社交网络技术与应用

实验一：
如何新建一个Python工程

实验内容

- ❑新建一个Python工程
- ❑数据存储与展示

准备工作

- **Python 3.x**
- **第三方库Django**
  - **Django是一个开放源代码的Web应用框架，由Python写成**
- **编辑器**
  - **Sublime Text**
  - **Jupyter**
  - **Pycharm**
  - **…**

Python安装

Python安装

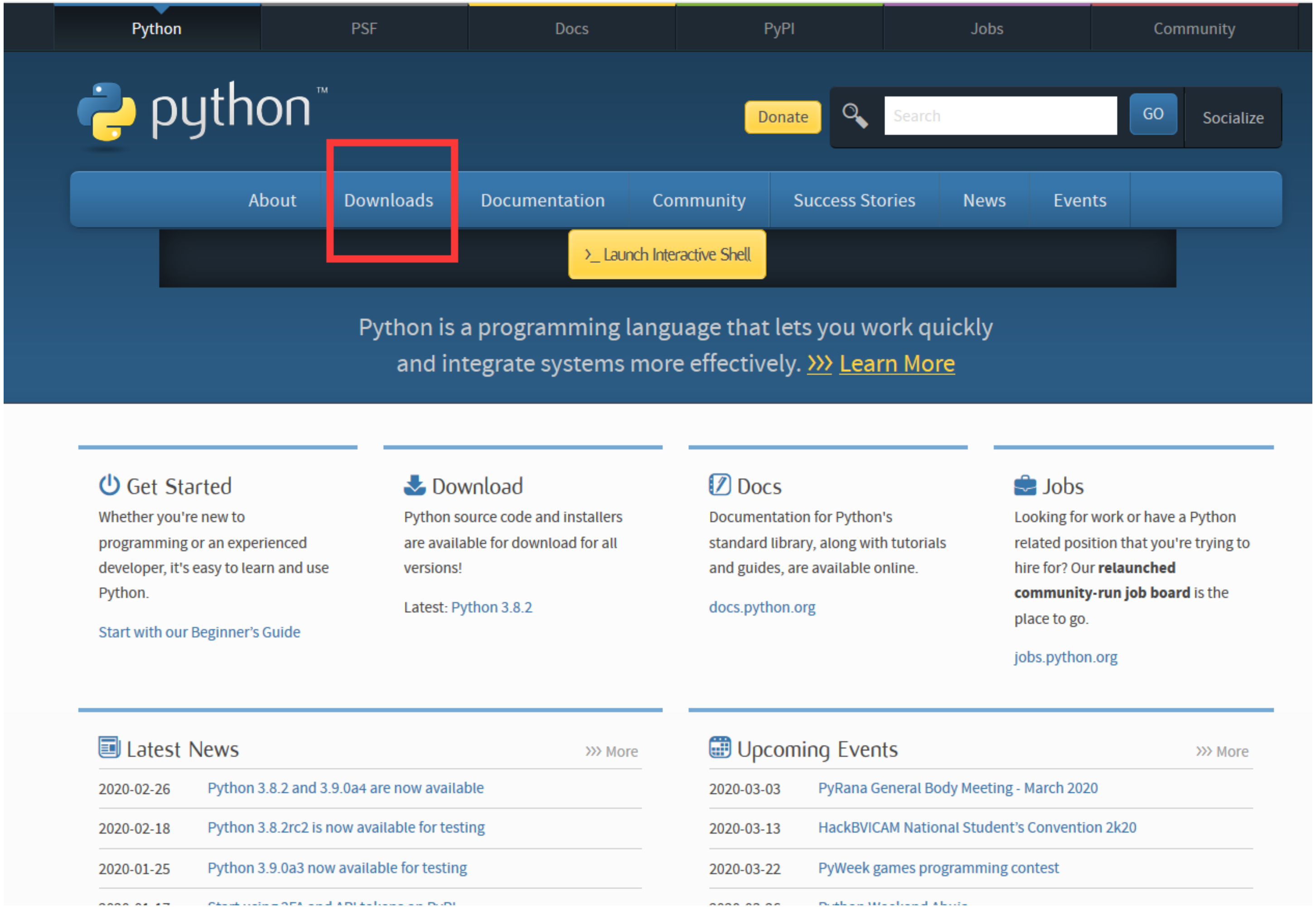
前言

这里主要介绍python3版本的安装

下载安装包

Python官网：python.org ，下载官方安装包（这里选择python3的最新版本）

Python安装



Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Start with our Beginner's Guide

Download

Python source code and installers are available for download for all versions!

Latest: Python 3.8.2

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

docs.python.org

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

jobs.python.org

Latest News

>>> More

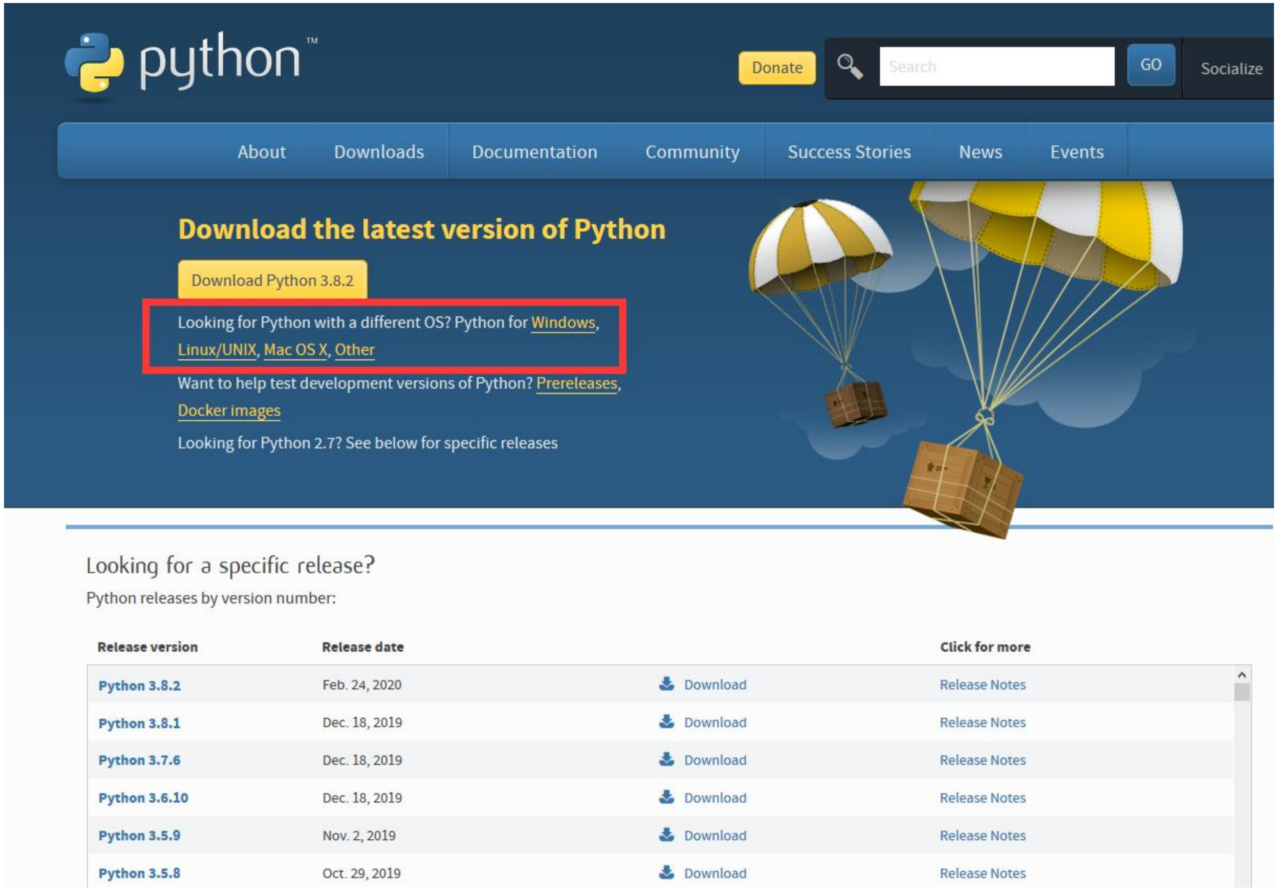
2020-02-26	Python 3.8.2 and 3.9.0a4 are now available
2020-02-18	Python 3.8.2rc2 is now available for testing
2020-01-25	Python 3.9.0a3 now available for testing

