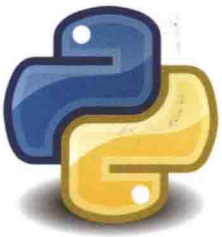
TURING 图灵程序设计丛书

真实项目中的经验总结

行业精英们的智慧结晶



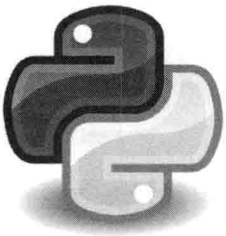
Python 开发实战

团队开发环境的搭建与管理 / ticket驱动开发 / 源代码管理（Mercurial）/ PyPI包
Jenkins持续集成 / 性能优化（nginx与gunicorn）/ GAE开发 / Django框架……

【日】BePROUD股份有限公司 著
盛 荣 译 张靖强 审

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书



Python 开发实战

【日】BePROUD股份有限公司 著
盛 荣 译 张靖强 审

人民邮电出版社
北 京

图书在版编目（CIP）数据

Python开发实战 / 日本BePROUD股份有限公司著 ；盛荣译. -- 北京 ： 人民邮电出版社，2014.6
（图灵程序设计丛书）
ISBN 978-7-115-32089-6

Ⅰ. ①P… Ⅱ. ①日… ②盛… 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(2013)第115494号

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。

作者从 Python 的环境搭建讲起，介绍了 Web 应用开发方法、任务管理，以及评审、测试及高效部署、服务器调试等内容，尽可能网罗了 Python 开发流程中所涉及的方方面面。在这里，Python 仅仅是一个载体，很多知识点在非 Python 下也适用，这也是本书最大的特色所在。

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

◆ 著 [日] BePROUD股份有限公司
译 盛 荣
审 张靖强
责任编辑 乐 馨
执行编辑 金松月
责任印制 焦志炜

◆ 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
北京天宇星印刷厂印刷

◆ 开本：800×1000 1/16
印张：24.5
字数：379千字 2014年 6 月第 1 版
印数：1－4 000册 2014年 6 月北京第 1 次印刷

著作权合同登记号 图字：01-2012-9107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：京崇工商广字第 0021 号

版权声明

前　言

BePROUD 股份有限公司在众多的项目开发中使用了Python编程语言。我们希望通过撰写本书，与你分享在项目实践中所积累Python使用技巧。

最近，公司的员工数量不断增加，为了让新员工快速上手，我们亦将在BePROUD工作必须掌握的基础知识写入本书中。

本书从整个工程环境的建立开始，介绍了Web应用开发方法、任务管理，以及评审、测试及高效部署、服务器调试等内容，尽可能网罗了Python开发流程中所涉及的一系列内容。如书名所示，本书更多关注项目实战技巧。

在进入正题前，我想先谈谈BePROUD公司日常工作的思维方式。

◉Geek/Nerd[①]云集的公司

BePROUD是一个Geek和Nerd众多的公司，云集了许多行业精英。

出于兴趣，这些员工经常利用自己的业余时间学习，将学到的知识付诸实践。作为Geek/Nerd，他们永远不会吝惜时间和精力。

当然，Geek/Nerd中有许多怪人，但BePROUD公司所有成员都有这样的共识：

- 尽量简化不想做的工作。
- 掌握了好的方法就要付诸实践。
- 带着好心情快乐地工作。

◉尽量简化不想做的工作

在工作中，重复地做一项工作是非常无聊的。我们都尽量想办法使其一劳永逸。另外，我们也不喜欢那些过程复杂且又容易失败的工作。因此我们尽可能将其简化，防止失败。

◉掌握了好的方法就要付诸实践

世间总有一些好办法或一些新想法，我们总是积极地在工作中去尝试。

在工作中去实践好方法往往可以减轻我们的负担。但在实践之前，我们必须先甄别出可以真正起到作用的方式方法，然后再开始学习、应用。

◉带着好心情快乐地工作

如前所述，当掌握了好的方法，提高工作效率之后，我们就可以抱有一种好心情去享受工作

① Geek和Nerd都有痴迷某物的意思，本书指的是对于软件开发等领域极度痴迷的人。——译者注

了。比如做一个好玩的Skype机器人，在下班后找一个空闲的会议室玩LT游戏等。我们都认为，公司不仅是工作场所，也是一个让大家在做好工作的同时又能够享受它的地方。

本书的内容基于BePROUD公司员工的实践经验，因此不会强迫你去单纯地记忆复杂的知识，而是为你提供能够在实践中直接应用的内容。如果本书中介绍的知识能够对你的实际工作有帮助，我们将感到莫大的欣慰。

