

人生苦短，我用Python。
在项目实践中学习Django，简单快速建网站！



全新升级，基于Django 2版本



齐伟 编著

“跟老齐学Python”系列图书之一
面向已经“轻松入门”之后的读者

以实际Web开发项目为主线，既综合应用Python基础知识，又学习新的网站开发技能！

跟老齐学
Python
Django实战 **第2版**

—— 齐伟 编著 ——

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是以 Python 为基础进行 Web 应用开发的进阶读物。书中以一个实例项目为主线，使用 Django 2 开发框架，在实践中边学边做，理论联系实际。每节都配有思维导图，使读者对项目需求一目了然；每章都有知识点和文档导读，引导读者“知其所以然”。相信认真阅读本书的读者，不仅能够得到“鱼”，更能得到“渔”，从而具备独立开发项目的能力。

本书适合已经具有 Python 基础技能、进行 Web 应用开发的读者阅读。

如有此意愿，但尚缺乏 Python 基础技能者，推荐阅读《跟老齐学 Python：轻松入门》。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

跟老齐学 Python: Django 实战 / 齐伟编著. —2 版. —北京: 电子工业出版社, 2019.1

ISBN 978-7-121-35503-5

I. ①跟… II. ①齐… 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 (2018) 第 258023 号

策划编辑: 高洪霞

责任编辑: 牛 勇 特约编辑: 赵树刚

印 刷: 北京京师印务有限公司

装 订: 北京京师印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 17.5 字数: 448 千字

版 次: 2017 年 10 月第 1 版

2019 年 1 月第 2 版

印 次: 2019 年 1 月第 1 次印刷

定 价: 79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：(010) 51260888-819, faq@phei.com.cn。

序

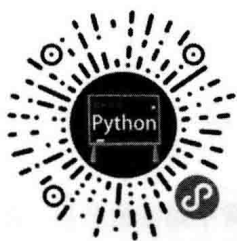
按照 Django 的版本发布计划，现在已经是 Django 2 的时代了，所以，本书也要跟进。在本书第 2 版的代码中，使用了 Django 2，源码依然放在：<https://github.com/qiwsir/DjangoPracticeProject>。

本书的第 2 版相对第 1 版，不仅是 Django 版本的迭代，对部分内容也进行了修改，采用了读者对第 1 版提出的各种意见和建议，在此对众多热心读者表示感谢。

Web 开发所需要的技术比较庞杂，那些“从前到后”都能做的开发者被称为“全栈工程师”。但是，如果本书宣称“全栈”，肯定是骗人的——请读者注意，凡是宣称“全栈”的书或课程，都要小心。要达到“全栈”的目的，不仅需要读书，更需要项目的积累。

为了协助读者了解 Web 开发的各种相关技术，我特意配合本书开发了一个小程序（名为：跟老齐学）。希望能够通过小程序，实现“把书读厚”的目的。至于“读薄”，抑或“放弃”，则应视读者个人的修为而定了。

扫描二维码，关注小程序（小程序名称：跟老齐学）。



除了小程序之外，QQ 群（新的群号：746028518）依然是读者之间的交流园地；网站（itdiffer.com）也会一如既往地提供相关阅读资料。

非常感谢编辑老师为本书所付出的辛苦工作，以及刘燕清先生在百忙之中帮助我调试代码，在此一并表示感谢！

齐伟

2018年7月

前言

是的，你应该开始阅读本书了。

如果读者已经通过《跟老齐学 Python：轻松入门》这本书完成了 Python 入门，那么接下来就要用 Python 做一些实际项目了，做什么呢？可以做的东西真不少，比如数据统计、爬虫、机器学习、人工智能等，当然还有不可或缺的 Web 开发。

本书就是要带领读者：

- 学习 Django（特别声明，本书中的项目选用 Django 2）；
- 完成一个项目；
- 边做项目边学知识。

因此，本书按照完成一个项目从易到难的顺序，以项目为主线逐渐展开对 Django 的学习。Django 是一种在 Web 开发中应用非常广泛的开发框架，它能够让程序员非常快捷地建设一个网站，并且支持网站开发中各种复杂的业务。如果需要快速建设一个网站，并且希望后期还能通过迭代开发实现网站功能和性能的提升，那么 Django 是一个不错的选择。

