

TURING

图灵程序设计丛书

Python

第2版

项目开发实战

【日】日本BePROUD股份有限公司 著 支鹏浩 译

会写代码≠能做好项目!



- ✓ 建立有序生产环境
- ✓ 迅速融入开发团队
- ✓ 高效处理项目问题

网罗Python项目开发中的流程

让你的编程事半功倍

Python项目与封装/团队开发环境/问题驱动开发/源码管理（Mercurial）
Jenkins持续集成（CI）/环境搭建与部署的自动化（Ansible）/Django框架……



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

Python

第2版

项目开发实战

【日】日本BePROUD股份有限公司 著 支鹏浩 译

人民邮电出版社
北京

图书在版编目(CIP)数据

Python 项目开发实战 / 日本 BePROUD 股份有限公司著;
支鹏浩译. -- 2 版. -- 北京: 人民邮电出版社,
2017.1

(图灵程序设计丛书)

ISBN 978-7-115-43856-0

I. ①P… II. ①支… ②支… 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 (2016) 第 254060 号

Python Professional Programming Dai 2 Han

Copyright © BeProud Inc. 2015

All rights reserved.

First original Japanese edition published by SHUWA SYSTEM CO., LTD., Japan.

Chinese (in simplified character only) translation rights arranged with SHUWA SYSTEM CO., LTD., Japan.
through CREEK & RIVER Co., Ltd. and CREEK & RIVER SHANGHAI Co., Ltd.

本书中文简体字版由 SHUWA SYSTEM CO., LTD., Japan. 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。

内 容 提 要

本书来自真正的开发现场, 是 BePROUD 公司众多极客在真实项目中的经验总结和智慧结晶。作者从 Python 的环境搭建开始讲起, 介绍了 Web 应用的开发方法、项目管理及审查、测试与高效部署、服务器调试等内容, 尽可能网罗了 Python 项目开发流程中的方方面面, 有助于开发者建立有序生产环境, 提高开发效率, 让编程事半功倍。此外, 在本书中 Python 仅仅是一个载体, 很多知识点在非 Python 下也适用。

本书适合有一定基础的 Python 开发者, 以及使用 PHP 或 Ruby 进行开发的读者阅读。

- ◆ 著 [日] 日本 BePROUD 股份有限公司
译 支鹏浩
责任编辑 傅志红
执行编辑 高宇涵 侯秀娟
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>
大厂聚鑫印刷有限责任公司印刷

- ◆ 开本: 800×1000 1/16
印张: 27.5

字数: 650 千字 2017 年 1 月第 1 版
印数: 1-4 000 册 2017 年 1 月河北第 1 次印刷

著作权合同登记号 图字: 01-2015-6980 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

引言

迄今为止，BePROUD 公司已使用 Python 开发了诸多项目。我们之所以撰写本书，是为了与各位读者分享我们在实践中总结出的一些技巧。

同时，鉴于最近公司员工数量增长，我们把在 BePROUD 工作所需的知识也写入了本书，以便新的公司成员能尽快熟悉工作。

因此本书从搭建工作环境开始讲起，逐步涉及 Web 应用的开发、项目管理及审查、测试代码的编写与高效部署、服务器调试等方面，网罗了 Python 项目开发工作中的一系列流程。书名中的“实战”一词就包含了“工作”的意思。

书中所写的技巧主要源于我们的 Python2 开发经验。也正因为如此，本书将以 Python2 为例进行讲解。如今新的开发项目已经在使用 Python3，这些技巧转移到 Python3 上理应同样适用。

进入正题之前，先来聊聊我们的日常思路。

● 极客 / 书虫常伴身边的公司

BePROUD 里不乏极客和书虫们。在这里，很多人对特定领域的了解程度能吓掉你的下巴。

在这里，人们一旦发现感兴趣的事，就会拿出私人时间来学习、实践。要知道，极客和书虫们不会为这种事情吝啬时间。

正如人们印象中的那样，极客和书虫们大多有些怪癖，但 BePROUD 的员工都具备下列共识。

- 希望能不做不想做的事
- 希望学会好的方法并付诸实践
- 希望工作时有个好心情

● 希望能不做不想做的事

在工作中，重复单调的作业是一种极其无趣的事，因此能一次办完的事谁都不想去办两次。另外，大家都讨厌工序复杂、容易出错的工作。所以要开动脑筋，把复杂的工序简单化，同时尽量减少出错的机会。