Upcoming Events

>>> More

2020-03-03	PyRana General Body Meeting - March 2020
2020-03-13	HackBVICAM National Student's Convention 2k20
2020-03-22	PyWeek games programming contest

Python安装



Download the latest version of Python

Download Python 3.8.2

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

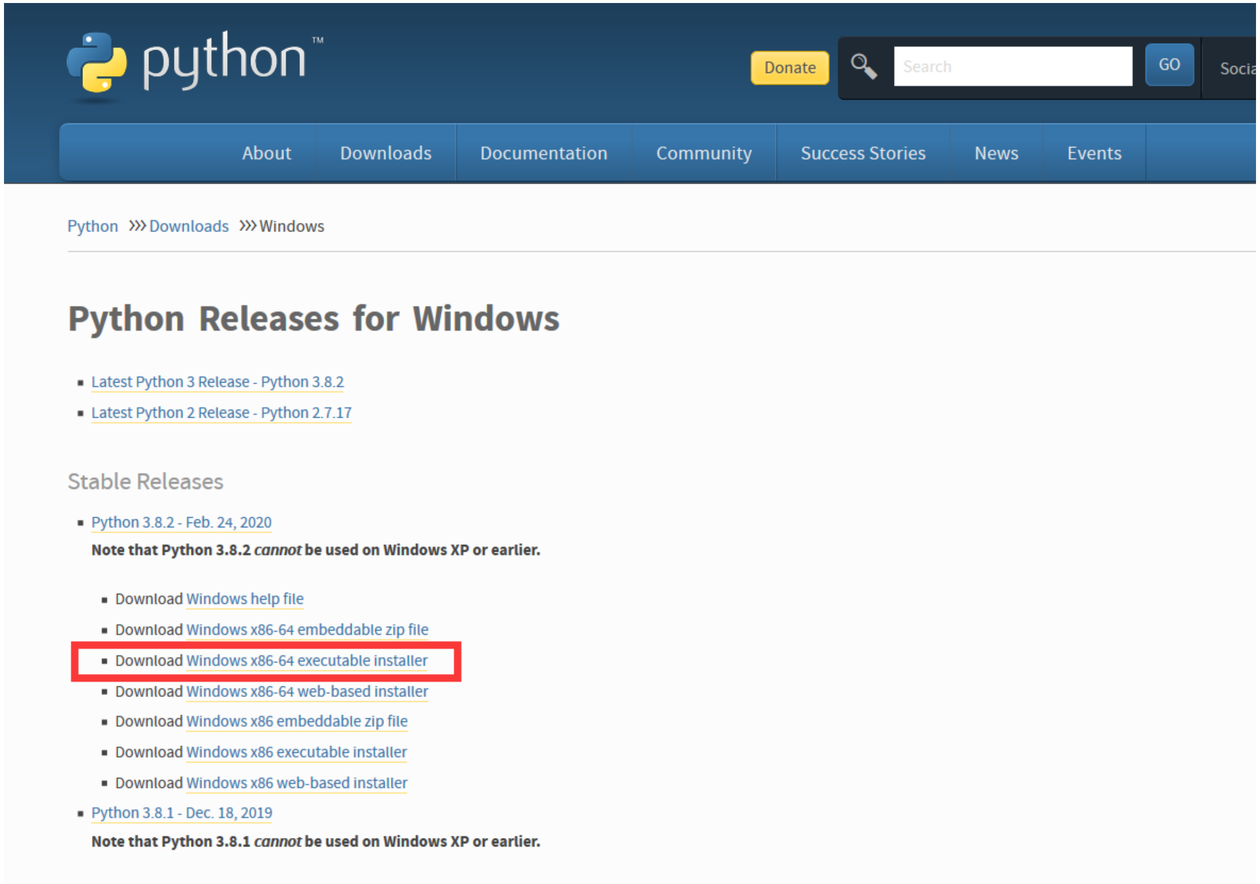
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.8.2	Feb. 24, 2020	Download	Release Notes
Python 3.8.1	Dec. 18, 2019	Download	Release Notes
Python 3.7.6	Dec. 18, 2019	Download	Release Notes
Python 3.6.10	Dec. 18, 2019	Download	Release Notes
Python 3.5.9	Nov. 2, 2019	Download	Release Notes
Python 3.5.8	Oct. 29, 2019	Download	Release Notes

根据自己电脑的系统选择相应的版本进行下载

Python安装



Python ›› Downloads ›› Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.8.2
- Latest Python 2 Release - Python 2.7.17

Stable Releases

- Python 3.8.2 - Feb. 24, 2020

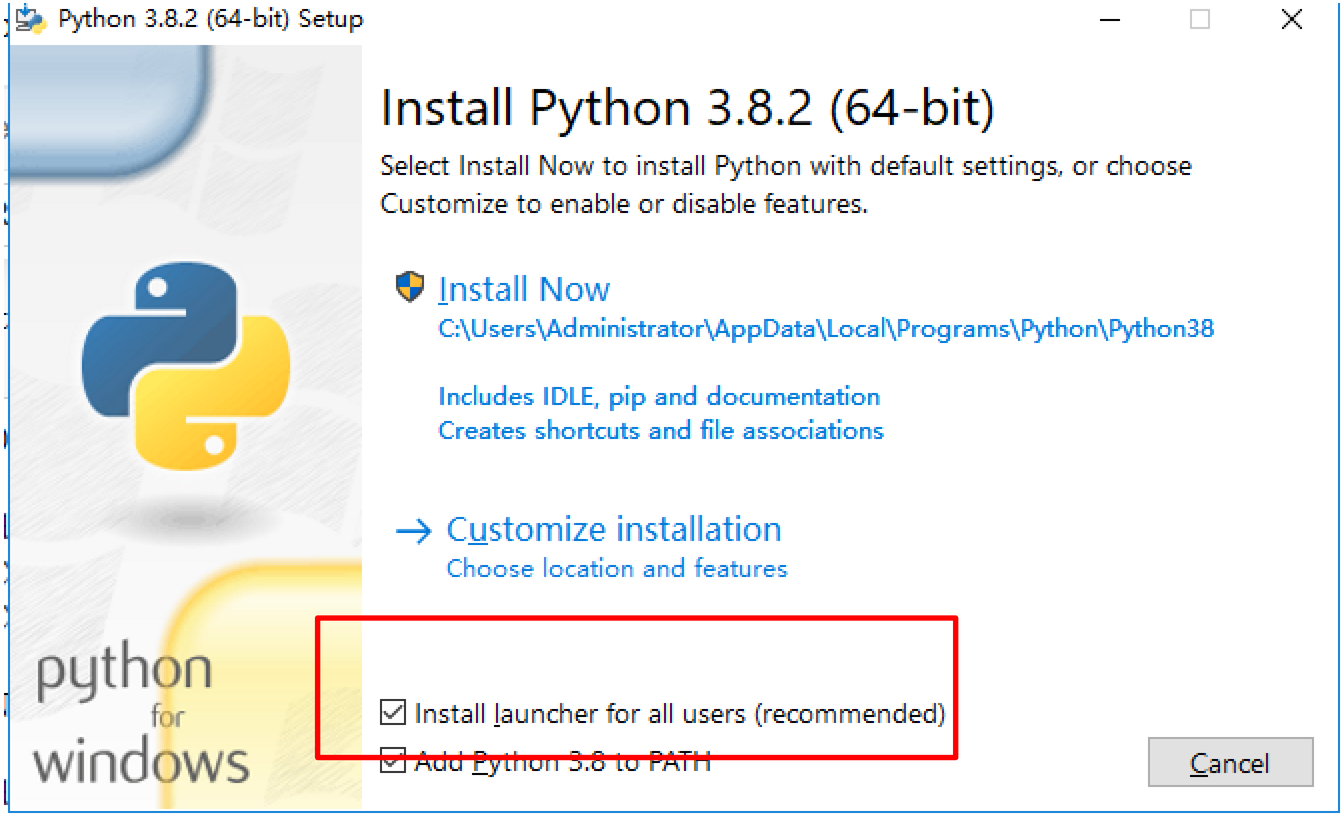
  Note that Python 3.8.2 *cannot* be used on Windows XP or earlier.

  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer
- Python 3.8.1 - Dec. 18, 2019

  Note that Python 3.8.1 *cannot* be used on Windows XP or earlier.