Django 的官方网站中有不少内容完备的文档，当然，这种完备不是针对初学者而言的。对于初学者来说，也许官方文档太简要了，因此本书的作用就是将初学者领入 Django 开发的大门，并引导读者熟悉使用 Django 官方文档。当读者阅读并练习完本书的项目后，就能够使用官方文档进行开发了，本书的目的也就达到了。

本书从始至终贯穿着一个项目实践，将开发实战中常用的 Django 知识串联起来，因此它不像课本那样按部就班地把每个知识点都罗列出来，而是根据项目需要，将技能融汇在实战之中。此外，如果仅从开发实践的角度学习 Django，难免会出现就问题而解决问题的现象，缺少系统和深入的学习。为此，书中的每个小节都配有“知识点”，用以扩展或者系统化有关知识，特别是在“文档导读”中会向读者提供有关网络资料，试图让读者知其然并知其所以然。

在项目进行中，读者可能会遇到暂时无法理解的部分，不要纠结于一时之感，而要跟着项目继续进行下去，或许到后面就能理解前面的内容了——“发展是硬道理”。

如果正在阅读本书的你是 Web 开发的初学者，此前没有做过任何网站开发，那么笔者特别建议遵循如下学习方法。

- 严格按照本书要求, 将所有代码都认真敲过。不要复制代码 (本书提供的源代码, 仅仅是一个参考), 要自己一个字母一个字母地敲进去。
- 将自己的调试结果与本书所示结果进行对照, 看看是否实现了预期的功能。
- 在调试不能通过时, 首先要认真检查自己的代码, 包括但不限于拼写问题、缩进问题等, 若还不能解决, 可通过 Google 搜索该异常 (错误), 请相信, 你的错误别人也同样遇到过。
- 本书所支持的网站 www.itdiffer.com 和 QQ 群 (26913719) 也是读者学习的助手。
- 本书提示读者阅读的官方文档, 一定要认真阅读。
- 本书实际上是一个 Django 项目, 项目中包含了多个应用。如果读者按照本书的要求, 第一遍就实现了应用的功能, 那只能说明本书描述无误且你的阅读能力尚可, 并不代表你已经掌握了什么。只有当你把刚刚调试通过的应用删除, 然后完全靠自己的记忆和理解, 同样实现了该应用的功能, 才能说明你掌握了相关知识技能。再次强调, 要理解相关内容, 就必须多重复几遍。

因为本书是以“项目为主线”的, 所以不会面面俱到地将 Django 的所有内容介绍一遍, 而是根据项目的功能需要, 有选择地使用和介绍有关内容。对于具体功能的实现, 书中所给出的代码也非唯一写法, 读者可以使用其他更好的方式实现。

在实际的 Web 开发项目中, 除使用 Django 框架外, 还会用到 HTML、CSS、JavaScript 等知识, 从而做出一个看起来有点设计感的页面。虽然说“人不可貌相”, 但使用 Django 做出一个漂亮的前端, 还是能够让人感觉很舒服的。因此, 书中也涉及一些前端内容, 读者在阅读时, 如果缺少相关知识, 可以实时补充。当然, 前端知识并不是学习本书所必备的, 因为凭借读者的聪明才智和无所不包的网络, 掌握应付本书所需要的前端知识是非常容易的。

读者可以在 GitHub 上 (<https://github.com/qiwsir/DjangoPracticeProject>) 获得本书的所有代码, 这些代码是最终结果, 希望不会给读者的学习带来不便。

感谢为本书的面市提供帮助的编辑们, 感谢我的妻子在本书编写过程中提供的支持。

齐 伟

2018 年 8 月

目录

第 1 章 简单的博客系统	1
1.1 Django 起步	1
1.1.1 Django 简介	1
1.1.2 安装 Django	3
1.1.3 创建项目	4
1.1.4 创建应用	6
1.1.5 网站配置	12
1.1.6 知识点	13
1.2 编写博客的数据模型类	13
1.2.1 数据模型类	14
1.2.2 发布博客文章	18
1.2.3 知识点	22
1.3 显示博客信息	24
1.3.1 显示文章标题	25
1.3.2 查看文章内容	30
1.3.3 知识点	33
第 2 章 用户管理	37
2.1 自定义模板和静态文件位置	37
2.1.1 自定义模板位置	38
2.1.2 自定义静态文件位置	38
2.1.3 通用静态文件和基础模板	39
2.1.4 重置管理后台模板	43
2.1.5 知识点	45
2.2 用户登录	46