● 希望学会好的方法并付诸实践

世界上有许多公认的好方法、新思路和新技巧，我们要勇于尝试，学习它们并付诸实践。

使用好的方法必然能帮助我们削减不想做的工作。不过，方法的好坏不能人云亦云，我们必须选出对自己真正有帮助的方法，然后再将所学方法应用到实际业务当中。

● 希望工作时有个好心情

现在，我们学会了优秀的方法、削减了繁杂的工作，之后自然希望带着好心情去工作。此时不妨给 Skype 做个好玩的 bot，或者在下班后找个会议室搞一场妙趣横生的快速演讲。我们希望大家能在保质保量完成工作的同时有个好心情，而不是只把公司当作工作的场所。这是我们的理念。

本书的内容全部基于事实，都是 BePROUD 员工实际尝试、实践过的。我们希望给各位提供一些能实际应用且行之有效的知识，而不是让各位去死记硬背一大堆晦涩难懂的概念。我们很愿意看到本书的知识能对各位有所帮助，愿各位能在工作中有个好心情。

● 谢辞

本书在编撰过程中承蒙多名 IT 业界高人指点：寺田学 (@terapyon)、金子望、关根裕纪 (@checkpoint)、畠弥峰 (@flag_boy)、小坂健二郎 (@inoshiro)、筒井隆次 (@ryu22e)、永井孝 (@ngi644)、中西直树 (@nk24)、尾曾越雅文、柴田正明 (@_mshibata)、真幡康德 (@mahata)、中石宜亨 (@ciryplus)。各位在百忙之中仍担起审校工作，慷慨赐教，我们在此表示由衷的感谢。此外还有 BePROUD 公司的 haru、altnight、masaya、crohaco、nakagami、yyyk，感谢几位一边处理着公司内繁忙的开发工作，一边见缝插针地为本书进行审校。

最后感谢各位未能在著者处署名的 BePROUD 员工。如果没有各位员工长期以来的切磋琢磨，这本书永远不会问世。

至此，希望这本集诸人之力编撰出来的书，能为 IT 业界出一份绵薄之力。

全体执笔者

2015 年 1 月

📍 本书网址

<http://www.ituring.com.cn/book/1719>

📍 本书介绍的软件版本和 URL 均为截止到 2015 年 1 月底的最新信息，当前可能已发生变更。

前言

● 本书涉及的内容

本书分为4个部分，共15章。

第1部分“Python 开发入门”的重点将放在个人开发。内容涵盖 Python 开发过程中必不可少的工具的安装（第1章），简单的 Web 应用开发（第2章）以及 Python 项目的结构与包的创建（第3章）。

第2部分“团队开发的周期”将为各位说明多人团队开发的相关问题。这部分将重点介绍团队高效开发过程中不可或缺的技术和技巧，内容涵盖团队开发前的环境调整（第4章）、项目管理与审查（第5章）、源码管理（第6章）、文档（第7章）、模块设计与单元测试（第8章）、封装及其运用（第9章）、持续集成（第10章）等。

第3部分“服务公开”将向各位讲解如何搭建与运用正式环境公开 Web 服务（第11章），此外就是有关性能调节的一些方法（第12章）。

第4部分“加速开发的技巧”可以说是加速开发的一些小贴士。例如将测试的概念导入整个开发流程以加快项目进度（第13章），Django 的基础及其进阶性、实践性的用法（第14章），Python 的辅助模块（第15章）等。

● 阅读本书前的准备

环境及版本

- OS: Ubuntu-14.04
- Python: 2.7.6
- Bash: 4.3
- 从第2章起，如无特别说明，则运行环境皆由 virtualenv 搭建。