选择Windows x86-64 executable installer版本下载

Python安装



Python 3.8.2 (64-bit) Setup

Install Python 3.8.2 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Administrator\AppData\Local\Programs\Python\Python38

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

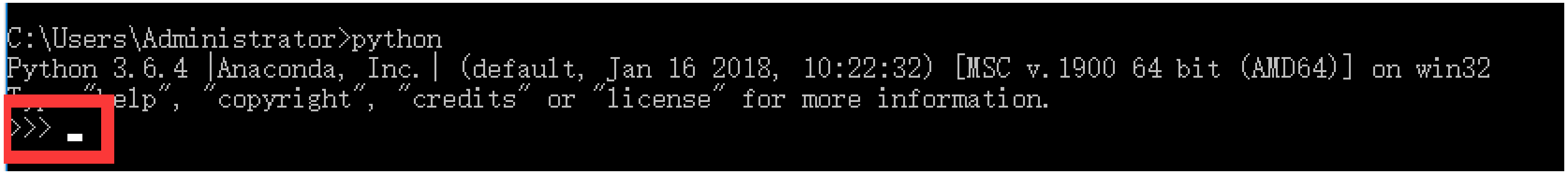
☑ Add Python 3.8 to PATH

其中**Add python 3.xx to PATH一定记得勾选上**，可以选择默认安装也可以进行自定义安装

Python安装

测试

安装成功后，打开命令行，输入python，若出现了如图所示的图标则表明安装成功



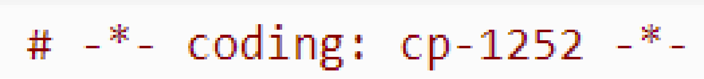
```
C:\Users\Administrator>python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Python3基础语法

Python 3基础语法

❑编码

❖ 默认情况下，Python 3源码文件以UTF-8编码，所有字符串都是unicode字符串，可以为源码文件指定不同的编码：



```
# -*- coding: cp-1252 -*-
```

❑标识符

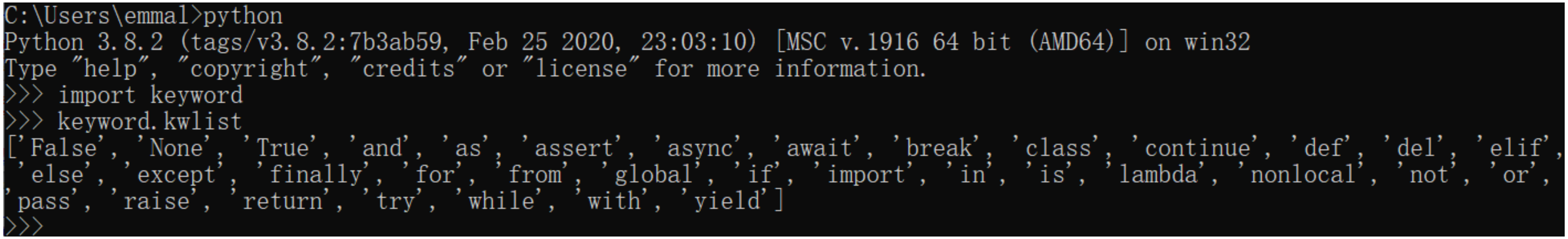
❖ 第一个字符必须是字母或下划线_

❖ 标识符的其他部分由字母、数字和下划线组成，标识符对大小写敏感

❖ 在 Python 3中，可以用中文作为变量名

Python保留字

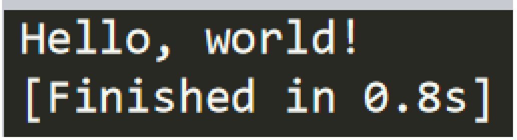
- ❑ 保留字即关键字，不能用作任何标识符名称
- ❑ Python的标准库提供了一个keyword模块，可以输出当前版本的所有关键字



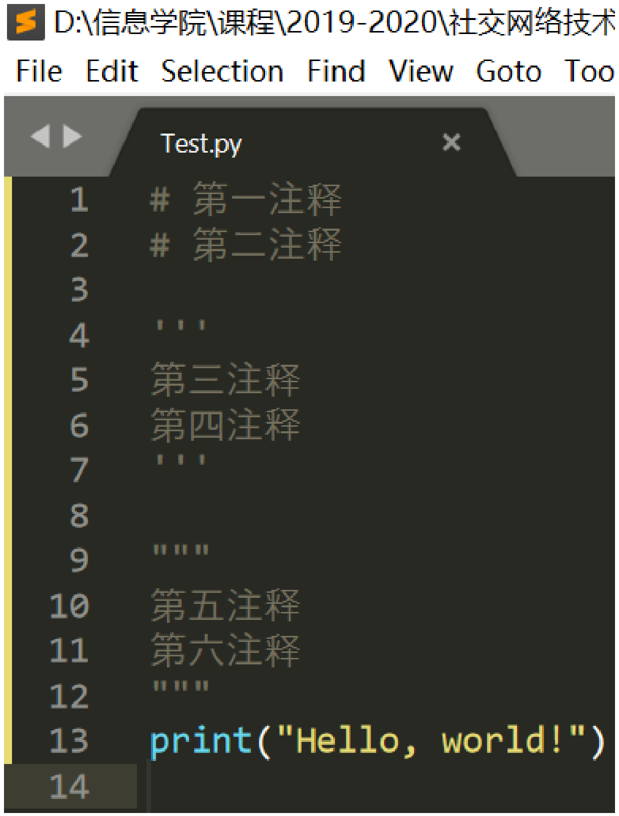
```
C:\Users\emmal>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif',
'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
>>>
```

注释

- ❑Python中单行注释以#开头
- ❑多行注释可以用多个#号，还有 ''' 和 """
- ❑执行右边代码，输出：



```
Hello, world!
[Finished in 0.8s]
```



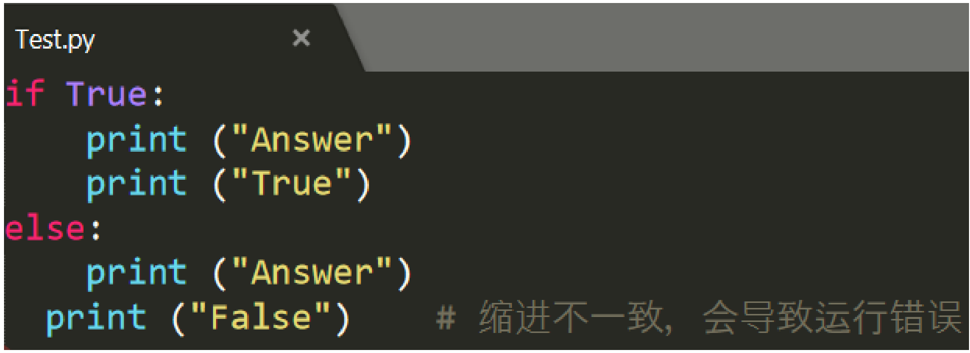
D:\信息学院\课程\2019-2020\社交网络技术
File Edit Selection Find View Goto Too
Test.py

```python
# 第一注释
# 第二注释

'''
第三注释
第四注释
'''

"""
第五注释
第六注释
"""
print("Hello, world!")

```

行与缩进

- **Python最具特色的就是使用缩进来表示代码块，不需要使用大括号{}**

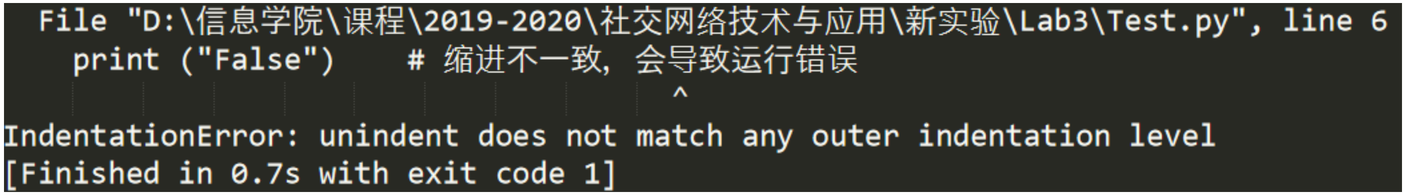
- **缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数**



Test.py

