

开发人员专业技术丛书



*Python Web  
Development with Django*

# Django Web开发指南

Jeff Forcier

(美) Paul Bissex 著

Wesley Chun

徐旭铭 等译



机械工业出版社  
China Machine Press

开发人员专业技术丛书

本书适用于Python框架Django初学者。 Django Web开发技术人...  
Engine上使用Django。 参与Django项目。  
禁止运行Django。 实用Django开发工具。 发现。 评估。 使用Django应用程序。 在Google App  
系统。 Livelihood。 Pastiche。 高级Django编程。 高级Django部署。 附录内容包指令行基础。 安  
装。 定义和使用模型。 URL。 HTTP机制和视图。 模板和表单处理。 Photo Gallery。 内容管理  
系统。 全书分为12章和6个附录； 内容包括。 Django Python实践。 Django集成。 构建一个Blog。  
Django 1.0版的各种主要特性所需的技术。 工具以及概念。

*Python Web*

*Development with Django*

# Django Web开发指南

Jeff Forcier

Paul Bissex 著

Wesley Chun

徐旭铭 等译



机械工业出版社  
China Machine Press

许是未就业市民人技开

本书讲述如何用Python框架Django构建出强大的Web解决方案, 本书讲解了使用新的Django 1.0版的各种主要特性所需要的技术、工具以及概念。

全书分为12章和6个附录, 内容包括, Django Python实战, Django速成: 构建一个Blog, 起始, 定义和使用模型, URL、HTTP机制和视图, 模板和表单处理, Photo Gallery, 内容管理系统, Liveblog, Pastebin, 高级Django编程, 高级Django部署。附录内容包括命令行基础, 安装运行Django, 实用Django开发工具, 发现、评估、使用Django应用程序, 在Google App Engine上使用Django, 参与Django项目。

本书适用于Python框架Django初学者, Django Web开发技术人员。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Python Web Development with Django* (ISBN 978-0-13-235613-8) by Jeff Forcier, Paul Bissex, Wesley Chun, Copyright © 2009 .

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

本书封面贴有Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2009-2370

### 图书在版编目(CIP)数据

Django Web开发指南 / (美)杰佛 (Jeff, F.) / (美)鲍尔 (Paul, B.), (美)陈仲才 (Wesley, C.) 著; 徐旭铭等译. —北京: 机械工业出版社, 2009.5

(开发人员专业技术丛书)

书名原文: Python Web Development with Django

ISBN 978-7-111-27028-7

I. D… II. ①杰… ②鲍… ③陈… ④徐… III. 软件工具—程序设计—指南  
IV. TP311.56-62

中国版本图书馆CIP数据核字(2009)第065987号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 李东震

北京京北印刷有限公司印刷

2009年5月第1版第1次印刷

186mm×240mm·18.25印张

标准书号: ISBN 978-7-111-27028-7

定价: 49.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换  
本社购书热线: (010) 68326294

## 译者序

计算机行业真是一个很有意思的行业，它每天都发明无数新名词却又拒绝它们进入生产领域。一项技术往往需要数十年的成长才会被业界慢慢接受，而其中的大部分还来不及成浪就已死在沙滩上了。Java当初熬了快十年才火起来，C++即使借了C的光也是多年媳妇熬成婆。Python比Java发明的更早，早期和Perl抢饭吃，虽有Zope/Plone这样出色的作品，依旧难逃曲高和寡的命运。

随着Web 2.0的改革号角吹响，开发人员们开始意识到，轻型的框架才是可持续发展的硬道理。一时间，小到只有一个文件的web.py，大到像Quixote等能支持豆瓣(douban.com)这样大型应用的平台百花齐放，连Python的创始人Guido van Rossum都忍不住跑来凑热闹，扬言要挑一个顺手的用用。那么最后到底是谁入了Guido的法眼呢？那就要隆重推出本书的主角——Django了！能被Guido看中那就算一只脚伸进了Google，这不，Google的新概念云计算的产品之一——Google App Engine (GAE) 已率先支持了一个剪裁过的Django框架，本书会向你介绍如何在GAE上运行Django的程序。