关于 OS

实体机使用 Windows/OS X/Linux，服务器的测试环境使用虚拟机上的 Ubuntu。

Python 的官方手册

<https://docs.python.org/2.7/>

我们仅对 Python 官方手册中的内容做最低限度的介绍，部分说明会被省略。因此建议各位

手边时常准备一份参考手册以便阅读。

Python 的官方教程非常适用于学习 Python 的基本安装流程、语法、术语、类以及模块。本书将以各位看过这份教程为前提进行讲解。

Unix/Linux 的一般命令操作

本书虽以 Ubuntu Linux 为前提讲解，但书中不对 Ubuntu Linux 的基本命令操作进行说明。

关于 PyPI (Python Package Index)

PyPI^① 是一个集中管理包的网站，pip 等自动包安装工具会用到它。本书使用的包也来自 PyPI。

关于敏捷过程与极限编程

本书并不对敏捷过程 (Agile Process) 和极限编程 (Extreme Programming) 做单独的说明。如今在许多书籍和网站上都能找到这二者的介绍，感兴趣的读者可以去读一读。

本书面向的人群

- 希望改善个人开发环境的人
- 希望改善团队开发的人
- 想学习工作中可使用的 Python 技巧的人
- 新加入 BePROUD 公司项目的成员

● 注意

- 本书基于作者本人的调查结果而成。
- 我们在加工本书时力求完美。不过若您发现本书存在不足和错误、漏记等问题，请书面联系出版方。
- 对于因本书内容运用不当而导致的结果及其影响，无论是否因上述两项内容引起，我们均不负责，请知悉。
- 未获得出版方书面许可不得全部或部分复制本书。

● 商标等

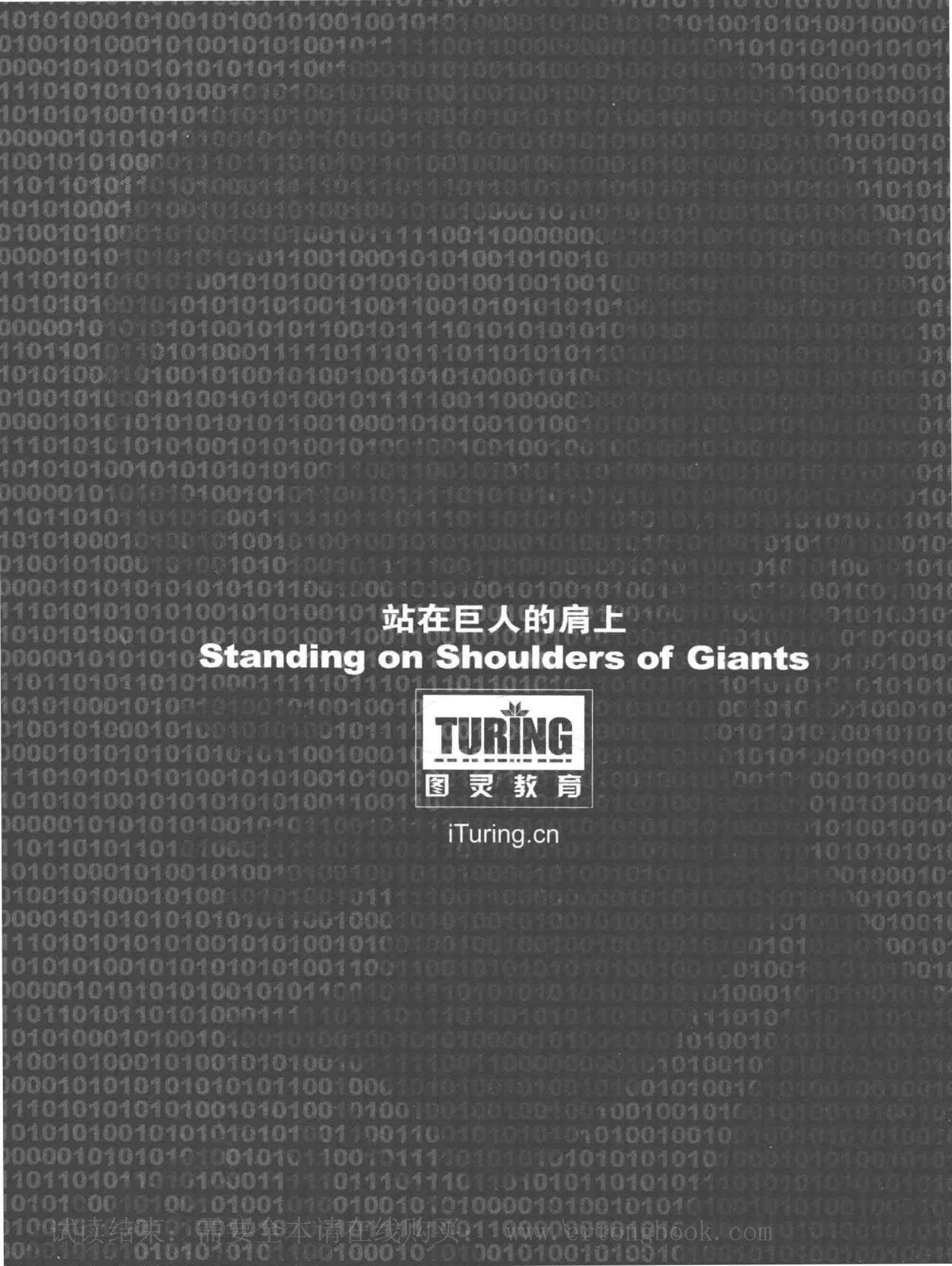
- 本书已省略™ ® © 等符号。
- Python 徽标是 the Python Software Foundation 的商标。
- Django 和 Django 徽标是 Django Software Foundation 的商标。
- Google App Engine 是 Google Inc. 的商标。
- Jenkins 是 SOFTWARE IN PUBLIC INTEREST, INC. 的商标。
- nginx 是 Nginx Software Inc. 的商标。
- VirtualBox 是 ORACLE AMERICA, INC. 的商标。
- Ubuntu 是 Canonical Limited 的商标。
- 此外，公司名和商品名、系统名一般为各开发者的注册商标。
- 本书注册商标中还使用了普遍使用的通用名。

