择 Web 框架	209	14.1 POP 服务器的兼容性	258
11.8 WebSocket	210	14.2 连接与认证	259
11.9 网络抓取	211	14.3 获取邮箱信息	261
11.9.1 获取页面	212	14.4 消息的下载与删除	263
11.9.2 抓取页面	215	14.5 小结	265
11.9.3 递归抓取	217	第 15 章 IMAP	266
11.10 小结	221	15.1 在 Python 中使用 IMAP	267
第 12 章 电子邮件的构造与解析	222	15.1.1 IMAPClient	269
12.1 电子邮件消息格式	222	15.1.2 查看文件夹	271
12.2 构造电子邮件消息	224	15.1.3 消息号与 UID	272
12.3 添加 HTML 与多媒体	226	15.1.4 消息范围	272
12.4 添加内容	231	15.1.5 摘要信息	272
12.5 解析电子邮件消息	232	15.1.6 下载整个邮箱	274
12.6 遍历 MIME 部件	234	15.1.7 单独下载消息	276
12.7 邮件头编码	236	15.1.8 标记并删除消息	281
12.8 解析日期	237	15.1.9 删除消息	282
12.9 小结	238	15.1.10 搜索	282
第 13 章 SMTP	239	15.1.11 操作文件夹与消息	284
13.1 电子邮件客户端与 Web 邮件服务	239	15.1.12 异步性	285
13.1.1 最开始使用命令行发送 电子邮件	239	15.2 小结	285
13.1.2 客户端的兴起	240	第 16 章 Telnet 和 SSH	286
13.1.3 转移到 Web 邮件	241	16.1 命令行自动化	286
13.2 SMTP 的使用方法	243	16.1.1 命令行扩展与引用	287
13.2.1 发送电子邮件	244	16.1.2 UNIX 命令行参数几乎可以 包含任意字符	288
		16.1.3 对字符进行引用	290