2.2.1	创建应用	47
2.2.2	理解表单类	48
2.2.3	登录的视图函数	49
2.2.4	登录的前端界面	52
2.2.5	知识点	56
2.3	用内置方法实现登录和退出	58
2.3.1	内置的登录方法	58
2.3.2	判断用户是否登录	60
2.3.3	内置的退出方法	61
2.3.4	知识点	62
2.4	用户注册	64
2.4.1	简单注册	64
2.4.2	增加注册内容	67
2.4.3	管理新增的注册内容	71
2.4.4	知识点	72
2.5	关于密码的操作	74
2.5.1	修改密码	74
2.5.2	重置密码	79
2.5.3	知识点	85
2.6	维护个人信息	87
2.6.1	个人信息的数据模型类和表单类	87
2.6.2	展示个人信息	88
2.6.3	编辑个人信息	91
2.6.4	上传和裁剪头像图片	95
2.6.5	优化头像上传功能	103
2.6.6	对个人信息进行管理	106
2.6.7	知识点	107
第 3 章	文章管理和展示	109
3.1	管理文章栏目	109
3.1.1	设置栏目	109
3.1.2	编辑栏目	117
3.1.3	删除栏目	119
3.1.4	知识点	121

3.2	发布和显示文章.....	124
3.2.1	简单的文章发布.....	125
3.2.2	使用 Markdown.....	130
3.2.3	文章标题列表.....	132
3.2.4	知识点.....	140
3.3	删除和修改文章.....	141
3.3.1	删除.....	141
3.3.2	修改.....	143
3.3.3	设置分页功能.....	147
3.3.4	知识点.....	149
3.4	文章展示.....	151
3.4.1	新写文章标题列表.....	151
3.4.2	重新编写“查看文章”功能.....	155
3.4.3	知识点.....	156
第 4 章	拓展网站功能.....	158
4.1	梳理已有功能.....	158
4.1.1	修改导航栏.....	158
4.1.2	修改登录和注册后的跳转.....	159
4.1.3	知识点.....	161
4.2	查看作者全部文章.....	162
4.2.1	查看某作者的文章列表.....	162
4.2.2	知识点.....	166
4.3	为文章点赞.....	167
4.3.1	修改数据模型类.....	167
4.3.2	编写视图函数.....	168
4.3.3	修改模板文件.....	169
4.3.4	知识点.....	173
4.4	文章的阅读次数.....	174
4.4.1	安装 Redis.....	175
4.4.2	在 Python 中使用 Redis.....	176
4.4.3	记录阅读次数.....	177
4.4.4	显示最“热”文章.....	178
4.4.5	知识点.....	180

4.5	文章的评论功能.....	181
4.5.1	数据模型类和表单类.....	181
4.5.2	实现评论功能.....	183
4.5.3	知识点.....	185
4.6	多样化显示.....	186
4.6.1	统计文章总数.....	187
4.6.2	最新发布的文章.....	190
4.6.3	评论最多的文章.....	191
4.6.4	自定义模板选择器.....	194
4.6.5	知识点.....	195
4.7	管理和应用文章标签.....	196
4.7.1	管理文章标签.....	197
4.7.2	发布文章时选择标签.....	202
4.7.3	在文章中显示文章标签.....	204
4.7.4	推荐相似文章.....	205
4.7.5	知识点.....	207
第 5 章	收集和展示图片.....	208
5.1	收集网络图片.....	208
5.1.1	创建图片相关类.....	209
5.1.2	收集和管理图片.....	212
5.1.3	完善图片管理功能.....	217
5.1.4	知识点.....	220
5.2	展示图片.....	221
5.2.1	瀑布流方式展示图片.....	222
5.2.2	查看图片的详细信息.....	225
5.2.3	知识点.....	226
第 6 章	中场休整.....	228
第 7 章	创建在线学习应用.....	229
7.1	基于类的视图.....	229
7.1.1	最简类视图.....	230
7.1.2	读取数据.....	232
7.1.3	初步了解 Mixin.....	235
7.1.4	知识点.....	236