① <https://pypi.python.org/pypi>

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

试读结束！需要全本请在线购买！www.epdfbook.com

目 录

第 1 部分 Python 开发入门	1
第 1 章 Python 入门	2
1.1 安装 Python	2
1.1.1 安装 deb 包	3
1.1.2 安装第三方包	4
1.1.3 virtualenv 的使用方法	5
1.1.4 多版本 Python 的使用	7
1.2 安装 Mercurial	9
1.2.1 Mercurial 概述	10
1.2.2 安装 Mercurial	10
1.2.3 创建版本库	11
1.2.4 文件操作	12
1.3 编辑器与辅助开发工具	14
1.3.1 编辑器	14
1.3.2 开发辅助工具	20
1.4 小结	22
第 2 章 开发 Web 应用	24
2.1 了解 Web 应用	24
2.1.1 Web 应用是什么	24
2.1.2 Web 应用与桌面应用的区别	25
2.1.3 Web 应用的机制	25
2.2 前置准备	28
2.2.1 关于 Flask	28
2.2.2 安装 Flask	28
2.3 Web 应用的开发流程	29
2.4 明确要开发什么应用	30
2.4.1 留言板应用的需求	30
2.4.2 明确必备的功能	30
2.4.3 明确必备的页面	31
2.5 页面设计	31
2.5.1 确定成品页面的形式	31

