

### ###课程内容:

1. 了解Django基础知识
2. 创建第一个Django项目

## # Django @ 5.15

### ## 预备知识

1. `virtualenv`
2. MVT 设计模式和 MVC 设计模式
- //3. RESTful services and RESTful URL

### ## Django

### ### 基础知识

1. Why Django
2. 安装 Django
3. 从命令行出发：创建一个 Django 项目
  
4. Django 项目的文件结构：Project & App
5. Django 是如何工作的：models.py/views.py/templates 与 MVT 的对应关系
6. views.py 处理访问请求：urls.py 与 views.py 的对应关系
  
7. 无需了解 SQL 语句：Django ORM 对 SQL 封装的便利性
8. 首次一键运行 Django

#### **Get Your Student Pack**

Free access to desktop products including IntelliJ IDEA Ultimate, ReSharper Ultimate and other IDEs

### ### 创建一个 Django App: 以工程视角切入

[Official Documentation: Writing your first Django app](<https://docs.djangoproject.com/en/2.2/intro/tutorial01/>)

1. 使用 Django 创建一个用于投票系统的项目: 了解 Django 更多的项目目录结构、文件和作用
2. 在 `views.py` 中书写第一个请求处理函数: 了解 `django.http` 包中重要的 HTTP 响应类和方法
3. 在 `urls.py` 中书写第一个 URL 映射: 了解 `urls.py` 的结构和 Django 如何将 URL 映射到函数
4. 使用 Django ORM 轻松建立第一个数据库和表: 掌握 Django ORM 的语法、迁移数据库和同步数据表的命令

5. 创建一个新问题：使用 shell 了解 Django database API
6. 查询存储过的具体问题：向 URL 传递参数
7. 载入前端模板并填充上下文：使用 `render()` 方法将数据传至 Templates 层
8. 在前端显示查询的问卷问题和选项：掌握简单的 Django Templates 语法，包括访问变量、for 循环、if 分支语句
9. 为 URL 添加别名以去除硬编码，方便引用：使用 `path()` 的 `name` 参数，并在 Django Templates 中引用
10. 编写真实表单和投票功能函数：在 views 中处理 GET 或 POST 请求

建立项目环境

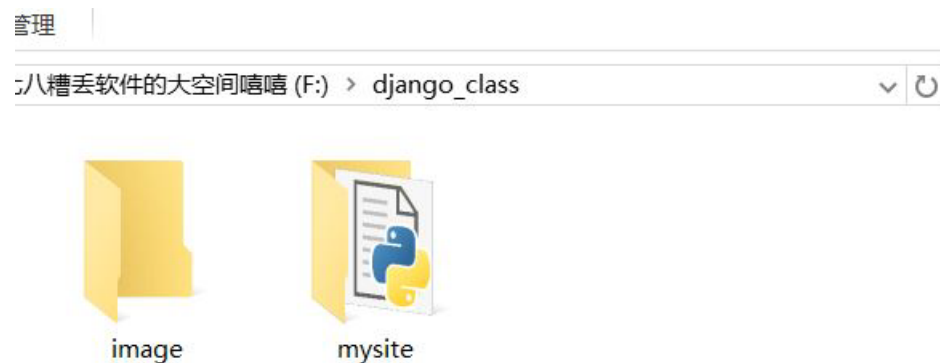
## 2. 安装 Django

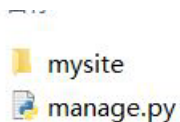
### 3. 从命令行出发：创建一个 Django 项目

```
F:\>cd django_class  
F:\django_class>django-admin startproject mysite  
F:\django_class>
```

django-admin是Django用于管理任务的命令行实用程序。

进入文件夹，新建mysite项目目录

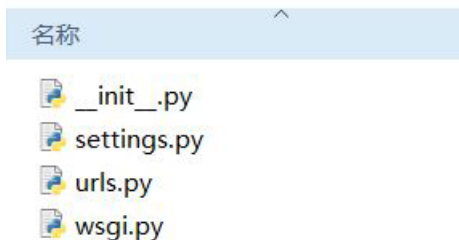




**外部mysite/根目录：**项目的容器。它的名字对Django来说无关紧要；你可以将它重命名为你喜欢的任何东西。

**manage.py：**一个命令行实用程序，允许您以各种方式与此Django项目进行交互。

详情见：<https://docs.djangoproject.com/en/2.2/ref/django-admin/>



**内部mysite/目录：**项目的实际Python包。它的名称是您需要用来导入其中任何内容的Python包名称（例如mysite.urls）。

**mysite/\_\_init\_\_.py：**一个空文件，告诉Python该目录应该被视为Python包。

**mysite/settings.py：**此Django项目的设置/配置。

**mysite/urls.py：**这个Django项目的URL声明；您的Django支持的站点的“目录”。URL：统一资源定位符。

**mysite/wsgi.py：**与WSGI兼容的Web服务器的入口点，用于为您的项目提供服务。



验证Django项目是否有效:

runserver: 在本地计算机上启动轻量级开发Web服务器。  
默认情况下, 服务器在IP地址的端口8000上运行127.0.0.1。

```
F:\django_class\mysite>py manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 15, 2019 - 00:12:10
Django version 2.1.5, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



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.

```
F:\django_class\mysite>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK

F:\django_class\mysite>
```

创建民意调查应用

## 项目与应用

项目和应用程序之间有什么区别？

项目是特定网站的配置和应用程序的集合。

应用程序是执行某些操作的**Web**应用程序

- 例如，**Weblog**系统，公共记录数据库或简单的轮询应用程序。

项目可以包含多个应用程序。一个应用程序可以在多个项目中。

```
F:\django_class\mysite>py manage.py startapp polls
```

## 创建polls应用



```
编辑(E) 选择(S) 查看(V) 转到(G) 调试(D) 任务(I) 帮助(H)
web_django_handbook.md x urls.py x
1 |from django.urls import path
2
3 |from . import views
4
5 urlpatterns = [
6 |    path('', views.index, name='index'),
7 |    ]
```

要调用视图，我们需要将其映射到URL

```
15 """
16 from django.contrib import admin
17 from django.urls import include, path
18
19 urlpatterns = [
20     path('polls/', include('polls.urls')),
21     path('admin/', admin.site.urls),
22 ]
23
```

下一步是将根URLconf指向polls.urls模块。

```
f:\django_class\mysite>py manage.py runserver
Performing system checks...
```

```
[15/May/2019 12:06:01] "GET /favicon.ico HTTP/1.1" 404 2078
Not Found: /
[15/May/2019 12:06:03] "GET / HTTP/1.1" 404 2027
[15/May/2019 12:06:09] "GET /polls HTTP/1.1" 301 0
[15/May/2019 12:06:09] "GET /polls/ HTTP/1.1" 200 40
[15/May/2019 12:06:27] "GET /polls/ HTTP/1.1" 200 40
Not Found: /favicon.ico
[15/May/2019 12:06:27] "GET /favicon.ico HTTP/1.1" 404 2078
```



Hello, world. You're at the polls index.



5. 创建一个新问题：使用 shell 了解 Django database API
6. 查询存储过的具体问题：向 URL 传递参数
7. 载入前端模板并填充上下文：使用 `render()` 方法将数据传至 Templates 层
8. 在前端显示查询的问卷问题和选项：掌握简单的 Django Templates 语法，包括访问变量、for 循环、if 分支语句
9. 为 URL 添加别名以去除硬编码，方便引用：使用 `path()` 的 `name` 参数，并在 Django Templates 中引用
10. 编写真实表单和投票功能函数：在 views 中处理 GET 或 POST 请求