```python
if True:
    print ("Answer")
    print ("True")
else:
    print ("Answer")
  print ("False")    # 缩进不一致，会导致运行错误
```

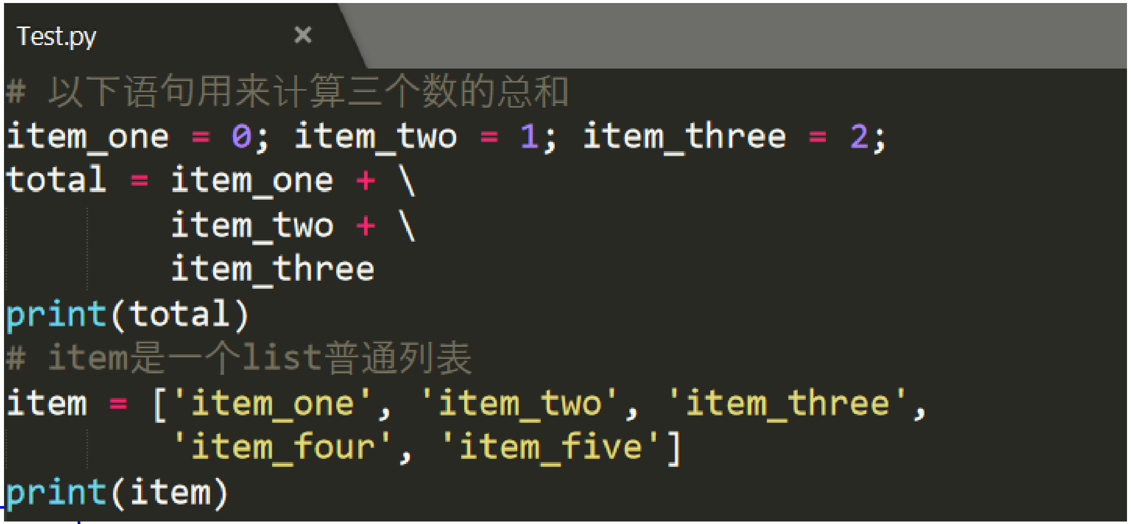
执行后出错



```
  File "D:\信息学院\课程\2019-2020\社交网络技术与应用\新实验\Lab3\Test.py", line 6
    print ("False")    # 缩进不一致，会导致运行错误
                                        ^
IndentationError: unindent does not match any outer indentation level
[Finished in 0.7s with exit code 1]
```

多行语句

- Python通常是一行写完一条语句，但如果语句很长，我们可以使用反斜杠(\)来实现多行语句
- 在[]，{}，或()中的多行语句，不需要使用反斜杠(\)
- 可以同一行显示多条语句，可以用分号 ; 分开

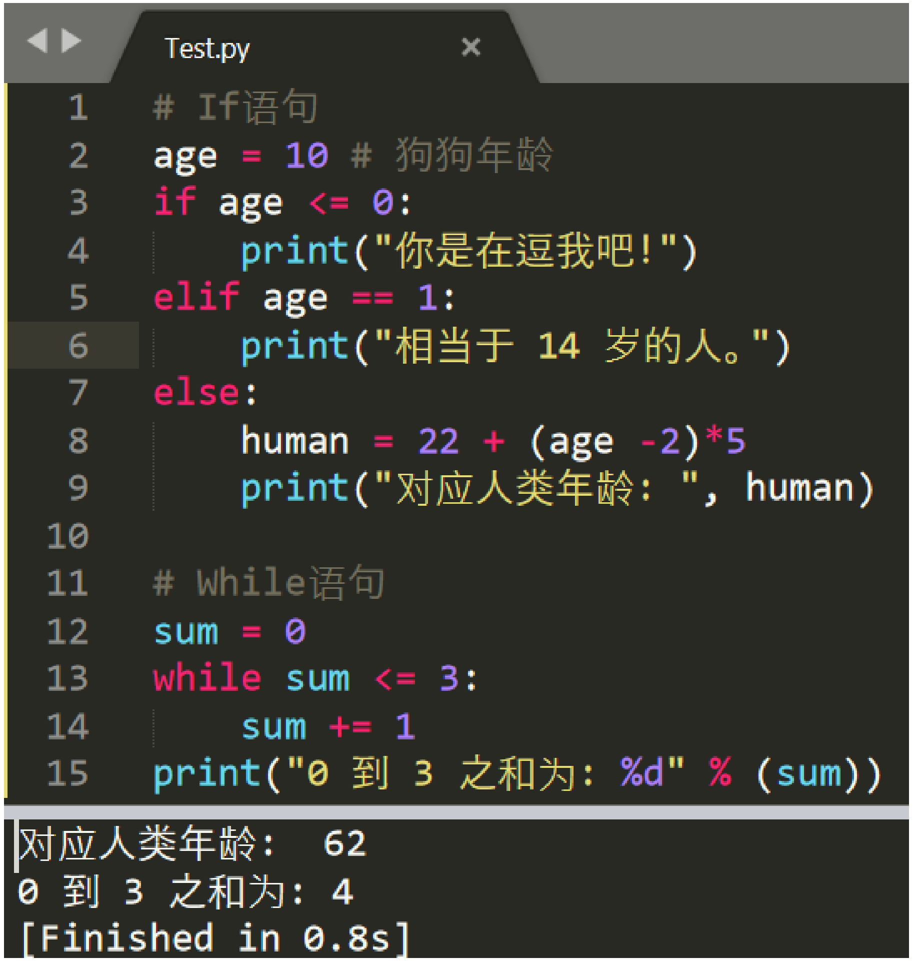


```python
# 以下语句用来计算三个数的总和
item_one = 0; item_two = 1; item_three = 2;
total = item_one + \
        item_two + \
        item_three
print(total)
# item是一个list普通列表
item = ['item_one', 'item_two', 'item_three',
        'item_four', 'item_five']
print(item)
```

代码组

❑ 缩进相同的一组语句构成一个代码块，我们称之代码组

❑ **if、while、def和class** 这样的复合语句，首行以关键字开始，以冒号(:)结束，该行之后的一行或多行代码构成代码组

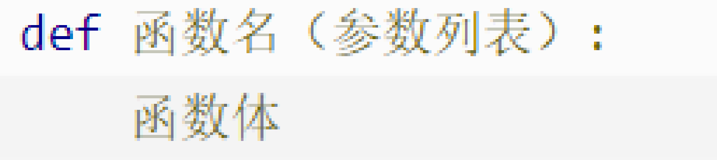


```
# If语句
age = 10 # 狗狗年龄
if age <= 0:
    print("你是在逗我吧!")
elif age == 1:
    print("相当于 14 岁的人。")
else:
    human = 22 + (age -2)*5
    print("对应人类年龄： ", human)

# While语句
sum = 0
while sum <= 3:
    sum += 1
print("0 到 3 之和为： %d" % (sum))
```

```
对应人类年龄：  62
0 到 3 之和为： 4
[Finished in 0.8s]
```

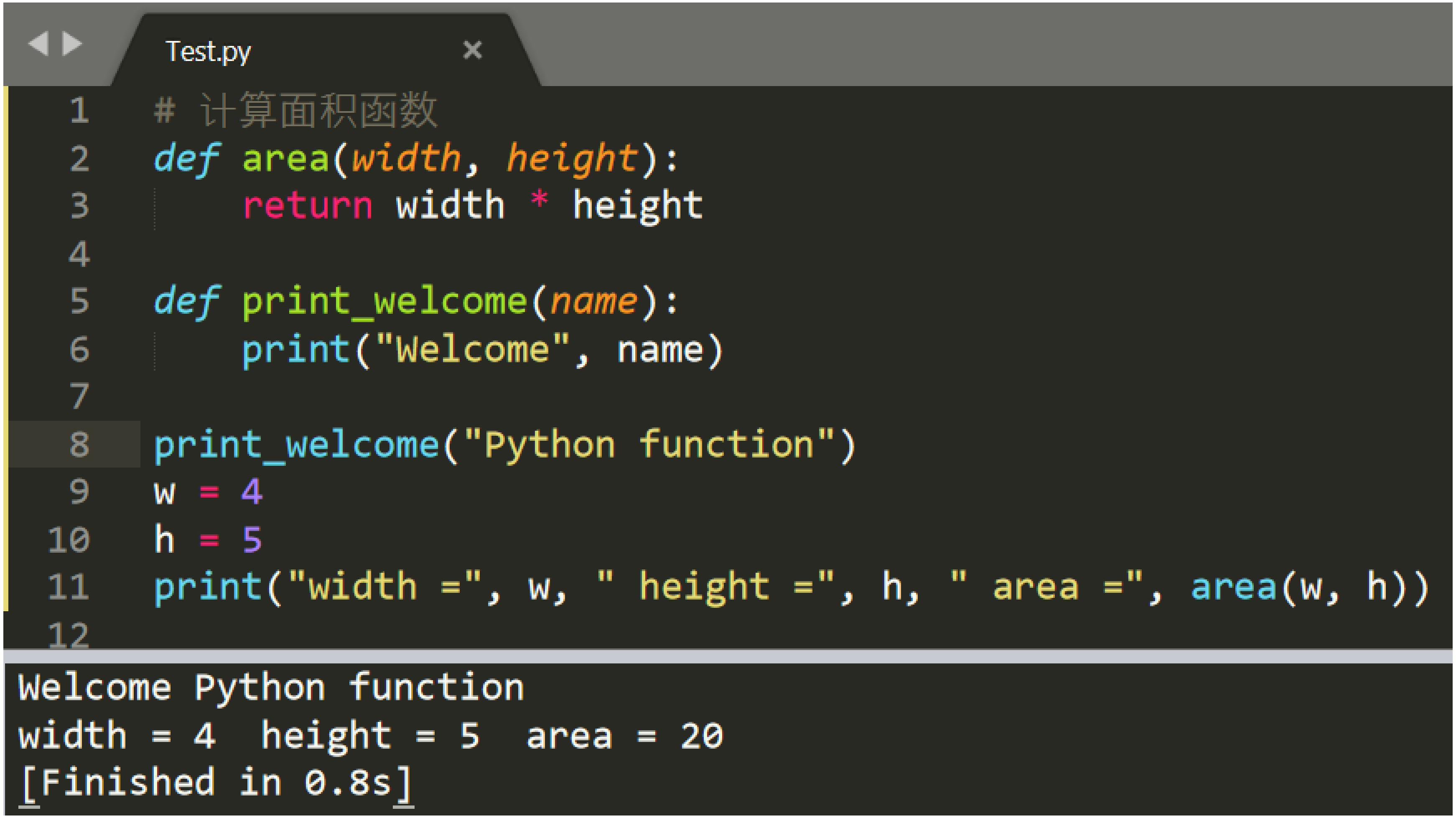
函数

- 函数代码块以def关键词开头，后接函数标识符名称和圆括号()
- 任何传入参数和自变量必须放在圆括号中间，圆括号之间可以用于定义参数
- 函数内容以冒号起始，并且缩进
- return [表达式] 结束函数，选择性地返回一个值给调用方。不带表达式的return相当于返回None