2.5.2	编写 HTML 和 CSS	32
2.6	实现功能	37
2.6.1	保存留言数据	37
2.6.2	获取已保存的留言列表	38
2.6.3	用模板引擎显示页面	39
2.6.4	准备评论接收方的 URL	42
2.6.5	调整模板的输出	43
2.7	查看运行情况	45
2.8	小结	47
第 3 章	Python 项目的结构与包的创建	48
3.1	Python 项目	48
3.2	环境与工具	49
3.2.1	用 virtualenv 搭建独立环境	49
3.2.2	用 pip 安装程序包	55
3.2.3	小结	62
3.3	文件结构与发布程序包	62
3.3.1	编写 setup.py	62
3.3.2	留言板的项目结构	64
3.3.3	setup.py 与 MANIFEST.in——设置程序包信息与捆绑的文件	65
3.3.4	setup.py——创建执行命令	69
3.3.5	python setup.py sdist——创建源码发布程序包	71
3.3.6	提交至版本库	71
3.3.7	README.rst——开发环境设置流程	73
3.3.8	变更依赖包	75
3.3.9	通过 requirements.txt 固定开发版本	77
3.3.10	python setup.py bdist_wheel——制作用于 wheel 发布的程序包	78
3.3.11	上传到 PyPI 并公开	79
3.3.12	小结	85
3.4	小结	86

第 2 部分 团队开发的周期

87

第 4 章	面向团队开发的工具	88
4.1	问题跟踪系统	88
4.1.1	Redmine	88
4.1.2	安装 Redmine	89
4.1.3	Redmine 的设置	91
4.1.4	插件	93
4.2	版本控制系统	94

4.2.1	Mercurial 与 Redmine 的联动	94
4.2.2	用于生成版本库的插件	95
4.3	聊天系统	97
4.3.1	Slack	97
4.3.2	Slack 的特点	98
4.3.3	Slack 做不到的事	101
4.3.4	Slack 的注册	102
4.4	对团队开发有帮助的工具	102
4.4.1	Dropbox	102
4.4.2	Google Drive	102
4.5	小结	103
第 5 章	项目管理与审查	104
5.1	项目管理与问题的区分使用	104
5.1.1	项目管理的前置准备工作	104
5.1.2	创建问题	105
5.1.3	整理问题	107
5.1.4	分割问题	107
5.2	问题模板	108
5.2.1	安装插件	108
5.2.2	问题模板的使用方法	109
5.2.3	Global Issue Templates	111
5.2.4	问题模板示例	112
5.3	问题驱动开发	114
5.3.1	别急着敲代码, 先建问题	114
5.3.2	创建与问题编号同名的分支	115
5.3.3	让发布与分支相对应	115
5.3.4	分支的合并	116
5.4	审查	117
5.4.1	为什么需要审查	117
5.4.2	审查委托: 代码审查篇	118
5.4.3	审查委托: 作业审查篇	119
5.4.4	实施审查: 代码审查篇	120
5.4.5	实施审查: 作业审查篇	123
5.5	小结	123
第 6 章	用 Mercurial 管理源码	125
6.1	Mercurial 版本库的管理与设置	125
6.1.1	服务器上的 Unix 用户群设置	125
6.1.2	创建版本库	126
6.1.3	hgrc 的设置	127

6.1.4	使用设置好的版本库	127
6.1.5	使用 hgweb 建立简易中央版本库	127
6.2	灵活使用“钩子”	128
6.2.1	钩子功能的设置方法	129
6.2.2	尝试钩子脚本	129
6.2.3	钩子事件	130
6.2.4	钩子功能的执行时机	131
6.2.5	编写钩子脚本	134
6.3	分支的操作	136
6.4	关于合并	137
6.4.1	未发生冲突的合并	138
6.4.2	合并时发生冲突以及用文本编辑器解决冲突的方法	140
6.4.3	合并的类型与冲突	143
6.4.4	用 GUI 的合并工具进行合并	144
6.5	GUI 客户端	147
6.5.1	GUI 客户端的介绍	147
6.5.2	GUI 客户端的优点	149
6.5.3	GUI 客户端的缺点	151
6.6	考虑实际运用的 BePROUD Mercurial Workflow	152
6.6.1	概述	152
6.6.2	背景	152
6.6.3	版本库的结构	153
6.6.4	提交源码	154
6.6.5	提交设计	156
6.6.6	分支的合并	157
6.6.7	集成分支	158
6.7	小结	160
第 7 章	完备文档的基础	162
7.1	要记得给项目写文档	162
7.1.1	写文档时不想做的事	162
7.1.2	什么样的状态让人想写文档	164
7.2	Sphinx 的基础与安装	165
7.2.1	Sphinx 的安装	166
7.2.2	reStructuredText 入门	167
7.2.3	用 Sphinx 写结构化文档的流程	169
7.2.4	Sphinx 扩展	174
7.3	导入 Sphinx 可解决的问题与新出现的问题	175
7.3.1	由于是纯文本，所以能在平时用的编辑器上写文档	176
7.3.2	信息与视图相分离，所以能集中精神编辑内容，不用顾虑装饰等外观问题	176
7.3.3	可根据一个源码输出 PDF 等多种格式	179