相比其他Python Web框架，Django更为一体化，它安装简单且灵活多变，这很符合Python开箱即用的特点。选择Django，你无需安装其他组件就能写程序运行了，同时它的灵活性体现在每个部分都能拆下来换装其他组件。简而言之，它的集成度高又保持了松耦合，这一点是相当了不起的。开发人员可以在掌握了一些基础后，把各个组件换成自己顺手的工具，这对快速开发要求很高的Web 2.0来说至关重要。如果你还在Web开发的门口徘徊不决的话，不妨来试试Django，或者你可以访问CPUG的wiki来看看Python主流框架从而加以比较。

<http://wiki.woodpecker.org.cn/moin/PyWebAppFrameworks>

<http://wiki.woodpecker.org.cn/moin/PyWebFrameVs>

最后，感谢机械工业出版社华章公司的陈冀康老师和本书的编辑，没有你们就没有这本书中文版的出版。

徐旭铭

# 前言

## 欢迎使用 Django

欢迎来到 Django 的世界，很高兴能和你一起进行这趟旅程。你会发现有了这个强大的 Web 框架，做每件事情都变得便捷起来——从设计开发新应用到不用大刀阔斧地修改代码就能为现有代码提供新的特性和功能。

## 关于本书

市面上已经有了一些讲解 Django 的书籍，但是本书的特别之处在于它着重介绍的三个方面：Django 基础、各种示例以及 Django 的进阶内容。我们希望写出一本关于这个主题最完整的教程，无论你的背景是什么都能读懂它。同时，你还能完整地了解这个框架以及它的能力。

## 章节指引

图 P.1 根据你对 Python 和 Django 的了解程度给出了不同的起点。当然，我们的建议是最好从头读到尾，不过如果时间比较紧张的话，这张图或许能提供给你一些帮助。不管你的水平如何，你都可以去阅读代码，毕竟这是学习了解应用程序最好的方法之一。此外我们还提供了一份详细的章节指引来帮助决定从哪里开始阅读。

**第一部分：入门**

第一部分向 Django 以及 Python 的新用户介绍了基本的内容，不过即使是有经验的读者，我们也推荐您读一下第 3 章，“起始”。

### 第 1 章， Django Python 实战

这一章为那些不了解 Python 的读者做了一下介绍。本章不仅展示了基本语法，还进一步深入介绍了 Python 的内存模型、数据类型，特别是在 Django 里大量使用的结构。

