

# Django 开发

---

**Organization:** 千锋教育 Python 教学部 **Date:** 2019-03-10 **Author:** [张旭](#)

## 一、Django 是什么

---



Django 是一个开放源代码的 Web 应用框架, 由 Python 写成。采用了 MVT 的软件设计模式, 即模型 Model, 视图 View 和模板 Template。它最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站的。并于 2005 年 7 月在 BSD 许可证下发布。

这套框架是以比利时的吉普赛爵士吉他手 Django Reinhardt 来命名的。

Django 的主要目标是使得开发复杂的、数据库驱动的网站变得简单。Django 注重组件的重用性和“可插拔性”, 敏捷开发和 DRY 法则 (Don't Repeat Yourself)。在 Django 中 Python 被普遍使用, 甚至包括配置文件和数据模型。

### Django 小历史

Django 是从真实世界的应用中成长起来的, 它是由 堪萨斯 (Kansas) 州 Lawrence 城中的一个网络开发小组编写的。它的最初版本诞生于 2003 年秋天, 那时 Lawrence Journal-World 报纸的程序员 Adrian Holovaty 和 Simon Willison 开始用 Python 来编写程序。

当时他们的 World Online 小组制作并维护当地的几个新闻站点, 并在以新闻界特有的快节奏开发环境中逐渐发展。这些站点包括有 LJWorld.com、Lawrence.com 和 KUsports.com, 记者 (或管理层) 要求增加的特征或整个程序都能在计划时间内快速的被建立, 这些时间通常只有几天或几个小时。因此, Adrian 和 Simon 开发了一种节省时间的网络程序开发框架, 这是在截止时间前能完成程序的唯一途径。

2005 年的夏天, 当这个框架开发完成时, 它已经用来制作了很多个 World Online 的站点。当时 World Online 小组中的 Jacob Kaplan-Moss 决定把这个框架发布为一个开源软件。

从今往后数年, Django 是一个有着数以万计的用户和贡献者, 在世界广泛传播的完善开源项目。原来的 World Online 的两个开发者 (Adrian and Jacob) 仍然掌握着 Django, 但是其发展方向受社区团队的影响更大。

## 二、快速开始

---

### 1. 安装 Django

打开自己的终端, 执行:

```
pip install django
```

等着进度条走完就 OK 啦!

## 2. 项目初始化

继续执行:

```
django-admin startproject demo
```

## 3. 第一个页面

继续:

```
cd demo  
python manage.py runserver
```

好了, 打开浏览器, 输入地址 <http://127.0.0.1:8000> 看一下吧!

**django**

View [release notes](#) for Django 2.1



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



**Django Documentation**

Topics, references, & how-to's

## 三、它是怎么运行的

先来看看我们都执行了写神马。

### 1. `django-admin startproject demo`

这句话用来创建项目。他会自动创建 Django 项目的初始代码, 并自动帮你配置好项目运行所需的变量。

### 2. `cd demo`

进入项目目录。`cd` 命令的含义是 `change directory`

### 3. `python manage.py runserver`

这句的才是真的将代码运行起来

然后再来看看 Django 都初始化出来了些啥？

```
demo
├── demo
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
└── manage.py
```

这些目录和文件的用处是：

1. 外层的 `demo/` 根目录只是你项目的容器, Django 不关心它的名字, 你可以将它重命名为任何你喜欢的名字。
2. `manage.py`: 一个让你用各种方式管理 Django 项目的命令行工具。
3. 内层的 `demo/` 目录包含你的项目, 它是一个纯 Python 包。它的名字就是当你引用它内部任何东西时需要用到的 Python 包名。(比如 `demo.urls`)。
4. `demo/__init__.py`: 一个空文件, 告诉 Python 这个目录应该被认为是一个 Python 包。如果你是 Python 初学者, 阅读官方文档中的 [更多关于包的知识](#)。
5. `demo/settings.py`: Django 项目的配置文件。如果你想知道这个文件是如何工作的, 请查看 [Django settings](#) 了解细节。
6. `demo/urls.py`: Django 项目的 URL 声明, 就像你网站的“目录”。阅读 [URL 调度器 文档](#) 来获取更多关于 URL 的内容。
7. `demo/wsgi.py`: 作为你的项目的运行在 WSGI 兼容的 Web 服务器上的入口。阅读 [如何使用 WSGI 进行部署](#) 了解更多细节。

## 四、创建自己的第一个应用

什么是应用呢？

说白了应用就是一个具有某些功能的程序。比如咱们手机上安装的各种 App，都算是应用。手机上的叫做“移动应用”，网页端的就叫做“Web 应用”。

项目和应用有啥区别？

应用是一个专门做某件事的网络应用程序——比如博客系统，或者公共记录的数据库，或者简单的投票程序。项目则是一个网站使用的配置和应用的集合。项目可以包含很多个应用。应用可以被很多个项目使用。

创建应用：

```
python manage.py startapp joke
```

依旧先来看一下这个应用的内容：

```
joke
├── __init__.py
├── admin.py
├── apps.py
├── migrations
│   └── __init__.py
├── models.py
├── tests.py
└── views.py
```

每个文件的作用：

`__init__.py`: 每个 app 都是一个软件包，所以必须要有一个 `__init__.py` `admin.py`: 用于项目管理的文件 `apps.py`: 用于 app 配置的文件 `migrations`: 用于数据库迁移的文件 `models.py`: 记录数据模型的文件 `tests.py`: 测试代码放在这里 `views.py`: 页面的视图程序

## 五、编写第一个视图

刚创建完的 app 是没有任何视图的，需要我们手动编写用来控制显示的视图程序。

打开 `views.py` 写入以下内容：

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Hello, please always face the world with a smile. ")
```

这是 Django 中最简单的视图。如果想看见效果，我们需要将一个 URL 映射到它——这就是我们需要 URLconf 的原因了。

为了创建 URLconf，请打开在 demo 目录里的 `urls.py` 文件。并编辑为：

```
from django.contrib import admin
from django.urls import include, path

from joke import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('joke/index', views.index),
]
```

代码改为这样后，服务器会自动重启。当一个新的请求访问 `http://127.0.0.1:8000/joke/index` 时，Django 的 URL 映射程序就会为用户的请求指向到我们开发好的 `index` 函数，并交给 `index` 函数处理。

到此位置，咱们的第一个页面就开始工作了！