```
def 函数名（参数列表）：
    函数体
```

函数示例



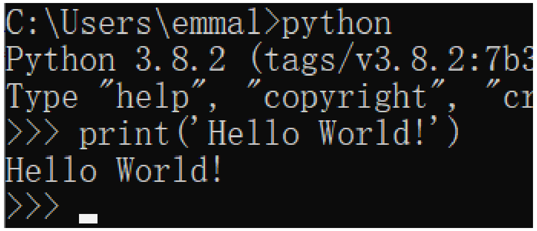
```python
# 计算面积函数
def area(width, height):
    return width * height

def print_welcome(name):
    print("Welcome", name)

print_welcome("Python function")
w = 4
h = 5
print("width =", w, " height =", h, " area =", area(w, h))
```

```
Welcome Python function
width = 4  height = 5  area = 20
[Finished in 0.8s]
```

运行Python

❑ 交互式解释器

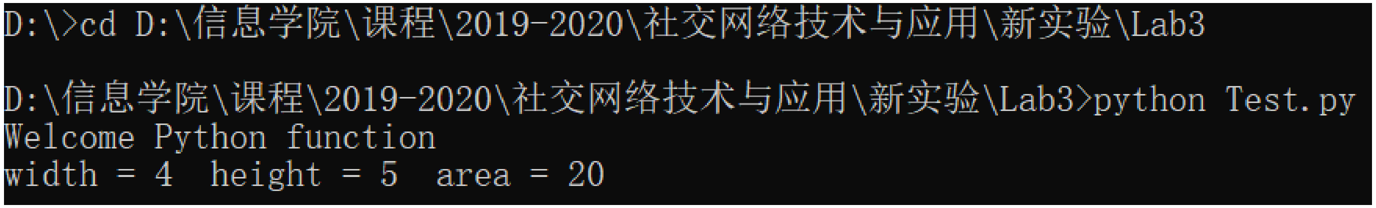
- ❖ 可以通过命令行窗口进入python并在交互式解释器中开始编写Python代码



```
C:\Users\emmal>python
Python 3.8.2 (tags/v3.8.2:7b3
Type "help", "copyright", "cr
>>> print('Hello World!')
Hello World!
>>> _
```

❑ 命令行脚本

- ❖ 在你的应用程序中通过引入解释器可以在命令行中执行Python脚本



```
D:\>cd D:\信息学院\课程\2019-2020\社交网络技术与应用\新实验\Lab3

D:\信息学院\课程\2019-2020\社交网络技术与应用\新实验\Lab3>python Test.py
Welcome Python function
width = 4  height = 5  area = 20
```

❑ 集成开发环境：Sublime text

- ❖ **Build：Ctrl+B**

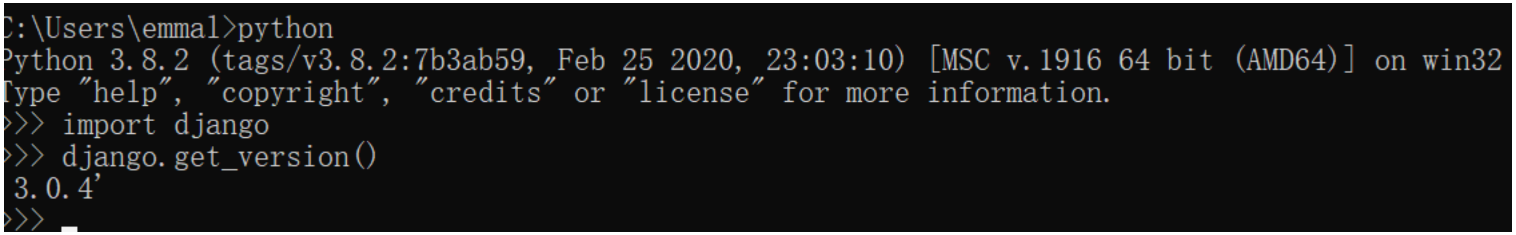
Django安装

安装第三方模块Django

- ❑pip install django==3.0.4
- ❑pip install django-import-export==2.5.0
- ❑也可以使用国内镜像安装

  pip install -i https://pypi.douban.com/simple django

- ❑测试是否安装成功
  - ❖ 命令行环境测试



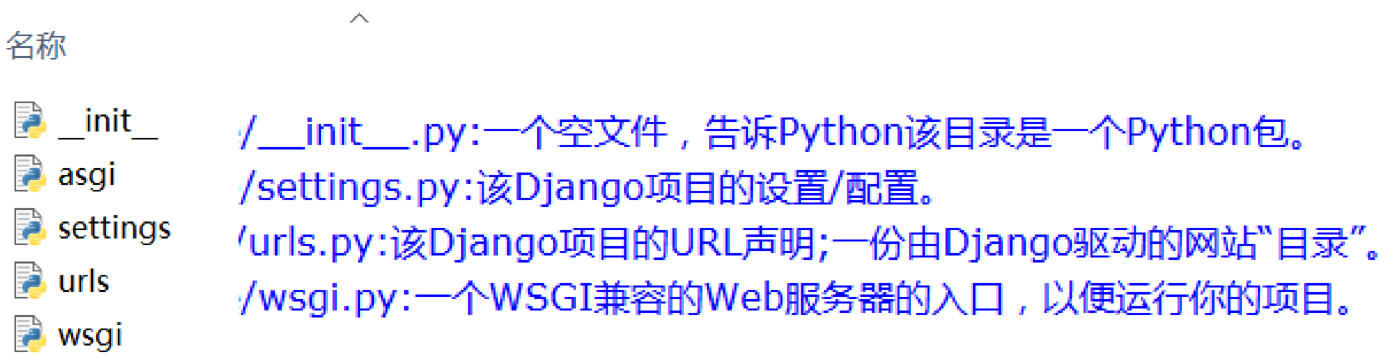
```
C:\Users\emmal>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> django.get_version()
'3.0.4'
>>> _
```

Django创建第一个项目

- ❑ 安装后，Django自带管理工具django-admin，可以使用django-admin来创建一个新项目
  - ❖ 通过命令行环境，进入你希望放置python工程的目录下，键入以下命令