### 第 2 章， Django 速成：构建一个 Blog

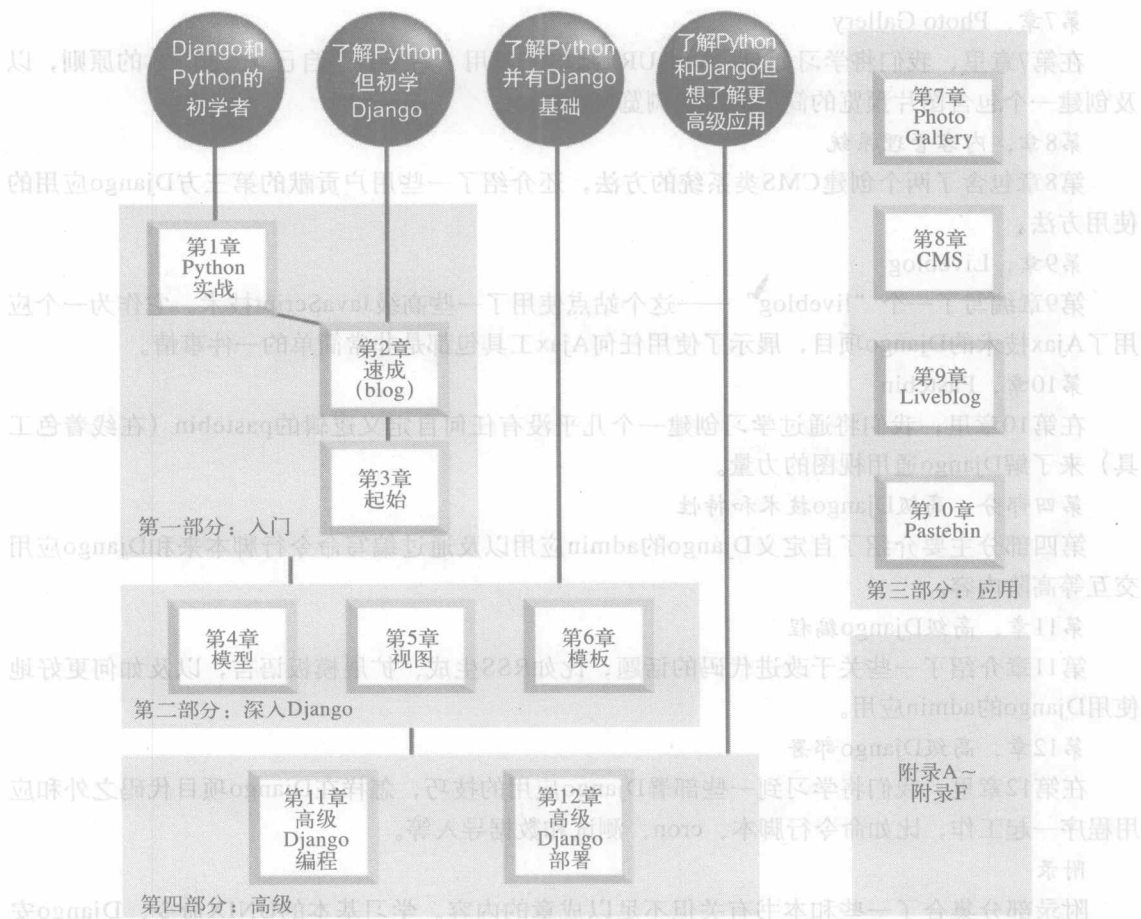
这一章是为那些希望跳过 Python 介绍，直接在 15~20 分钟完成一个 Django 应用的读者准备的。本章完美地展示了这个框架的强大能力。

### 第 3 章， 起始

对那些更没耐性的读者，这一章介绍了开发 Web 应用基础的方方面面（对新手和老鸟都有益）。之后，我们会讲解这些概念是怎样和 Django 结合起来的、Django 的设计哲学以及它独立于其他 Web 应用框架的魅力所在。

## 第二部分：深入 Django

第二部分覆盖了框架所有基本组成部分。为第三部分中的例子打下了基础。



图P.1 根据你的Python以及Django经验所建议的阅读指引

#### 第4章，定义和使用模型

在第4章里，我们将学习如何定义和使用数据模型（model），以及Django里对象关系映射（ORM）的基础，包括从简单的值到复杂的关系。

#### 第5章，URL、HTTP机制和视图

这一章详细介绍了Django处理URL和HTTP协议的方法，这包括中间件层，以及如何使用Django简便快捷的通用视图（view）等。另外还介绍了怎样自定义或者重新写一个视图。

#### 第6章，模板和表单处理

第6章介绍了框架最后一个主要的部分，Django的模板语言和它处理表单的机制。这包含了向用户显示数据以及从用户那里获取数据。

#### 第三部分：Django应用实例

在第三部分里，我们将创建4个不同的应用，每一个都展示了Django开发中不同的部分和组件。它们将介绍一些新构想，并且扩展在第一部分和第二部分里讲解的概念。

## 第7章, Photo Gallery

在第7章里,我们将学习如何在你的URL结构上应用“不要重复自己”(DRY)的原则,以及创建一个包含图片预览的简单的照片浏览应用程序。

## 第8章, 内容管理系统

第8章包含了两个创建CMS类系统的方法,还介绍了一些用户贡献的第三方Django应用的使用方法。

## 第9章, Liveblog

第9章编写了一个“liveblog”——这个站点使用了一些高级JavaScript技术。它作为一个应用了Ajax技术的Django项目,展示了使用任何Ajax工具包都是非常简单的一件事情。

