

# Django介绍

# 本章大纲

- 开发接口环境搭建
- Django简介
- 第一个Demo
- Template

# 安装Django

- 在线安装

`pip install Django`

- 离线安装

方法一：单击 .tar.gz 结尾的文件，并且对文件进行解压，进入到解压目录中，通过python命令进行安装。

进入到目录名中 > `python setup.py install`

方法二：.whl 文件本质上面是一个 .zip 包格式。

进入到目录名中 > `pip install python-3.14.1-py2.py3-none-any.whl`

# 本章大纲

- 开发接口环境搭建
- Django简介
- 第一个Demo
- Template

# Django简介

- 什么是Django?

Django是一个基于python的高级Web开发框架  
能够让开发人员进行高效且快速的开发  
高度集成，而且开源

# 项目目录介绍

- `manage.py`

与项目进行交互的命令行工具集的入口

项目管理器

执行`python manage.py`

# 项目目录介绍

- DjangoDemo 目录

项目的一个容器

包含项目最基本的一些配置

目录名称不建议修改

# 项目目录介绍

- `wsgi.py`  
python web server gateway interface  
python 应用与web服务器之间的接口
- `urls.py`  
URL配置文件  
Django项目所有的地址（页面）需要自己去配置URL
- `settings.py`  
项目的总配置文件，包含数据库，Web应用，时间等各种配置
- `__init__.py`  
声明模块的文件



# 启动Django服务器

- `python manage.py runserver`
- `python manage.py runserver 9000`

```
D:\pythonCode\DjangoDemo1105>python manage.py runserver 127.0.0.1:9000
```

**It worked!**

Congratulations on your first Django-powered page.

Next, start your first app by running `python manage.py startapp [app_label]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

# 创建应用

- 创建步骤

- 1、命令行输入 `python manage.py startapp api`
- 2、添加应用到 `settings.py` 中的 `INSTALLED_APPS` 中

# 本章大纲

- 开发接口环境搭建
- Django简介
- 第一个Demo
- Template

# 应用目录介绍

- migrations

数据移植模块，Django 自动生成的

- admin.py

该应用的后台管理系统配置

- apps.py

该应用的一些配置

- models.py

数据模块，使用ORM框架

创建应用程序数据表模型（对应数据库的相关操作）

# 应用目录介绍

- tests.py

自动化测试模块

- views.py

执行响应的代码所在模块

项目大部分代码在这里编写

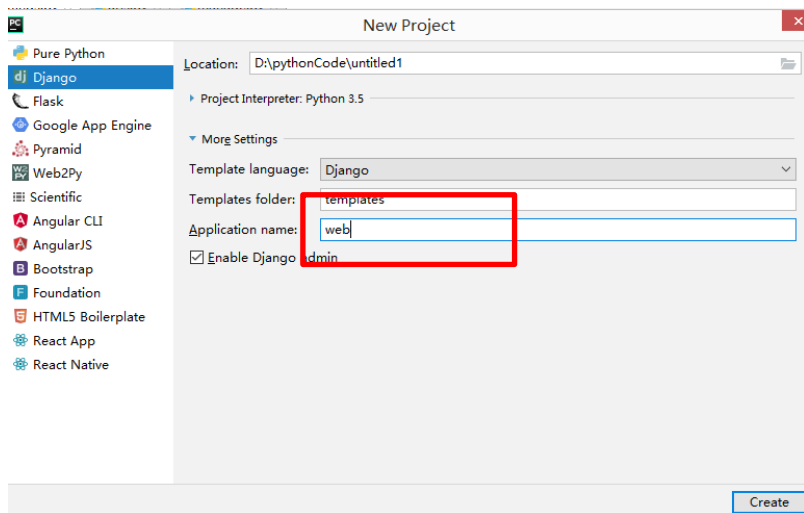
# Python的正则表达式

- `urls.py`包含了URL模式

<code>r '^admin/\$'</code>	
匹配符	含义
<code>r</code>	字符串前面加 “ <code>r</code> ” 是为了防止字符串中出现类似 “ <code>\t</code> ” 字符时被转义
<code>^</code>	匹配字符串开头；在多行模式中匹配每一行的开头 <code>^abc abc</code>
<code>\$</code>	匹配字符串末尾；在多行模式中匹配每一行末尾。 <code>abc\$ abc</code>

