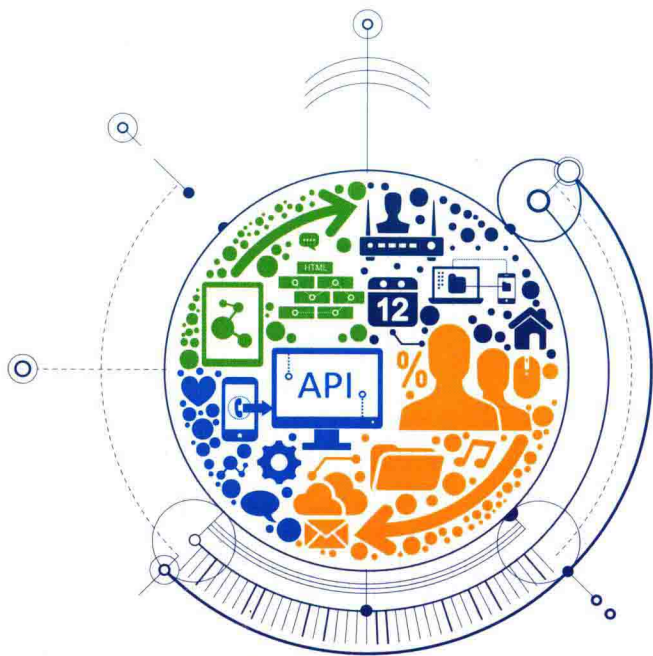




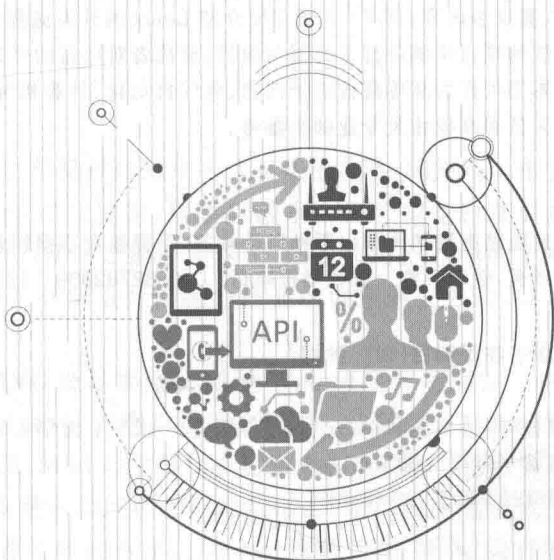
Django

开发宝典

◎ 王友钊 黄 静 编著



清華
科技大講堂
前沿·科技·分享



Django 开发宝典

◎ 王友钊 黄 静 编著

清华大学出版社
北京

内 容 简 介

本书共分9章,第1章介绍HTML、CSS、JavaScript等技术在界面设计方面的应用;第2章主要介绍MySQL的安装和配置及对数据库的操作等内容;第3章对Java语言的内容、功能、特性和对面向对象、多线程及网络编程等内容做了详细的讲解;第4~6章主要介绍Django应用框架和Nginx、uWSGI服务器的安装、测试等内容;第7~8章主要介绍Linux的开发流程及线程、进程、网络通信的内容;第9章对用户认证系统的项目实例进行了部署与设计,使读者对项目的开发流程有基本的了解。

本书可作为高等院校电子信息、电气自动化、计算机等专业的本科生和硕士研究生教材,也可供工程技术人员和高校相关专业师生参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Django 开发宝典/王友钊,黄静编著. —北京:清华大学出版社,2017

(清华科技大讲堂)

ISBN 978-7-302-43696-6

I. ①D… II. ①王… ②黄… 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084734号

责任编辑:刘 星 王冰飞

封面设计:刘 键

责任校对:时翠兰

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印装者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:25.25