## 第10章, Pastebin

在第10章里,我们将通过学习创建一个几乎没有任何自定义逻辑的pastebin(在线着色工具)来了解Django通用视图的力量。

## 第四部分: 高级Django技术和特性

第四部分主要介绍了自定义Django的admin应用以及通过编写命令行脚本来和Django应用交互等高阶内容。

## 第11章, 高级Django编程

第11章介绍了一些关于改进代码的话题,比如RSS生成、扩展模板语言,以及如何更好地使用Django的admin应用。

## 第12章, 高级Django部署

在第12章里,我们将学习到一些部署Django应用的技巧,怎样在Django项目代码之外和应用程序一起工作,比如命令行脚本、cron、测试和数据导入等。

## 附录

附录部分集合了一些和本书有关但不足以成章的内容。学习基本的UNIX命令、Django安装和部署策略、开发工具等。

### 附录A, 命令行基础

附录A介绍了基本的UNIX命令行。如果你从来没接触过的话,这个附录将对你非常有用。

### 附录B, 安装运行Django

在附录B里,我们将学习安装所有运行Django所需的必要组件,包括各种可能的数据库和Web服务器,以及一些特定的部署策略的小技巧。

### 附录C, 实用Django开发工具

附录C简要介绍了一些你可能熟悉也可能不熟悉的开发工具,比如代码控制、编辑器等。

### 附录D, 发现、评估、使用Django应用程序

出色的程序员写代码,但是伟大的程序员重用代码!在附录D中,我们来分享一些关于到哪里以及如何寻找可重用的Django应用的经验。

### 附录E, 在Google App Engine上使用Django

附录E从一个独特的视角出发,考察了Google新的App Engine如何使用Django,以及学习

如何把你的Django应用放到App Engine框架上去运行。

附录F, 参与Django项目

附录F讨论了如何向Django项目贡献代码并且成为社区的一份子。

## 编写体例

在本书中, 我们使用粗体来介绍新的和重要的术语, 用斜体来表示强调, `http://links/`表示URL, 用等宽字符来描述Python变量名和命令行命令等。如果有多行的代码和命令, 则把它们包含在等宽字符的块里, 像这样:

```
>>> print "This is Python!"
This is Python!
```

在撰写这本书和示例应用期间, 我们用到了所有三个主要的平台——Mac OS X、Linux和Windows。另外, 我们还测试了所有主要的浏览器 (虽然可能没有在书里的图中显示出来), 包括Firefox、Safari、Opera和Internet Explorer。

## 本书资源

你可以通过[authors@withdjango.com](mailto:authors@withdjango.com)来和作者联系。我们的网站<http://withdjango.com>包含了大量辅助的资料, 本书也在很多地方有所提及。

Jeff Forcier  
New York, NY  
2008年8月

Paul Ritzler  
Northampton, MA  
2008年9月



## 致谢

虽然我的名字可能是排在作者列表里的头一个，但是如果没有另外两位作者的努力和贡献的话，这本书是不可能完成的。Paul和Wesley是两位出色的绅士学者，能和他们共事感觉很棒。

说到绅士学者，Django的核心团队也一样担得起这个名号。最开始的四大金刚——Adrian Holovaty、Jacob Kaplan-Moss、Simon Willison和Wilson Miner——为Django奠定了（而且继续在贡献）一份坚实的基础，在这之上，Malcolm Tredinnick、Georg Bauer、Luke Plant、Russell Keith-Magee和Robert Wittams进一步扩展了这个基础。要知道我不是一个容易被感动的人，但是这里的每一位都给了我很大的灵感。

我还要感谢另外两位“Django人”和IRC高手，Kevin Menard和James Bennett，以及NYCDjango组的各位朋友，这是Django社区里典型的牛人汇聚的地方。

最后，大力感谢Pearson的各位同仁，包括编辑和技术评论等（Wesley会提到你的！），这里特别要感谢排版的工作人员，非常感谢你们仔细的校对。

Jeff Forcier

New York, NY

2008年8月

感谢所有围绕在Django、Python和其他Web应用开发的开源基础设施的社区。成百上千热忱的开发人员和维护者的辛勤努力让全世界都能免费用上强大的软件。

我的合作者让我受益匪浅，他们不仅给这项任务带来了必要的知识和技能，还为之做出了很大的贡献。尽管我们分散在美洲的不同角落，我还是有幸见到了Jeff和Wes。

感谢西马萨诸塞州开发组，感谢和你们之间许多非常有趣的geeky的讨论，感谢你们对本书的极大热情。

感谢Hallmark Institute of Photography的主席George J. Rosa III，给了我极大的信任和鼓励，让我选择最好的工具（当然，包括Django）来最好地完成工作。