7.3.4	通过结构化, 文档可分成几个文件来写	180
7.3.5	能用 Mercurial 等轻松实现版本管理	181
7.3.6	API 参考手册与程序的管理一体化	182
7.3.7	通过 Web 浏览器共享	184
7.3.8	导入 Sphinx 后仍存在的问题	185
7.4	文档集的创建与使用	186
7.4.1	什么是文档集	186
7.4.2	项目所需文档的一览表	187
7.4.3	面向项目组长、经理	187
7.4.4	面向设计者	189
7.4.5	面向开发者	189
7.4.6	面向客户	189
7.5	小结	190
第 8 章	模块分割设计与单元测试	191
8.1	模块分割设计	191
8.1.1	功能设计	191
8.1.2	构成 Web 应用的组件	192
8.1.3	组件设计	194
8.1.4	模块与程序包	195
8.2	测试	197
8.2.1	测试的种类	197
8.2.2	编写单元测试	198
8.2.3	从单元测试中剔除环境依赖	209
8.2.4	用 WebTest 做功能测试	215
8.3	通过测试改良设计	219
8.4	推进测试自动化	221
8.4.1	用 tox 自动生成执行测试的环境	221
8.4.2	可重复使用的测试环境	223
8.5	小结	223
第 9 章	Python 封装及其运用	224
9.1	使用程序包	224
9.1.1	程序包的版本指定	224
9.1.2	从非 PyPI 服务器安装程序包	226
9.1.3	程序包的发布格式	228
9.1.4	生成 wheelhouse 的方法	230
9.1.5	从 wheelhouse 安装	231
9.2	巧用程序包	232
9.2.1	私密发布	232
9.2.2	巧用 requirements.txt	232

9.2.3	requirements.txt 层级化	233
9.2.4	为部署和 CI+tox 准备的 requirements	234
9.2.5	通过 requirements.txt 指定库的版本	235
9.3	小结	236

第 10 章 用 Jenkins 持续集成.....237

10.1	什么是持续集成	237
10.1.1	持续集成的简介	237
10.1.2	Jenkins 简介	239
10.2	Jenkins 的安装	239
10.2.1	安装 Jenkins 主体程序	239
10.2.2	本章将用到的 Jenkins 插件	240
10.3	执行测试代码	241
10.3.1	让 Jenkins 运行简单的测试代码	241
10.3.2	添加 Job	242
10.3.3	Job 的成功与失败	244
10.4	测试结果输出到报告	246
10.4.1	安装 pytest	246
10.4.2	调用 pytest 命令	246
10.4.3	根据 pytest 更改 Jenkins 的设置	246
10.5	显示覆盖率报告	247
10.5.1	安装 pytest-cov	248
10.5.2	从 pytest 获取覆盖率	248
10.5.3	读取覆盖率报告	248
10.6	执行 Django 的测试	250
10.6.1	安装 Python 模块	250
10.6.2	Django 的调整	251
10.6.3	示例代码	251
10.6.4	Jenkins 的调整	255
10.6.5	“构建后操作”选项卡的设置	257
10.7	通过 Jenkins 构建文档	260
10.7.1	安装 Sphinx	261
10.7.2	在 Jenkins 添加 Job	261
10.7.3	Sphinx 构建发出警告时令 Job 失败	261
10.7.4	查看成果	262
10.7.5	通过 Task Scanner Plugin 管理 TODO	263
10.7.6	Task Scanner Plugin 的设置示例	264
10.8	Jenkins 进阶技巧	265
10.8.1	好用的功能	265
10.8.2	进一步改善	267
10.9	小结	268