字 数:616千字

版 次:2017年1月第1版

印 次:2017年1月第1次印刷

印 数:1~2500

定 价:69.00元

产品编号:066013-01

前言

Web 开发激动人心且富于创造性,但它却是件烦琐而令人生厌的工作。Django 作为一款可使 Web 开发工作愉快并且高效的 Web 开发框架,能够以最小的代价构建和维护高质量的 Web 应用。Django 通过减少重复代码,使用户能够专注于 Web 应用上有趣的关键性东西,提供通用 Web 开发模式的高度抽象与频繁进行的编程作业的快速解决方法,以及为“如何解决问题”提供了清晰明了的约定。

Django 是一款基于 Python 语言及 MVC 设计模式实现的 Web 应用开发框架。MVC 设计模式适用于大型可扩展的 Web 应用开发,它将客户端请求、请求处理、服务器响应划分为模型、视图、控制器 3 个部分。其中,模型(Model)主要负责后台数据库操作;视图负责响应页面的呈现;控制器接收用户请求,根据请求访问模型获取数据,并调用视图显示这些数据。控制器将模型和视图隔离,并成为二者之间的枢纽。Django 对传统的 MVC 设计模式进行了改进,将视图分成 View 模块和 Template 模块两部分,将动态的逻辑处理与静态的页面展现分离开。而 Model 采用了 ORM 技术,将关系型数据库表抽象成面向对象的 Python 类,将表操作转换成类操作,避免了复杂的 SQL 语句编写。

利用 Web 开发框架能够有效缩短研发时间,实现业务模块化开发和敏捷部署。Rod Johnson 和 Juergen Hoeller 等开发的 Spring Framework 是开源 Java EE 全栈应用程序框架,利用控制翻转原则实现配置管理便于应用程序快速组建,对数据库进行一般化抽象使事务划分处理与底层无关;Django 是基于 Python 的 Web 开发框架,基于动态脚本语言的实现方式避免了应用程序像 Java 程序一样庞大臃肿,基于 MTV 模式(Model、Template、View),利用模型对象关系映射、URL 匹配模块、内建模板语言和缓存系统,实现业务模块拆分和快速部署,Django 尝试留下一些方法,让用户根据需要在 Framework 之外开发。

Django 是笔者和团队成员在搭建智慧农业服务平台时触碰到的新思想和新工具。针对智慧农业系统缺乏通用服务器架构、软件重复开发和数据资源浪费的问题,我们团队提出了一种通用性的服务器平台,降低界面呈现、服务流程控制及数据处理的耦合度,利用面向对象思想简化数据库设计的复杂度,提高服务器开发效率,实现各类应用服务的快速部署,其中,应用服务程序基于 Django 框架实现。

本书结合本团队开发经验和相关知识按体系撰写而成,本书包括:HTML、CSS、JavaScript 等技术在界面设计方面的应用;MySQL 的安装和配置及对数据库的操作等内

Foreword

容; Java 语言的内容、功能、特性和面向对象、多线程及网络编程等内容; Django 应用框架和 Nginx、uWSGI 服务器的安装、测试等内容; Linux 的开发流程及线程、进程、网络通信的内容。在本书的最后一章对用户认证系统的项目实例进行了部署与设计,使读者对项目的开发流程有基本的了解。

本书可以作为开发工具和宝典,通过阅读和参照实现,可以完成开发者最初的开发指导,也可以作为平台搭建的实例指导。

作者

2016 年 9 月

目 录

第 1 章 前端技术	1
1.1 初识 HTML	1
1.2 走进 HTML	1
1.3 CSS 基础	8
1.4 CSS 样式	11
1.5 CSS 框模型	18
1.6 CSS 定位	19
1.7 JavaScript 基础	21
1.8 JavaScript HTML DOM	26
1.9 JavaScript 库	31
1.9.1 JavaScript 库简介	31
1.9.2 jQuery	31
第 2 章 MySQL	41
2.1 MySQL 的安装和配置	41
2.2 MySQL 基本操作	43
2.2.1 数据库相关操作	43
2.2.2 表的操作	44
2.2.3 数据的操作	48
2.2.4 数据记录查询	49
2.3 数据的备份与恢复	52
2.4 访问数据库	53
第 3 章 Java 程序开发	56
3.1 Java 简介	56
3.2 Java 多线程编程	57
3.2.1 一个线程的生命周期	58