◉谢辞

感谢以下IT界同行审阅本书：寺田学（@terapyon）、铃木孝典（@takanory）、田村佑树（@isoparametric）、川口耕介（@kohsukekawa）、和田贵久（@wadatka）、矢野勉（@t_yano）、涉川芳树（@shibukawa）、西尾泰和（@nishio）、饭野卓见（@troter）、关根佳直、四柳刚（@ytyng）、波田野欲一（@tcsh）。大家在百忙之中，非常爽快地接受了审稿工作并且给出了中肯的意见，再次深表感谢。还有BePROUD公司的小田切笃（@aodag）、菊田刚（@nyomui）、高桥和之（@kazyk）等，他们在繁忙的项目中抽出时间来审阅书稿，同样非常感谢他们。

最后，还要感谢那些为本书执笔却没有署名的BePROUD的各位员工。没有大家平时的斟酌切磋，本书也是无法完成的。

正是得到大家的鼎力相助，本书才得以完成。让我们一起祝愿本书能够为IT业界奉献微薄之力。

本书所有执笔者

导　读

◉本书涵盖的内容

本书分成四部分，共15章。

第一部分以个人开发为重点，介绍了如何在个人开发中构建Python开发环境（第1章），并说明了开发简单的Python应用程序的方法（第2章）。

第二部分介绍了团队开发的相关内容，依次探讨了如何准备团队开发环境（第3章）、文档（第4章）、项目管理与评审（第5章）、模块设计与单元测试（第6章）、库的发布与打包（第7章）、版本控制（第8章）、持续集成（第9章）等。这些主题都是在提高团队开发效率方面必不可少的技术与方法。

第三部分介绍了Web服务发布的环境构建与运用所必须的技术（第10章）和性能调优的方法（第11章），并详细探讨了使用Google App Engine进行Web应用构建的方法（第12章）。

第四部分先阐述了将测试引入整个开发流程从而提高项目效率这一话题（第13章），随后从Django的基础开始讲起，介绍它在实践中的使用方法（第14章）。最后讲解了提高开发速度的诀窍，如Python中实用的开发模块（第15章）。

◉阅读本书时需要准备

软件环境以及版本

- 操作系统：Ubuntu-11.10
- Python：2.7.2

关于操作系统

主机可以使用Windows/Mac OS/Linux等作为服务器的测试环境，运行虚拟机，在虚拟机上运行Ubuntu。

Python的官方网站帮助链接

http://www.python.jp/doc/2.7/[①]

Python官方网站的参考文档一般只介绍最基本的使用方法，有时会省略部分说明，因此建议你常备更详细的参考手册慢慢阅读。

Python官方网站的学习教程非常适合用来学习Python的安装步骤、基本语法、类以及模块。该教程中所涉及的知识，本书将不再赘述。

---

① 可以参阅http://docs.python.org/release/2.7/ 。——译者注

Unix/Linux的一般操作命令

本书采用Ubuntu Linux作为操作系统进行阐述，因此有关Unix/Linux的基本操作均不再介绍。你如果想知道相关的操作命令，可以参考http://linuxjm.sourceforge.jp/。

关于PyPI（Python Package Index）

PyPI（http://pypi.python.org/pypi）提供了众多使用pip工具进行安装的Python程序包，并且该工具还提供了程序包自动安装的相关功能[①]。本书用到的程序包均可在PyPI站点中找到。

关于敏捷开发和极限编程

本书不会针对敏捷开发[②]和极限编程[③]作个别说明。关于这两个主题有众多的书籍和网站可以供参考。

◉本书读者对象

本书适合如下读者：

- 希望改善个人开发环境的开发者
- 希望使团队开发更上一层楼的开发者
- 在工作中使用Python环境并想学习相关技术的人
- 新加入BePROUD公司项目的人

① 非常类似Perl的CPAN。——译者注
② 即agile。——译者注
③ 即eXtreme Programming 。——译者注

目　录

第一部分　使用 Python 开发

第二部分　团队开发的周期和流程

第三部分 服务的公开

第四部分　加速开发的技术

Part 1 第一部分

使用 Python 开发

各位读者，当你在学习一门新技术或者新的编程语言时，是不是常有无从下手的感觉？初学者在学习前，往往不知道事先应该准备什么；或刚开始准备，就马上在安装问题上栽了跟头；抑或是准备工作完成后，打算开始学习时，才发现事先应该安装的软件没有安装到位等。上述各种情况最终导致初学者在学习之初，就花费了大量的时间。