```
django-admin startproject doubanrecommendation
```

生成一个工程目录“doubanrecommendation”(请使用个人名字拼音前缀的目录名称)



名称

- __init__
- asgi
- settings
- urls
- wsgi

/__init__.py:一个空文件，告诉Python该目录是一个Python包。
/settings.py:该Django项目的设置/配置。
'urls.py:该Django项目的URL声明;一份由Django驱动的网站"目录"。
/wsgi.py:一个WSGI兼容的Web服务器的入口，以便运行你的项目。

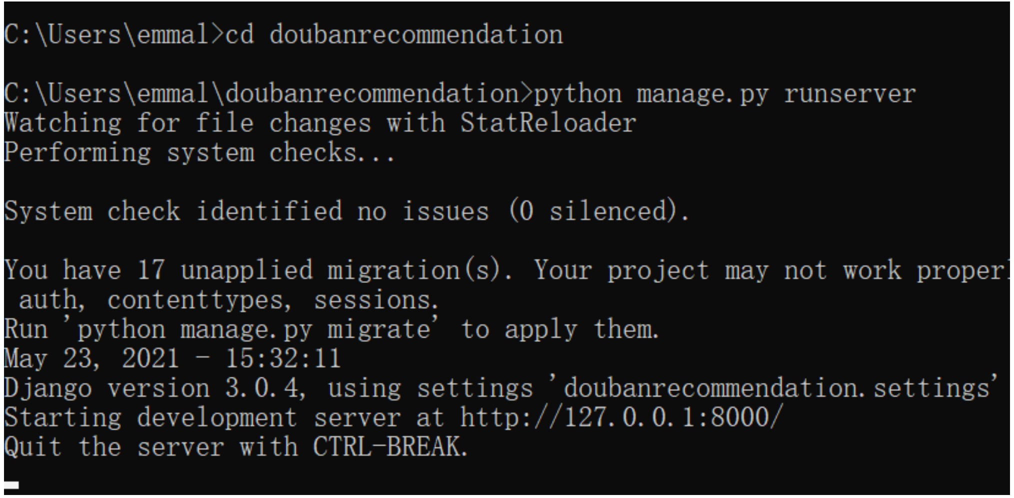
https://docs.djangoproject.com/en/3.0/intro/tutorial01/

Django创建第一个项目

- ❑ 测试Django工程环境，在工程目录“doubanrecommendation”(请进入自己创建的目录)下键入

  python manage.py runserver

  来启动服务器

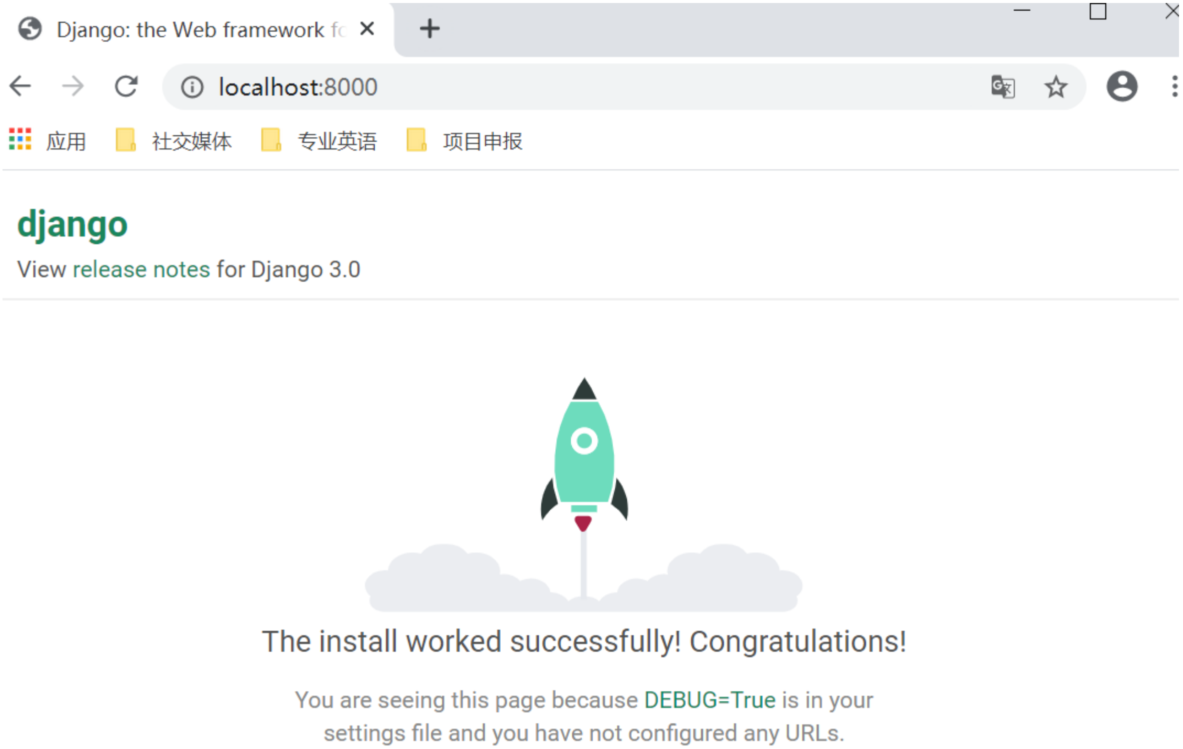


```
C:\Users\emmal>cd doubanrecommendation

C:\Users\emmal\doubanrecommendation>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work proper
 auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 23, 2021 - 15:32:11
Django version 3.0.4, using settings 'doubanrecommendation.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

https://docs.djangoproject.com/en/3.0/intro/tutorial01/

Django创建第一个项目

- ❑ 在浏览器中键入“http://localhost:8000” 或“http://127.0.0.1:8000”，如果服务器正常启动，输出结果如下：



django

View release notes for Django 3.0

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.

https://docs.djangoproject.com/en/3.0/intro/tutorial01/

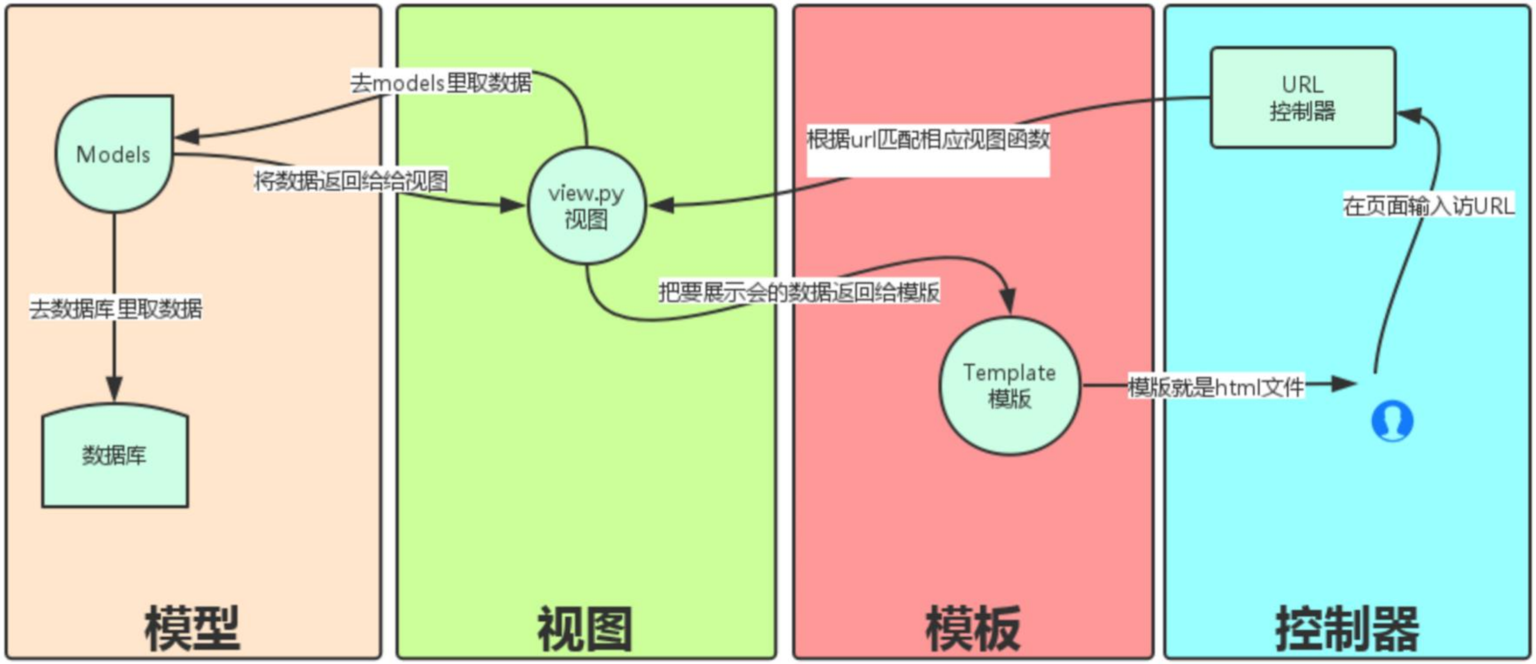
Django的 MVT 思想

- 核心思想：解耦合(将数据操作，业务处理，页面展示拆分)
  - **M: model (封装数据操作，用的最多的是数据库)**
  - **V: view (负责数据的显示和呈现，View是对用户的直接输出)**
  - **T: Template (模板处理业务，接收请求，协调model和view)**