Contents

3.2.2	创建一个线程	58
3.2.3	线程安全与共享资源	60
3.2.4	死锁	61
3.2.5	线程的调度	62
3.2.6	Java 同步块	72
3.2.7	并发容器	75
3.2.8	线程池的使用	80
3.3	Java 网络编程	84
3.3.1	Java 网络编程基础	84
3.3.2	非阻塞式的 Socket 编程	95
3.3.3	安全网络通信	100
第 4 章	Django 应用框架	106
4.1	Django 概述	106
4.2	安装	107
4.3	视图(View)和统一资源定位符(URL)	113
4.3.1	创建视图	113
4.3.2	创建 URLconf	114
4.3.3	正则表达式	116
4.3.4	Django 请求处理方式	118
4.3.5	关于 Request 与 Response	119
4.3.6	动态视图内容	122
4.3.7	动态 URL	123
4.4	模板(Template)	126
4.4.1	模板系统基本知识	127
4.4.2	如何使用模板系统	128
4.4.3	模板渲染	129
4.4.4	字典和 Context 替换	130
4.4.5	深度变量的查找	131
4.4.6	Context 对象的操作	134
4.4.7	理念与局限	141
4.4.8	在视图中使用模板	142
4.4.9	模板加载	143
4.4.10	locals() 技巧	146
4.4.11	include 模板标签	147
4.4.12	模板继承	148
4.5	模型(Model)	151

4.5.1	在视图中进行数据库查询的基本方法	152
4.5.2	MTV 开发模式	152
4.5.3	创建 APP 应用程序	153
4.5.4	在 Python 代码中定义模型	154
4.5.5	编写模型	155
4.5.6	模型安装	156
4.5.7	基本数据访问	159
4.5.8	Unicode 对象	161
4.5.9	数据过滤	163
4.5.10	获取单个对象	164
4.5.11	数据排序	165
4.5.12	连锁查询	166
4.5.13	更新多个对象	167
4.5.14	删除对象	168
4.6	Django 实例——搭建一个博客	169
4.7	Session	176
4.8	常用服务器命令	180
第 5 章	Nginx 模块开发	182
5.1	Nginx 简介	182
5.2	Nginx 配置	182
5.2.1	安装 Nginx	182
5.2.2	Nginx 命令行控制参数	187
5.2.3	Nginx 配置的基本方法	188
5.2.4	rewrite 重定向	192
5.3	简单的 HTTP 子请求模块开发	195
5.4	简单的 HTTP 过滤模块开发	199
5.5	SSL 模块	206
第 6 章	uWSGI 服务器	210
6.1	uWSGI 概述	210
6.2	uWSGI 安装及运行命令	211
6.3	uWSGI 选项配置	213
第 7 章	嵌入式开发	216
7.1	系统概述	216
7.1.1	嵌入式系统的基本概念	216

7.1.2	嵌入式系统的特点	217
7.1.3	嵌入式系统的发展趋势	218
7.2	嵌入式 Linux 基础	218
7.2.1	Linux 文件系统	218
7.2.2	Linux 目录结构	221
7.2.3	文件类型及文件属性	223
7.2.4	嵌入式 Linux 开发环境构建	225
7.2.5	Minicom 的安装	229
7.3	嵌入式 C 语言开发流程	231
7.3.1	Vim 编辑器	232
7.3.2	GCC 编译器	234
7.3.3	GDB 调试器	237
7.3.4	GDBServer 远程调试	241
7.3.5	Make 工程管理器	242
7.4	文件 I/O	251
7.4.1	文件 I/O 编程基础	251
7.4.2	基本 I/O 操作	253
7.4.3	标准 I/O 操作	256
7.4.4	Linux 串口编程	259
7.4.5	串口使用详解	264
7.4.6	串口编程实例	265
7.4.7	Modbus 通信协议	267
7.4.8	ZigBee 通信协议	269
7.5	Linux 进程	273
7.5.1	进程概述	273
7.5.2	Linux 进程编程	274
7.5.3	Zombie 进程	282
7.5.4	进程间的通信和同步	283
7.5.5	管道通信	284
7.5.6	共享内存通信	292
7.5.7	其他通信方式	295
7.6	线程概述	295
7.6.1	线程的分类和特性	296
7.6.2	线程的实现	296
7.6.3	线程属性	297
7.6.4	线程之间的同步与互斥	299