在2008年夏天经历了一场严重的车祸后，我得到了很多来自家庭、朋友和社区的关心和支持。每一份祝福、每一张卡片、每一分善款和每一顿饭都是很重要的。非常感谢你们。

最后，感谢我亲爱的妻子Kathleen，感谢你的支持、贤惠和爱。

Paul Bissex

Northampton, MA

2008年9月

编写我的第二本书实在是一个非常棒的体验。首先我想要向我的两位出色的合作者致敬，十分高兴能和他们一起工作。他们有汲取他人的Python技巧和介绍Django的经验的能力。我很高兴能合作完成这本出色的Django教程，并且期待能和他们有机会在写作或是教学上再次合作。能编写这样一本完全由开源项目组成的书是非常令人欣慰的，开发人员每天都在使用相同的工具开发改变社会的软件产品。

我要特别感谢Debra Williams Cauley从我第一天加入这个项目开始就帮助我们管理整个进程。虽然在这过程中我们发生了很大的人事变化，但是她一直都让我们专注于写作上面。就像她也赞同的一样，我们的理念是要为社区写出一本“合适的书”，而不仅仅是一本迎合市场需要的Django教程。感谢我们所有技术评论，Michael Thurston（策划编辑），Joe Blaylock和Antonio Cangiano，以及所有在这本书草稿阶段就在Rough Cuts上留下反馈的人，因为你们，这本书才会更加完美。我还要感谢Matt Brown，Django Helper for Google App Engine的首席维护，在审阅附录E中给予的协助，以及Eric Walstad和Eric Evenson最后的终审和评注。

最后，如果没有我们家里团结一致的支持，我是不可能完成这本书的。

Wesley Chun  
Silicon Valley, CA

2008年8月

## 引言

如果你是一名Web工程师，那么Django有可能彻底改变你的生活。至少它改变了我们。

只要你知道一点构建动态网站是怎么回事的话，那么就一定能体会到不断重复地发明某些标准特性是多么痛苦的一件事情。你得创建数据库结构，把数据导入导出数据库，处理URL，验证用户输入，提供编辑工具，还得关心安全性和可用性……

### Web框架的前世今生

终于，你意识到每次都重新实现这些特性实在是太浪费生命了。所以，你决定要重新开发一套自己的库来提供这些功能，或者说，从你最新、最伟大的“创造”中把这些库提取出来。然后，如果要开始一个新项目的話，你第一件要做的事情就是安装你的库。这能大大节约你的工作时间。

但是，事情可没那么简单。如果客户需要的特性不在你的库里怎么办，没关系，加进来就好了。而每个客户都需要不同的东西，结果就变成你在每个服务器上安装的库都有不同的版本。这绝对是没有办法维护的。

有了教训以后，你回过头来把基础库和最好的add-on从各个项目里拿出来重新组合在一起。对绝大多数项目来说你不再需要直接调整库代码，只需要改动一下配置文件就可以了。虽然你的代码库越来越大、越来越复杂，但是它也变得非常强大。

恭喜！你已经完成了一个Web框架。

不过，只要你（或者你的团队、公司或者客户）还在使用它，你就得保证它能正常工作。下次升级操作系统、Web服务器、编程语言的时候会不会把它弄挂了？能不能在未来引入修改时不需要伤筋动骨？它能不能支持复杂但是重要的功能如会话管理、本地化，又或者数据库事务？你的测试覆盖率有多少？

### 更好的做法

你捧着这本书的原因就是希望能找到更好的做法。要是有一个强大灵活、实现优雅、经历过完整测试而又不要你自己来维护的框架就好了。

你希望用一门真正的编程语言来编写代码，它必须是干净强大、成熟的语言，需要有大量的文档支持。你希望它有一个漂亮的标准库以及大量高品质的第三方库可以满足各种需要，比如生成CSV文件、生成饼状图、科学计算或者是图像处理等。