在构建 Python 的开发环境时，往往会遇到不同操作系统版本和 Python 版本，我们需要有一个兼容平滑的开发环境。此时，个人开发者尤其是自学者几乎都是参考书本或者 Web 站点等各种信息，并根据自己的需要建立开发环境。虽然这些开发环境并不完全相同，但有许多共同点。

在本书第 1 章中，我们将个人开发者建立开发环境的步骤尽可能深入浅出地展现出来。即使是初学者，也可按照里面的步骤进行操作。随后在第 2 章中，我们将会尝试实际开发一个 Web 应用。

本部分内容

01 开始使用Python

本章我们将学习以下内容。我们的说明会在虚拟机环境上进行，将用来学习的主机称为“Host系统”，并将虚拟机上安装的系统称为“Guest系统”。如果你自己已经准备好了Python的开发环境，跳过本章也没有关系。

- Python 环境搭建
- Mercurial 环境搭建
- 编辑器与方便的开发工具

01.01 Python环境搭建

本节介绍在Ubuntu下开发Python时，需要安装的一些实用的工具包。

NOTE

本书默认的操作系统为Ubuntu 11.10 （Server版）。该操作系统运行在Oracle VirtualBox虚拟机上。如果你还没有掌握这类环境的搭建方法，可以参考本书的附录A。初学者也请先参照附录，再阅读下面的内容。

01.01.01 安装apt软件包

Ubuntu使用aptitude命令进行操作系统的软件包管理[①]。首先我们使用下面的命令更新所有的安装包，这些安装包中包括了Python开发所必备的工具[②]。

apt软件包的更新与升级

```
$ sudo aptitude -y update
$ sudo aptitude -y upgrade
$ sudo aptitude -y install build-essential
$ sudo aptitude -y install libsqlite3-dev
$ sudo aptitude -y install libreadline6-dev
$ sudo aptitude -y install libgdbm-dev
$ sudo aptitude -y install zlib1g-dev
$ sudo aptitude -y install libbz2-dev
$ sudo aptitude -y install sqlite3
$ sudo aptitude -y install tk-dev
$ sudo aptitude -y install zip
```

① aptitude类似Redhat的rpm，属于Linux“方言命令”。——译者注

② 运行命令前须先联网。——译者注

在命令中，-y的作用是告知安装程序：在安装过程中，如果遇到Y或者N的提问，一律以Yes作为默认答案。如果你需要在安装过程中逐个确认，可以将-y去掉，再执行该命令。

build-essential 能够在Ubuntu上编译安装Python所必备的一系列工具链（如gcc、make等）。这些基本的工具链，在Python以及接下来提及的模块和包中，均会作为必要的使用工具，在这里就先行一步一起安装。①

接下来我们安装Python相关的包。

Python相关包的安装

```
# python-dev包的安装
$ sudo aptitude -y install python-dev

# Distribute包的安装
$ sudo chmod -R 0775 /usr/local
$ sudo chgrp –R bpbook /usr/local
$ wget http://python-distribute.org/distribute_setup.py
$ sudo python distribute_setup.py
```

Distribute 是支持Python模块构建与导入的工具包，虽然使用apt也能够安装，但通常安装的版本较老，不如将源代码下载下来，进行编译安装较为妥当。关于Distribute包的详细信息可以参考本书第7章。

Python相关包的安装到这里就结束了，最后我们从Ubuntu上确认一下默认安装的Python版本。

Python版本的确认

```
$ python -V
Python 2.7.2+
```

输入上面的命令，就可以知道本机现在所安装的Python版本，在Ubuntu 11.10上所安装的版本是Python 2.7.2 +。

01.01.02 easy_install 与pip

Python有一个叫做PyPI（Python Package Index，http://pypi.python.org/pypi）的公共资源库，它管理着Python相关的各个功能包②。任何人都可以登录该主页去下载Python相关的包资源。在使用Python开发某些软件时，我们经常会从PyPI下载安装一些必要的功能包。

NOTE

PyPI的发音为/P I Pi/。

可以想象，同Python和PyPI相类似的组合有这样几个：Perl和CPAN（http://www.cpan.org/）、Ruby和RubyGems（http://rubygems.org/）、PHP和PEAR（http://pear.php.net/）。

① 该命令表示更改/usr/local的目录所属用户组，读者可根据自己情况修改。——译者注

② 熟悉Perl的读者可以将PyPI理解为Perl中的CPAN。——译者注