7.2	管理课程标题.....	237
7.2.1	判断用户是否登录.....	237
7.2.2	创建课程.....	240
7.2.3	删除课程.....	242
7.2.4	知识点.....	247
7.3	发布和学习课程内容.....	249
7.3.1	课程内容的数据库模型.....	250
7.3.2	课程内容的表单类.....	253
7.3.3	课程内容的视图.....	253
7.3.4	查看课程内容.....	256
7.3.5	注册学习课程.....	260
7.3.6	知识点.....	264
第8章	结束和开始.....	266

第 1 章

简单的博客系统

从现在开始，请读者随我一起做一个项目，这个项目的名称叫作“多用户内容发布系统”。因为很多读者是来学习的，所以还要本着“循序渐进”“深入浅出”的原则来做这个项目——这当然不是工程项目开发的原则，这是教学的原则。

本项目从博客开始。

博客，现在已经有人把它看作历史文物了，因为互联网的迅猛发展，10年前的事物就算古董了。尽管如此，博客依然可以作为一个学习的样本。本章将通过一个单用户的博客系统的开发，初步说明利用 Django 开发网站的基本步骤和网站的基本组成。当然，这仅仅是起步，并不意味着学完本章就理解 Django 了。“不积跬步，无以至千里”，最后的复杂系统也是由一个个简单的小功能堆砌起来的。

1.1 Django 起步

Django 这个词对一些人来说或许并不陌生，有一部荣获第 85 届奥斯卡金像奖影片叫作 *Django Unchained*，中文名被翻译为《被解救的姜戈》，这或许是 Django 首度被翻译为中文。但是，作为网站开发框架的 Django 与这部电影没有任何关系。

Django 的起步内容如图 1-1 所示。

1.1.1 Django 简介

一般认为 Django 开发框架（简称 Django）诞生的时间是 2003 年的金秋时节。此时千年古城苏州正桂花飘香，而大洋彼岸的美国有两位程序员在使用后来被冠名为 Django 的框架做网站开发，这是全世界首次使用。让我们感谢这两位程序员，他们是 Adrian Holovaty 和 Simon Willison。2008 年虽然遇到了金融危机，但依然挡不住技术的进步，Django 有了自己的 DSF（Django Software Foundation），后来 Django 的发展就由这个基金会来主导了。