你需要一个背后有活跃的、助人为乐的用户和开发社区的框架。这个框架能作为一个整体流畅的工作，而其组件又是松散耦合，可以让你在需要的时候随时替换。

一句话，你想要的就是Python和Django。我们编写这本书的目的就是要帮助你尽量简单快

速地在实际环境下学习使用Django。

## 冲出堪萨斯，走向世界

Django原本是Adrian Holovaty和Simon Willison为堪萨斯州劳伦斯的一家家族媒体World Online编写的。当时是需要它能够迅速开发出数据库驱动的新闻系统。

在Web领域里证明了自己的实力后，Django在2005年7月作为一个开源项目公开发布。搞笑的是，虽然这个时候很多人都觉得Python已经有太多的Web框架了，Django还是迅速获得了大量追随者。今天，Django已经不仅仅是Python框架里的领头羊了，就是与其他所有Web框架里相比也毫不逊色。

当然，World Online还在大量使用Django，它的一些核心工程师还在那里工作。但是自从Django项目开源以来，全世界的公司和组织都已经把它用在了无数大大小小的项目里了。这里有一份不完整的列表：

- The Washington Post
- The Lawrence Journal-World
- Google
- EveryBlock
- Newsvine
- Curse Gaming
- Tabblo
- Pownce

虽然这里还有成千上百的Django网站没有列出，但是Django的传播和成长已经势不可挡，未来会有无数受欢迎的站点采用Django开发。我们希望你也其中之一。

## 用Python和Django更好地进行Web开发

Web开发通常是一件很烦琐的工作。你必须要从测试起就面对浏览器的不兼容性、疯狂的爬虫、宽带和服务器的限制以及总体框架等一系列挑战。

虽然我们相信本书将Django的基础讲解得非常完备了，但是我们还要讨论很多其他难点——你80%的时间都会花在这些20%的工作上。我们和许多Django工程师一起工作讨论，听取他们的意见和建议并且帮助他们解决问题。在编写本书的过程中，我们都提醒自己注意这些问题和难点。

要是我们觉得Django和Python不好的话，我们也不会费这么大力来为它们写一本书。但是当遇到有限制或者有陷阱的时候，我们还是会如实相告的。我们的目标是帮助你完成任务。

# 目 录

译者序	2.10 总结	60
前言	第3章 起始	61
致谢	3.1 动态网站基础	61
引言	3.2 理解模型、视图和模板	63
	3.3 Django架构总览	64
	3.4 Django的核心理念	66
	3.5 总结	68
	<b>第二部分 深入Django</b>	
	第4章 定义和使用模型	69
	4.1 定义模型	69
	4.2 使用模型	80
	4.3 总结	91
	第5章 URL、HTTP机制和视图	92
	5.1 URL	92
	5.2 HTTP建模：请求、响应和中间件	96
	5.3 视图与逻辑	100
	5.4 总结	105
	第6章 模板和表单处理	106
	6.1 模板	106
	6.2 表单	112
	6.3 总结	123
	<b>第三部分 Django应用实例</b>	
	第7章 Photo Gallery	125
	7.1 模型	126
	7.2 准备文件上传	127
	7.3 安装PIL	128
	7.4 测试ImageField	128
	7.5 构建自定义File变量	130
第1章 Django Python实战		1
1.1 Python技术就是Django技术		1
1.2 入门：Python交互解释器		2
1.3 Python基础		3
1.4 Python标准类型		5
1.5 流程控制		19
1.6 异常处理		21
1.7 文件		23
1.8 函数		24
1.9 面向对象编程		33
1.10 正则表达式		35
1.11 常见错误		36
1.12 代码风格		41
1.13 总结		43
第2章 Django速成：构建一个Blog		44
2.1 创建项目		44
2.2 运行开发服务器		46
2.3 创建Blog应用		47
2.4 设计你的Model		48
2.5 设置数据库		48
2.6 设置自动admin应用		51
2.7 试用admin		52
2.8 建立Blog的公共部分		55
2.9 最后的润色		57