Django的 MVT 思想

❑Django处理请求的流程

- ❖ Urls路由器-->视图函数View-->Models-->DB-->Models-->视图函数-->渲染数据到模板中



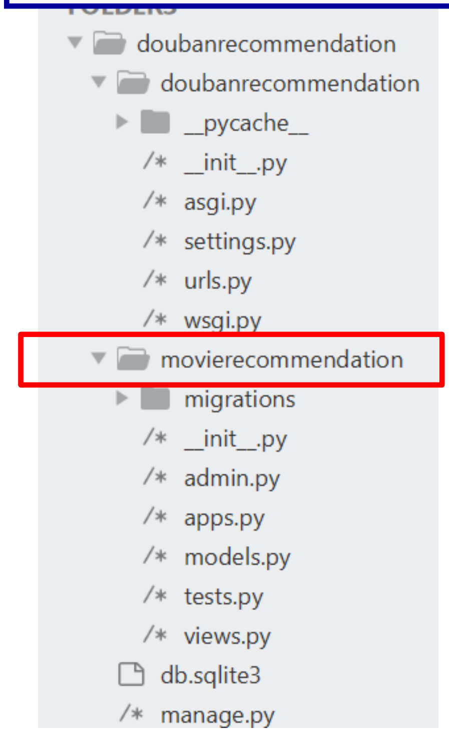
Django一般工作流程

- ❑编写相应的 urls.py
- ❑编写数据库models.py
- ❑编写处理 Http 请求的视图函数views.py
- ❑编写需要渲染的模板（Template）

Django创建第一个APP

❑创建一个APP（子系统，模块等），在工程目录下，键入

python manage.py startapp movierecommendation



```
▼ doubanrecommendation
  ▼ doubanrecommendation
    ▶ __pycache__
      /* __init__.py
      /* asgi.py
      /* settings.py
      /* urls.py
      /* wsgi.py
  ▼ movierecommendation
    ▶ migrations
      /* __init__.py
      /* admin.py
      /* apps.py
      /* models.py
      /* tests.py
      /* views.py
    db.sqlite3
    /* manage.py
```

请用名字拼音为前缀的APP名称

models.py文件主要用一个Python类来描述数据表

urls.py 指出了什么样的 URL 调用什么样的视图

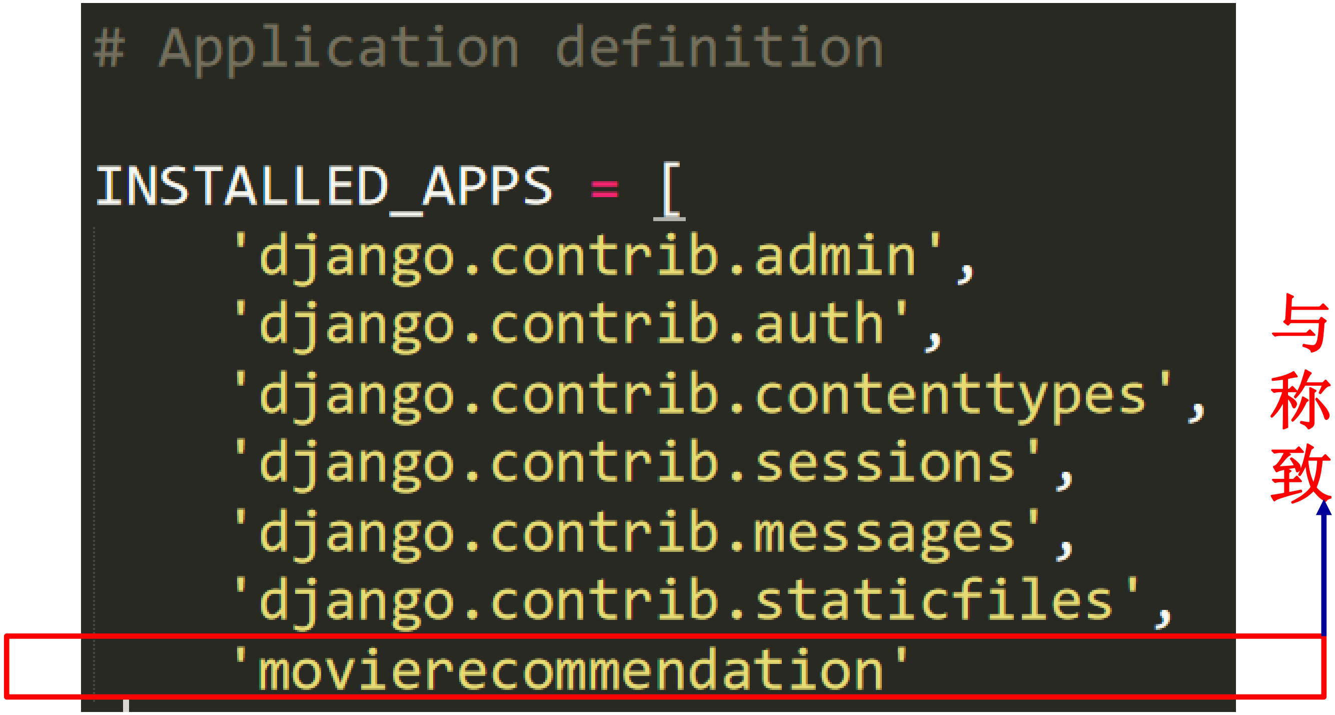
views.py文件包含了页面的业务逻辑

https://docs.djangoproject.com/en/3.0/intro/tutorial01/

Django创建第一个APP

❑注册APP

- ❖ 在settings.py中注册该APP

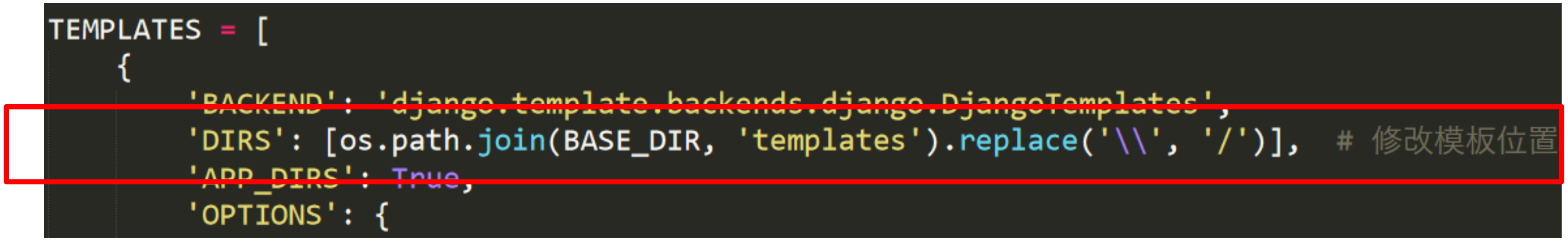


```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'movierecommendation'
```

与创建名称保持一致

Django创建页面模板目录

- □ 用sublime text等编辑器打开工程目录“doubanrecommendation” ” (请打开自己创建的目录)
  - ❖ File→Open Folder
- □ 在工程目录下创建templates目录用于存放所有的html模板页面
  - ❖ 在settings.py中添加模板路径