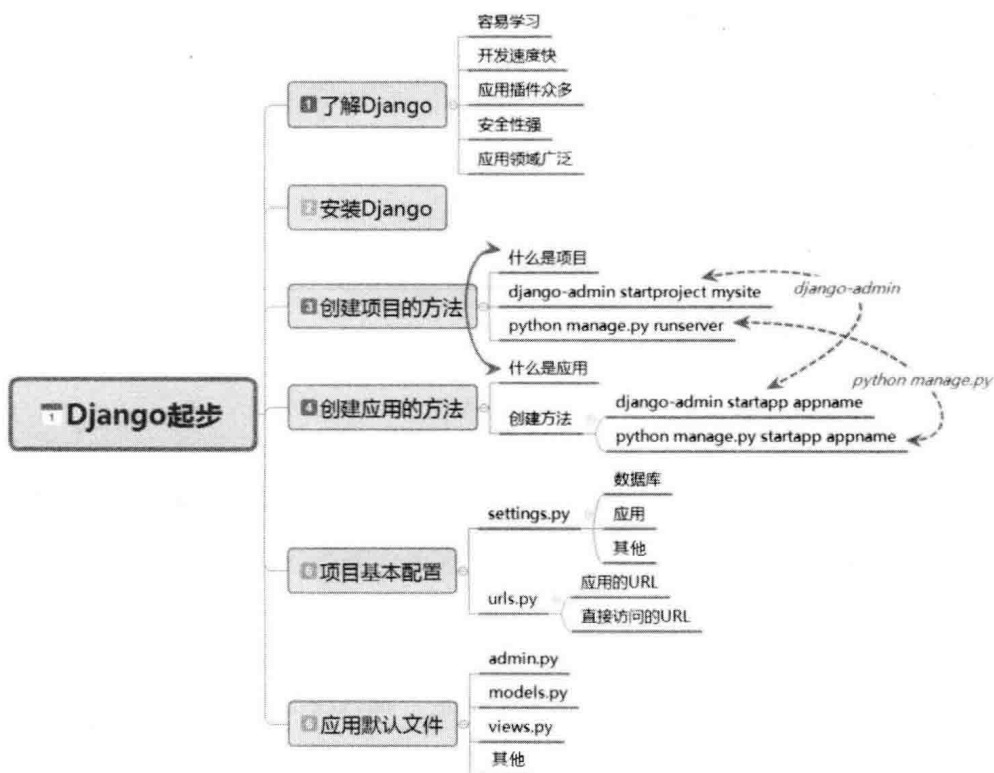


图 1-1 知识技能导图

对于 Django 的评价，借用李清照的《鹧鸪天·桂花》来表达，我认为是非常恰当的。

暗淡轻黄体性柔。情疏迹远只香留。何须浅碧深红色，自是花中第一流。
梅定妒，菊应羞。画栏开处冠中秋。骚人可煞无情思，何事当年不见收。

“自是花中第一流”，不仅仅是现在，未来依然蓬勃发展。在 Django 官方网站披露的信息中，可以看到其发展蓝图，如图 1-2 所示。

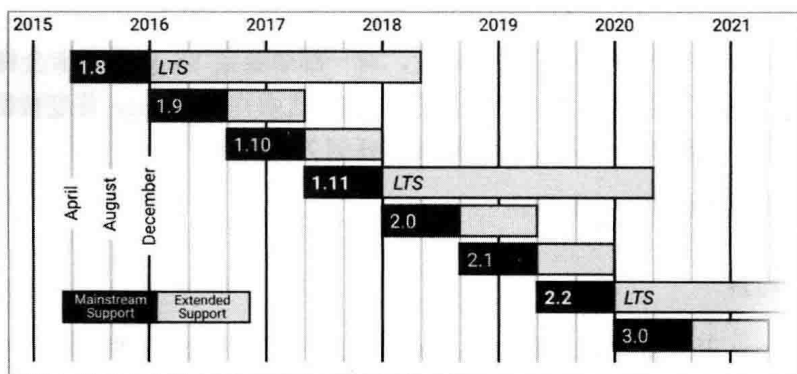


图 1-2 Django 各版本计划

关于 Django 的特点，用官方网站 (<https://www.djangoproject.com>) 上的大标题就可以概括了：“Django makes it easier to build better Web apps more quickly and with less code”。展开来说，就是如下几点。

- 容易上手，开发速度快。
- 囊括了网站开发中的用户管理、内容管理、网站地图、RSS 等常用的众多插件。
- 安全性强，比如 Django 默认解决了 SQL 注入、跨站攻击等问题。
- 应用广泛，类型多样化。使用 Django 开发的网站包括各类在线服务网站、社会组织 and 政府机构网站等，其类型包括但不限于管理系统、社交网站、计算平台等。

就开发网站的框架而言，除 Django 外，在 Python 领域还有 Tornado、Flask 等，它们各有各的特点，但 Django 的应用范围最广。

接下来就开始 Django 之旅——虽有解救姜戈的惊心动魄，但无生命之忧。

1.1.2 安装 Django

在一般情况下，读者所用的计算机操作系统上没有 Django，需要自己安装。

Django 是以 Python 为语言环境的，所以要先保证计算机上已经安装了 Python。读者如果对 Python 知之甚少，可以参阅《跟老齐学 Python：轻松入门》，这本书介绍了 Python 的基础知识。

Django 适用于 Python 3 和 Python 2.7 两种版本，如果你是一个新秀，并不是为了承接基于 Python 2.7 的旧项目开发，可以直接用 Python 3 进行 Django 的学习和开发。

从官方网站可以获悉，Django 项目团队正在不断发展 Django 的版本（见图 1-2），本书第 1 版在 2017 年刚刚发行的时候，使用的是 Django 1.10。2018 年，Django 已经演进到 Django 2.x 了。Django 2.x 相对于 Django 1.x 有部分变化，因此本书再次修订，转而采用 Django 2.x。读者在阅读的时候，可能 Django 的版本已经高于本书所使用的版本了，但是，影响不大，因为都是 Django 2.x 的范畴，小版本号的修改，仅意味着些许的优化。使用本书进行学习的时候，可以安装使用最新版的 Django，不用在意小版本号的更改。