7.6	使用ThumbnailImageField	134
7.7	设置DRY URL	135
7.8	Item应用的URL布局	137
7.9	用模板把它们都串在一起	138
7.10	总结	143
第8章	内容管理系统	144
8.1	什么是CMS	144
8.2	Flatpages	144
8.3	超越Flatpages: 一个简单的自定义CMS	147
8.4	改进建议	162
8.5	总结	163
第9章	Liveblog	164
9.1	究竟什么是Ajax	164
9.2	设计应用程序	165
9.3	应用程序布局	166
9.4	加入Ajax	169
9.5	总结	176
第10章	Pastebin	177
10.1	定义模型	177
10.2	创建模板	179
10.3	设计URL	180
10.4	试运行一下	181
10.5	限制最近Paste显示的数量	184
10.6	语法高亮	185
10.7	通过Cron Job清除	186
10.8	总结	187

#### 第四部分 高级Django技术和特性

第11章	高级Django编程	189
11.1	自定义Admin	189
11.2	使用聚合	193
11.3	生成下载文件	195
11.4	用自定义Manager来增强Django ORM	200
11.5	扩展模板系统	202
11.6	总结	211
第12章	高级Django部署	212
12.1	编写工具脚本	212
12.2	自定义Django codebase	214
12.3	缓存	215
12.4	测试Django应用	223
12.5	总结	229

#### 附 录

附录A	命令行基础	231
附录B	安装运行Django	240
附录C	实用Django开发工具	254
附录D	发现、评估、使用Django应用程序	261
附录E	在Google App Engine上使用Django	264
附录F	参与Django项目	273
后记		275

# 第一部分 入门

第1章 Django Python实战

第2章 Django速成：构建一个Blog

第3章 起始

## 第1章 Django Python实战

欢迎使用Django和Python！在开始介绍Django之前，我们将会给你介绍一下这个作为Django应用的基石的语言——Python。如果你有其他高级语言编程经验（C/C++、Java、Perl、Ruby等）的话，这一章的内容会比较容易理解。

不过要是你没有任何经验的话也不要紧，Python本身就非常适合作为入门语言来学习。市面上有很多教你如何Python编程的书。具体你可以参考本章的结尾。我们推荐那些新人先看看这些资源。一旦掌握了基础，反过来再读本章的时候会有更深的体会。

这一章要介绍的Python，主要是语言中和Django开发有关的核心特性和技巧。光有基本的Python技术是无法高效地开发Django的，你还需要知道一些更多Python内部的东西，这样当遇到这些特定的特性、这方面的知识和Django的需求时，就不会无所适从。对Python新人或者编程新人来说，先阅读一些其他基本的Python知识以及本章的内容会都可以让你获益良多——选择哪种方式完全可以按照你自己的节奏来。

### 1.1 Python技术就是Django技术

Django提供了一个高级的框架，用它只需要很少的几行代码就能完成一个Web应用。这种轻盈、强大、灵活的框架让你在设计方案时无需太多考量。Django是以Python写成，Python是一门面对对象的应用程序开发语言，它同时兼具系统语言（如C/C++和Java）的强大和脚本语言（如Ruby和Visual Basic）的灵活迅速。这让Python的用户创建能解决各种疑难杂症。

在这一章里，我们要向你介绍一些我们觉得如果要成为一名出色的Django工程师就必须知道的Python技术。我们主要关注是那些作为Django工程师一定要知道的Python概念，而不是再造一份通用的Python教程。所以基本上这一章里没有什么Django代码。

Python 2.x还是3.x

在编写本书的时候，Python正在从2.x向新一代的版本3.0迁移。3.x版本家族不保证对

旧版本的兼容性，所以用2.x编写的代码完全有可能无法在3.x下工作。但是Python的核心开发团队正在努力使这一转换尽可能的顺利：他们会提供可靠的2.x-to-3.x转换工具，并且给予足够的时间保证大家都能顺利完成。

Django的核心团队现在还不打算立刻迁移到3.0上去（和绝大多数面向框架的项目一样，这样的转换破坏性非常大，一定要非常小心），所以这里我们只是提一下这个转换而已。Django只有在绝大多数用户都准备好的时候才会进行升级。