第 8 章 网络编程	305
8.1 套接字编程简介	305
8.2 套接字选项	307
8.2.1 SOL_SOCKET 协议族选项	308
8.2.2 IPPROTO_IP 选项	312
8.3 基本 TCP 套接字编程	314
8.3.1 socket 概述	314
8.3.2 connect() 函数	315
8.3.3 bind() 函数	315
8.3.4 listen() 函数	316
8.3.5 accept() 函数	317
8.3.6 fork() 与 exec() 函数	318
8.3.7 close() 函数	319
8.3.8 TCP 编程实例	319
8.4 基本 UDP 套接字编程	322
8.4.1 recvfrom() 和 sendto() 函数	323
8.4.2 UDP 的 connect() 函数	324
8.4.3 UDP 程序实例	325
第 9 章 用户认证系统实例	328
9.1 静态资源部署	328
9.1.1 Nginx 配置	328
9.1.2 静态资源	330
9.2 Django 应用处理程序设计	332
9.2.1 项目创建及配置	332
9.2.2 数据库设计	334
9.2.3 应用处理程序设计	335
附录 A CSS 源码	341
附录 B HTML 文件	386
参考文献	394

1.1 初识 HTML

1. HTML 的定义

HTML 是用来描述网页的一种语言。

(1) HTML 是指超文本标记语言(Hyper Text Markup Language)。

(2) HTML 不是一种编程语言,而是一种标记语言(Markup Language)。

(3) 标记语言是一套标记标签(Markup Tag),HTML 使用标记标签来描述网页。

2. HTML 标签

HTML 标记标签通常被称为 HTML 标签(HTML Tag)。

(1) HTML 标签是由尖括号括起来的关键词,如<html>。

(2) HTML 标签通常是成对出现的,如和。

(3) 标签对中的第一个标签是开始标签,第二个标签是结束标签;开始标签和结束标签也被称为开放标签和闭合标签。

3. HTML 元素

HTML 元素是指从开始标签(Start Tag)到结束标签(End Tag)的所有代码,如表 1.1 所示。

表 1.1 HTML 元素

开始标签	元素内容	结束标签
<p>	This is a paragraph	</p>
	This is a link	

1.2 走进 HTML

1. HTML 编辑器

可以使用专业的 HTML 编辑器来编辑 HTML,如 Adobe Dreamweaver、Microsoft

Expression Web、CoffeeCup HTML Editor。

不过,同时推荐使用文本编辑器来学习 HTML,如 Notepad (PC)或 TextEdit (Mac),使用一款简单的文本编辑器是学习 HTML 的好方法。

通过记事本,依照以下 4 个步骤来创建第一张网页。

步骤 1,启动记事本。

步骤 2,用记事本来编辑 HTML。

步骤 3,保存 HTML。

在记事本的“文件”菜单中选择“另存为”选项,然后在弹出的对话框中进行保存。

当保存 HTML 文件时,既可以使用 .htm 扩展名也可以使用 .html 扩展名进行保存。两者没有区别,完全可以根据用户喜好来使用。

步骤 4,在浏览器中运行这个 HTML 文件。

启动浏览器,然后选择“文件”菜单中的“打开文件”命令,或者直接在文件夹中双击 HTML 文件。

2. HTML 标题

HTML 标题(Heading)是通过<h1>~<h6>等标签进行定义的。例如:

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

应该将 h1 用做主标题(最重要的),其次是 h2(次重要的),再次是 h3,以此类推。

3. HTML 段落

HTML 段落是通过<p>标签进行定义的。例如:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

注释:浏览器会自动地在段落的前后添加空行,<p>是块级元素。

4. HTML 链接

HTML 使用超级链接与网络上的另一个文档相连。几乎在任何一个网页中都能找到超级链接,单击超级链接可以从一张页面跳转到另一张页面。

超级链接可以是一个字、一个词或者一组词,也可以是一幅图像,用户可以单击这些超级链接的内容来跳转到新的文档或者当前文档中的某个部分。

通过使用<a>标签可以在 HTML 中创建超级链接,具体有以下两种使用<a>标签的方式。

(1) 通过使用 href 属性创建指向另一个文档的链接。

(2) 通过使用 name 属性创建文档内的标签。

超级链接的 HTML 代码很简单。例如:

```
<a href = "url">Link text</a> //href 属性规定链接的目标
```

开始标签和结束标签之间的文字被作为超级链接来显示。

使用 target 属性可以定义被超级链接的文档要在何处显示。下面的这行会在新窗口打开文档：

```
<a href = "http://www.baidu.com/" target = "_blank">百度一下,你就知道</a>
```

5. HTML 图像

在 HTML 中,图像由标签定义。是空标签,意思是说,它只包含属性,并且没有结束标签。

要在页面上显示图像需要使用源属性(src),src 是指“source”。源属性的值是图像的 URL 地址。

定义图像的语法是：

```
<img src = "url" /> //URL 指存储图像的位置
```

浏览器将图像显示在文档中图像标签出现的地方。如果将图像标签置于两个段落之间,那么浏览器会首先显示第一个段落,然后显示图像,最后显示第二个段落。

6. HTML 表格

表格由<table>标签来定义。每个表格均有若干行(由<tr>标签定义),每行被分割为若干单元格(由<td>标签定义)。字母 td 指表格数据(table data),即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等。

```
<table border = "1">
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
```

在浏览器显示效果如图 1.1 所示。

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

图 1.1 HTML 表格

7. HTML 列表

HTML 支持有序列表、无序列表和自定义列表。

1) 无序列表

无序列表是一个项目的列表,此列表项目使用粗体圆点(典型的小黑圆圈)进行标记。无序列表始于标签,每个列表项始于标签。

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

浏览器显示如下：

- Coffee
- Milk

列表项内部可以使用段落、换行符、图片、链接及其他列表等。

2) 有序列表

同样,有序列表也是一个项目的列表,此列表项目使用数字进行标记。有序列表始于`<ol>`标签,每个列表项始于`<li>`标签。

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

浏览器显示如下：

1. Coffee
2. Milk

列表项内部可以使用段落、换行符、图片、链接及其他列表等。

3) 自定义列表

自定义列表不仅是一个项目的列表,也是项目及其注释的组合。自定义列表以`<dl>`标签开始,每个自定义列表项以`<dt>`标签开始,每个自定义列表项的定义以`<dd>`标签开始。

```
<dl>
<dt>Coffee</dt>
<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

浏览器显示如下：

Coffee
 Black hot drink
Milk
 White cold drink

定义列表的列表项内部可以使用段落、换行符、图片、链接及其他列表等。

8. HTML 表单和输入

1) 表单

HTML 表单用于收集不同类型的用户输入信息。表单是一个包含表单元素的区域,表单元素是允许用户在表单中(如文本域、下拉列表、单选按钮、复选框等)输入信息的元素。表单使用表单标签(<form>)定义。

```
<form>
...
  input 元素
...
</form>
```

2) 输入

多数情况下被用到的表单标签是输入标签(<input>),输入类型是由类型属性(type)定义的。大多数经常被用到的输入类型如下。

(1) 文本域(Text Fields)。当用户要在表单中输入字母、数字等内容时,就会用到文本域。