安装 Django 的最简单方法是输入以下命令（关于 pip，请参阅《跟老齐学 Python：轻松入门》有关章节说明）：

```
$ pip3 install Django
```

安装好 Django 之后，可以先进入 Python 交互模式中，查看一下所安装的 Django 版本。

```
$ python3
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 03:03:55)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> print(django.get_version())    #打印出当前 Django 的版本号
2.1.4
```

上面的安装方法是一种简单而且常用的方法，除这种方法外，还可以下载 Django 源码进行安装：

```
git clone https://github.com/django/django.git
```

如果系统中没有安装 Git，会提示用户安装。

这时，会在当前目录中看到一个名称为“django”的目录，里面是最新版本的 Django。

随后在当前目录中进行如下操作:

```
$ pip3 install -e ./django
```

安装完毕, 系统会提示 “Successfully installed Django”。

通过这种方法得到的必然是 Django 的最新版本, 而且是正在开发中的最新版, 尚未正式发布——虽然如此, 也是能够正常使用的, Django 的开发者不会把不能使用的代码发布出来。喜欢尝试的读者可以用上面这种方式安装。

安装成功之后, 在 Python 交互模式中可以查看当前 Django 版本, 显示结果中包含 “dev” 字样, 说明我们使用的是开发版。

在一般情况下, 推荐使用 `pip install Django` 的方法来安装, 原因就是笔者提倡的保守主义。

基础设施已经建好, 下面就要创建项目了。

1.1.3 创建项目

Django 中的 “项目” (project) 可以看作一个专有名词, 因为后面还有一个与之有关的名词 “应用” (application)。

所谓 “项目”, 可以理解为一个网站。

先规划好将项目创建在什么地方, 比如, 笔者放在如下的目录中:

```
Chees-MBP:mysite2 qiwsir$ pwd
/Users/qiwsir/Documents/Codes/DjangoPracticeProject/mysite2
```

在刚刚创建的 `mysite2` 目录中, 创建一个 Django 项目:

```
Chees-MBP:mysite2 qiwsir$ django-admin startproject mysite
```

看一下这个目录, 多了一个 `mysite` 子目录, `mysite` 就是这个项目的名称:

```
Chees-MBP:mysite2 qiwsir$ ls
mysite
```

`mysite` 子目录里面的内容, 就是我们创建的项目内容, 如图 1-3 所示。

```
Chees-MBP:mysite2 qiwsir$ tree
├── mysite
│   ├── manage.py
│   └── mysite
│       ├── __init__.py
│       ├── settings.py
│       ├── urls.py
│       └── wsgi.py
2 directories, 5 files
```

图 1-3 mysite 的目录结构

这是创建项目的一种方法。

可以把刚才建立的项目删除, 即删除 `./mysite2/mysite/` 目录, 然后尝试另外一种创建项目的方法, 请读者注意观察两种方法的命令形式和结果。第二种创建项目的方法如下:

```
Chees-MBP:mysite2 qiwsir$ django-admin startproject mysite .
```

这次在项目名称 `mysite` 后面有一个空格，然后是一个句点（英文半角句号），如此创建项目，旨在将项目内容放到当前目录（`./mysite2/`）中。创建项目之后，仔细观察目录结构，如图 1-4 所示，与图 1-3 所示的目录结构进行对比，从而找出两种方式的差别。

```
Chees-MBP:mysite2 qiwsir$ tree
.
├── manage.py
└── mysite
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
1 directory, 5 files
```

图 1-4 `mysite2` 中的项目目录结构

在所创建的项目目录中，有一个名称很特殊的文件 `__init__.py`，熟悉 Python 的读者一定知道，其所在目录因它的存在而变成了一个 Python 包（package）（关于包、模块的详细内容请参阅《跟老齐学 Python：轻松入门》）。

此外，与 `manage.py` 在同一个级别上有一个目录 `mysite`。如果读者在创建项目时用其他的名称代替 `mysite`，那么就会出现以该名称命名的目录。