## 1.2 入门：Python交互解释器

交互解释器是在Python开发过程中最有用的工具之一，有了它就不用去创建、编辑、保存，再运行源文件了，你可以直接用它测试一小段代码。它不但会检查代码的正确性，还能让你在把这段新代码加入到源文件之前进行各种不同的操作，比如查看数据结构、修改键值等。

在阅读本章的过程中，我们建议你启动Python解释器来直接感受一下。大多数Python IDE都能轻易启动它，或者也可以直接从命令行或是你操作系统的应用程序菜单里启动。这样，你就会有一个切身的体会，对Python以及接下来的Django会更有信心。就算是Python高手，哪怕有十多年编程经验的老鸟，也还是每天都离不开Python shell的！

在本书中，你会看到很多用Python shell的交互提示符：`>>>`打头的代码段。你可以在阅读过程中直接把这些例子输入到解释器里去。像是这样：

```
>>> print 'Hello World!'
Hello World!
>>> 'Hello World!'
'Hello World!'
```

`print`是非常好用的命令。它不仅向你的用户提供了相关的应用程序信息，还是一个无可替代的调试工具。虽然也有不显示调用`print`就能“打印”变量值的方法，就像我们前面那样，但是这通常会产生和`print`不同的结果。

请注意Hello World例子中的不同之处。当直接把一个对象丢给解释器的时候，它会用引号告诉你这是一个字符串。而在使用`print`语句时，引号则不会显示出来，因为你告诉了解释器去显示字符串的内容（当然是不包括引号的了）。下面这个例子就和字符串的情况有点不同（对数字来说显示都是一样的）。

```
>>> 10
10
>>> print 10
10
```

但是对稍后我们讨论到的复杂对象来说，这个差异就会非常明显，这是因为Python会让你决定在有没有使用`print`时对象应该如何行为。

虽然后面会详细讲解变量和循环，这里我们先来体验一下稍微复杂一点的Python是什么样的。这里有几个`for`循环。

```
>>> for word in ['capitalize', 'these', 'words']:
...     print word.upper()
```



```

CAPITALIZE
THESE
WORDS
>>> for i in range(0, 5):
...     print i
...
0
1
2
3
4

```

Python语法里非常重要的一个方面就是没有区分代码块的花括号（{}）。我们用对齐来代替括号：对于给定的一段Python代码，它们必须有相同的对齐，通常是4个空格（任意数目的空格或者tab也都可以）。如果你习惯于其他语言的话，这可能需要一点时间来适应。但是过一段时间以后，你就会发现其实这种方式也没有看上去那么糟糕。

关于解释器最后一点要说的就是：当你熟练使用解释器了以后，你应该考虑试试另一个类似的工具IPython。要是你喜欢交互解释器的话，IPython的能力完全是在另一个数量级上的哦！它提供了系统shell访问、命令行记数、自动对齐、命令行历史等许多特性。你可以在<http://ipython.scipy.org>上找到更多关于IPython的信息。它不是随Python一起发布，而是一个第三方应用。

### 在解释器里使用Django

要是能在Python解释器里调试Django应用或框架的话那就太方便了。但是如果你只是按照常规启动解释器并且试着导入Django模块的话，你只能得到一个DJANGO\_SETTINGS\_MODULE没有设置的错误。为了方便起见，Django提供了一个manage.py shell命令，它能进行一些必要的环境设置来避免这个问题。

如果你安装了iPython的话，manage.py shell会默认使用它。如果你安装了iPython，但是又希望使用标准Python解释器的话，可以运行manage.py shell plain。虽然在后面的例子里我们是继续使用默认的解释器，不过我们还是强烈推荐iPython。

## 1.3 Python基础

在这一节里我们要介绍Python基本的几个方面，包括注释、变量、运算符和基本Python类型。下面几节里会更详细地介绍Python的主要类型。绝大多数Python（和Django）代码都是保存为以.py扩展名为结尾的文本文件，这是告诉系统一个Python文件的标准做法。你还会看到相关的扩展名诸如.pyc或者.pyo，这不会影响你的使用，现在可以暂时先忽略它们。

### 注释

Python的注释是用井字符（#）表示的。如果它是一行里第一个字符，那么这整个一行就