```
<form>
First name:
<input type = "text" name = "firstname" />
<br />
Last name:
<input type = "text" name = "lastname" />
</form>
```

浏览器显示如下:

```
First name:
Last name:
```

注意: 表单本身并不可见。同时,在大多数浏览器中,文本域的默认宽度是 20 个字符。

(2) 单选按钮(Radio Buttons)。当用户需要从若干给定的选择中选取其一时,就会用到单选按钮。注意,只能从中选取其一。

```
<form>
<input type = "radio" name = "sex" value = "male" /> Male
<br />
<input type = "radio" name = "sex" value = "female" /> Female
</form>
```

浏览器显示如下:

```
• Male
• Female
```


(3) 复选框(Checkboxes)。当用户需要从若干给定的选择中选取一个或若干选项时,就会用到复选框。

```
<form>
<input type="checkbox" name="bike" />
I have a bike
<br />
<input type="checkbox" name="car" />
I have a car
</form>
```

浏览器显示如下:

```
■ I have a bike
■ I have a car
```

3) 表单的动作属性(Action)和确认按钮

当用户单击确认按钮时,表单的内容会被传送到另一个文件。表单的动作属性定义了目的文件的文件名。由动作属性定义的这个文件通常会对接收到的输入数据进行相关的处理。

```
<form name="input" action="html_form_action.asp" method="get">
Username:
<input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

浏览器显示如下:

Username:

假如用户在上面的文本框内输入几个字母,然后单击确认按钮,那么输入数据会传送到"html_form_action.asp"的页面,该页面将显示出输入的结果。

9. HTML 背景

<body>拥有两个配置背景的标签,背景可以是颜色或者图像。

1) 背景颜色(Bgcolor)

背景颜色属性将背景设置为某种颜色,属性值可以是十六进制数、RGB 值或颜色名。

```
<body bgcolor="#000000">
<body bgcolor="rgb(0,0,0)">
<body bgcolor="black">
```

以上的代码均将背景颜色设置为黑色。

2) 背景(Background)

背景属性将背景设置为图像,属性值为图像的 URL。如果图像尺寸小于浏览器窗口,那么图像将在整个浏览器窗口进行复制。

```
<body background = "clouds.gif">  
<body background = "http://www.w3school.com.cn/clouds.gif">
```

URL 可以是相对地址,如第一行代码;也可以是绝对地址,如第二行代码。

提示: <body> 标签中的背景颜色(bgcolor)、背景(background)和文本(text)属性在最新的 HTML 标准(HTML4 和 XHTML)中已被废弃,W3C 在他们的推荐标准中已删除这些属性。因此,应该使用层叠样式表(CSS)来定义 HTML 元素的布局和显示属性。

10. HTML 样式

当浏览器读到一个样式表,它就会按照这个样式表来对文档进行格式化。有以下 3 种方式来插入样式表。

1) 外部样式表

当样式需要被应用到很多页面时,外部样式表将是理想的选择。使用外部样式表,用户就可以通过更改一个文件来改变整个站点的外观。

```
<head>  
<link rel = "stylesheet" type = "text/css" href = "mystyle.css">  
</head>
```

2) 内部样式表

当单个文件需要特别样式时,就可以使用内部样式表。用户可以在 head 部分通过 <style> 标签定义内部样式表。

```
<head>  
<style type = "text/css">  
body {background-color: red}  
p {margin-left: 20px}  
</style>  
</head>
```

3) 内联样式

当特殊的样式需要应用到个别元素时,就可以使用内联样式。使用内联样式的方法是在相关的标签中使用样式属性。样式属性可以包含任何 CSS 属性。以下实例显示出如何改变段落的颜色和左外边距。

```
<p style = "color: red; margin-left: 20px">  
This is a paragraph  
</p>
```