至此，已经建立了一个项目，也就意味着已经有一个网站的基本框架了，虽然现在还不能访问什么。

准备就绪，执行下述操作：

```
$ python3 manage.py runserver
```

在本书中，为了明确说明目录或者文件的位置，以“`./`”表示项目根目录，如图 1-4 所示的目录结构，`manage.py` 文件在项目中的位置就是“`./manage.py`”；在子目录 `mysite` 中看到的 `urls.py` 文件，则用“`./mysite/urls.py`”路径表示。

在上述指令中使用了“`python3`”，是因为笔者的开发环境配置原因，读者如果在本地的开发环境中以“`python`”指向了 Python3，也可以使用“`python manage.py runserver`”指令。

在执行上述指令后，如果一切正常，最终会看到如下的提示信息：

```
$ python3 manage.py runserver
```

```
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
```

```
Run 'python manage.py migrate' to apply them.
```

```
June 11, 2018 - 03:25:15
```

```
Django version 2.1.4, using settings 'mysite.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CONTROL-C.
```


由提示信息可知: 通过 `http://127.0.0.1:8000/` 访问所创建的项目; 按 `Ctrl+C` 组合键结束当前服务。

打开浏览器, 在地址栏中输入 `http://127.0.0.1:8000` 或者 `http://localhost:8000`, 就会看到如图 1-5 所示的结果。

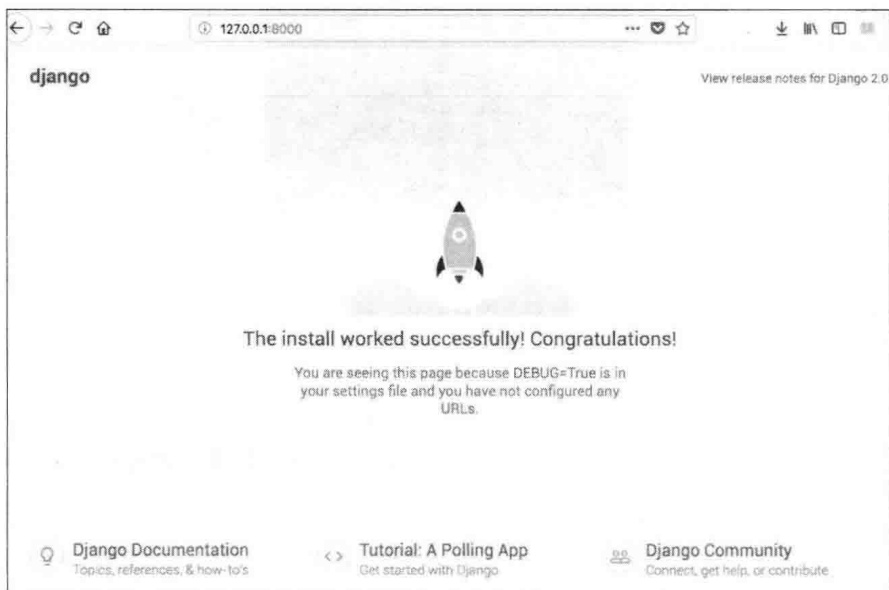


图 1-5 项目默认首页

祝贺! 祝贺! 热烈祝贺!

已经看到了第一个网页, 虽然它很简陋, 但未来它将被不断优化, 成为一个优秀的网站, 读者会用它改变世界——程序员, 天生就被赋予了改变世界的使命。某天某布斯找到你, “我现在只差一个程序员了”。为了改变世界, 继续学习吧!

网站的成长方式就是不断地增加功能。在 Django 中, 人们将完成某个或者某几个功能的集合称为一个“应用”, 所以一个功能比较多的网站常常是由多个“应用”组成的, 后面我们可以把注意力集中在一个个“应用”上。

1.1.4 创建应用

项目已经创建好, 网站也有了, 接下来要实现网站的具体功能。在 Django 中, 把这些具体的功能称为“应用”(application)。

进入刚才创建的项目目录中, 即 `manage.py` 文件所在的目录, 然后按照下面的方式操作:

```
$ ls
db.sqlite3  manage.py  mysite
$ python3 manage.py startapp blog #①
$ ls
blog      db.sqlite3  manage.py  mysite
```

从上述操作中可以看出, 执行了语句①之后, 在目录中多了一个名为 `blog` 的目录。如果观察这时候的 `mysite2` 目录结构, 会看到 `blog` 里面已经有默认的文件和目录了, 如图 1-6 所示。