# Django第一个Demo

## 1、有两种方式创建APP



```
D:\pythonCode\DjangoDemo1105>python manage.py startapp api
```

# django第一个Demo

## 2、在settings.py中增加'api'



```
settings.py x  urls.py x  manage.py x
.pyt
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'api'
41
42 ]
43
```



# Django第一个Demo

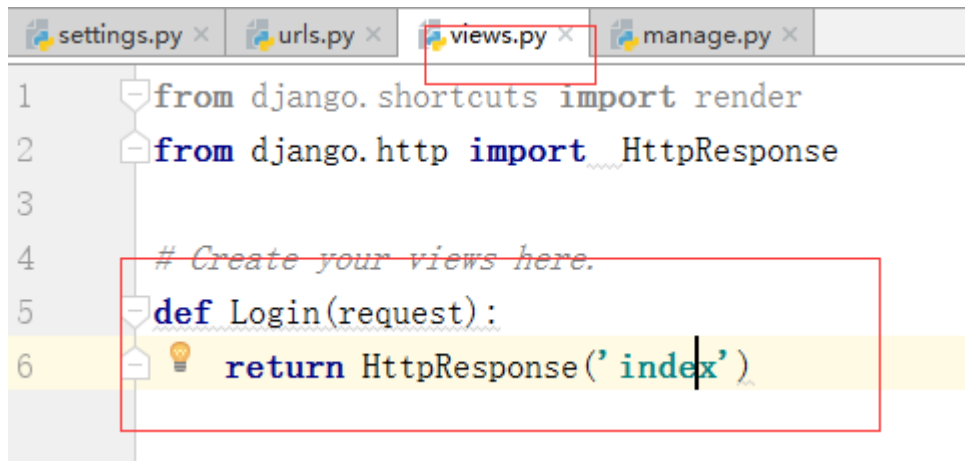
- 3、修改urls.py

```
from django.conf.urls import url
from django.contrib import admin
from api.views import Login

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^login/', Login)
]
```

# Django第一个Demo

## 4、在views.py中实现方法Login



```
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3
4 # Create your views here.
5 def Login(request):
6     return HttpResponseRedirect('index')
```

# Django第一个Demo

- 应用下views.py

每个响应对应一个函数，函数必须返回一个响应  
函数必须存在一个参数，一般为request

- 根目录下urls.py

每个URL都以url形式存在

url函数放在urlpatterns列表中

url函数三个参数：URL正则，对应方法，名称

# 本章大纲

- 开发接口环境搭建
- Django简介
- 第一个Demo
- Template

# URL的完善

1、在根urls中引入include , 并重新配置url

```
url(r'^api/', include("api.urls")),
```

2、APP 目录下创建urls.py文件 , 格式与根urls.api相同

```
from django.conf.urls import url
```

```
from . import views
```

```
urlpatterns = [
```

```
    url(r'^login/', views.Login),
```

```
]
```

注意：根urls.py针对APP配置的url名称是APP配置的是总路径

# templates

- 什么是templates文件?

HTML文件

使用了Django模板语言 ( Django Template Language ,  
DTL )

# templates

- 步骤:

- 1、在templates下创建html文件

- 2、在views.py中返回render

```
def First(request):
```

```
    return render(request,"index.html")
```

```
或者return render(request,"Login.html",{"name":"tom"})
```

页面中获取参数{{name}}

# templates

- HttpResponse 类存在django.http.HttpResponse 中，以字符串的形式传递给前端页面数据。
- render 函数，第一个参数是请求对象的，第二个参数返回一个index.html 页面。



# django处理流程