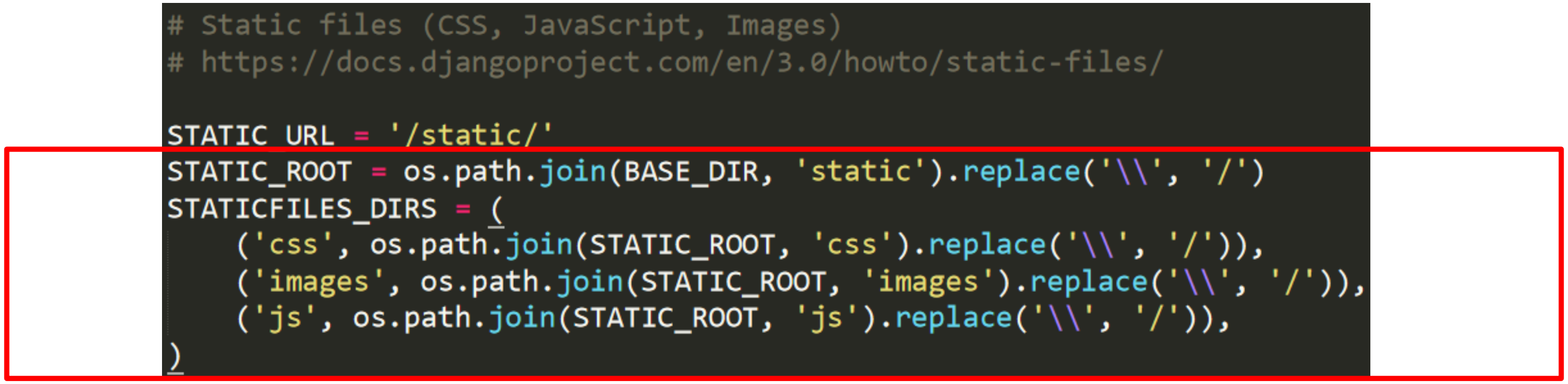


```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates').replace('\\', '/')],  # 修改模板位置
        'APP_DIRS': True,
        'OPTIONS': {
```

https://docs.djangoproject.com/en/3.0/intro/tutorial01/

Django创建静态文件目录

- ❑在工程目录下创建static文件夹，存放所有静态文件，如js，css，images等文件
  - ❖ 在settings.py文件最后添加路径



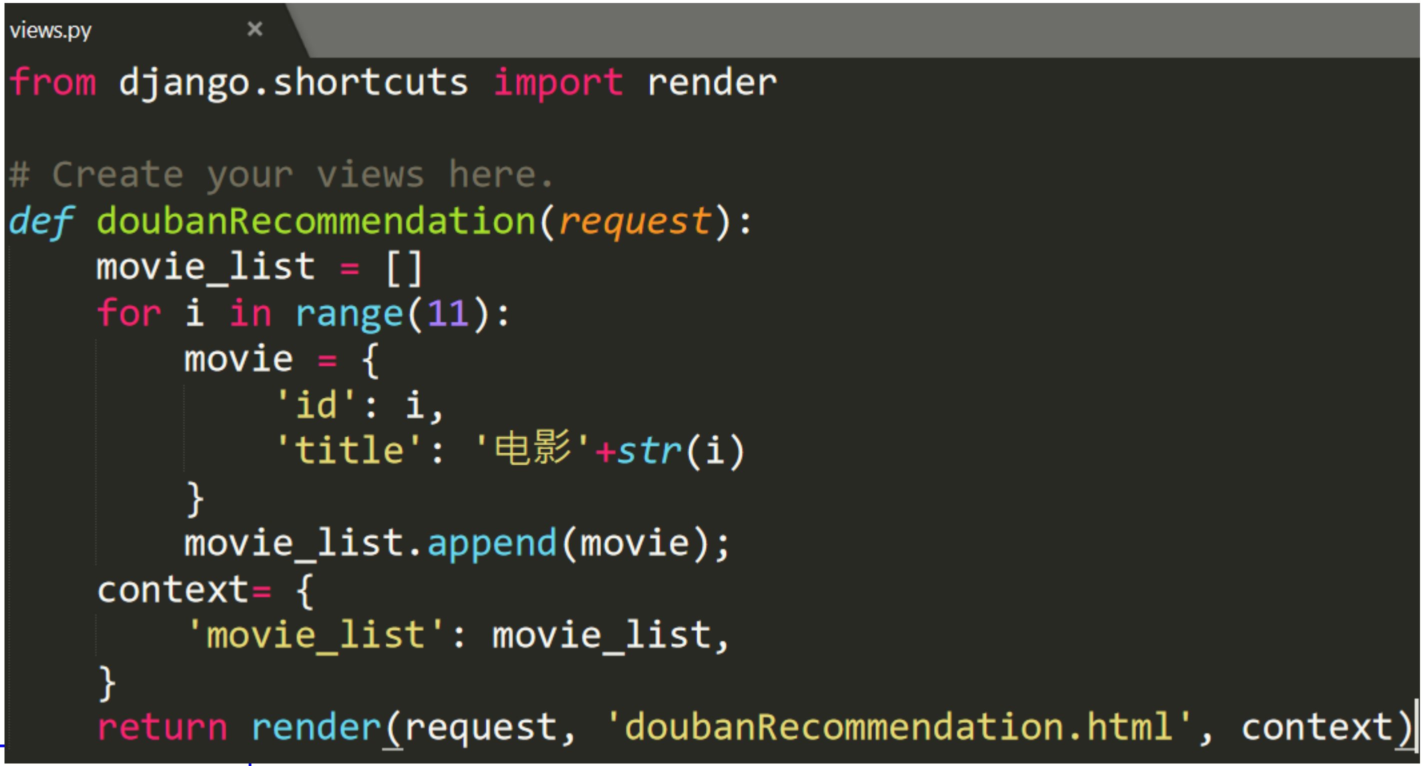
```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static').replace('\\', '/')
STATICFILES_DIRS = (
    ('css', os.path.join(STATIC_ROOT, 'css').replace('\\', '/')),
    ('images', os.path.join(STATIC_ROOT, 'images').replace('\\', '/')),
    ('js', os.path.join(STATIC_ROOT, 'js').replace('\\', '/')),
)
```

Django创建页面

与创建名称保持一致

❑创建一个静态页面

- ❖ movierecommendation/views.py编写处理Http 请求的视图函数



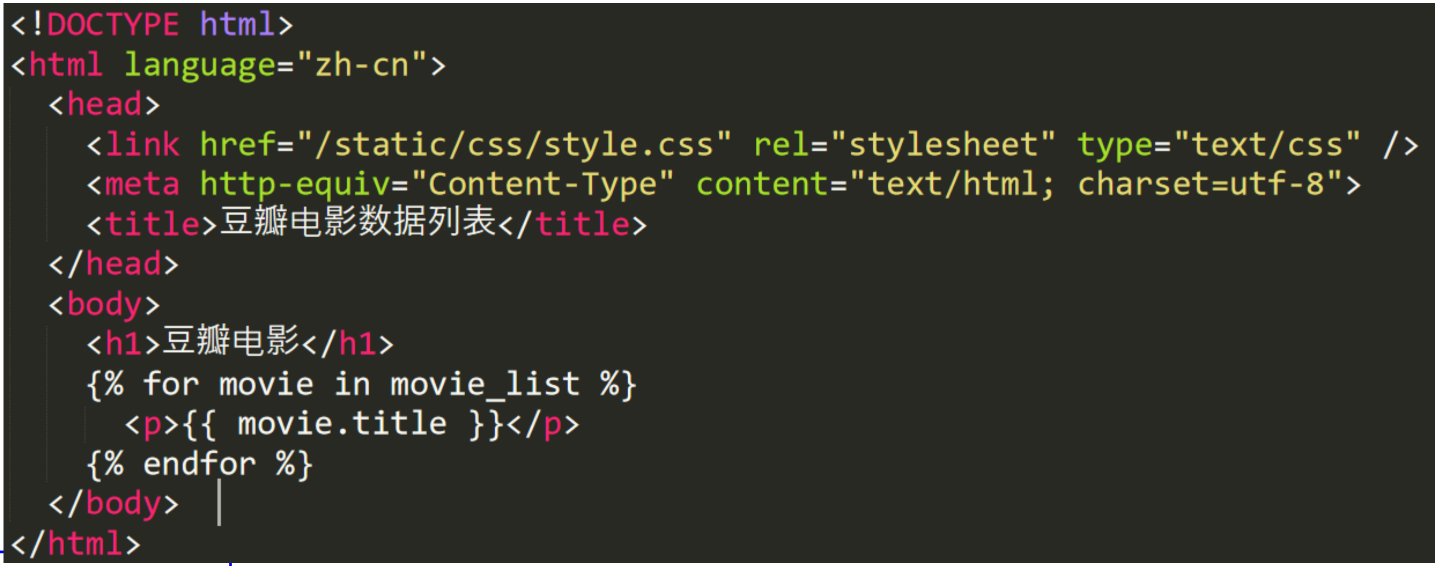
```python
from django.shortcuts import render

# Create your views here.
def doubanRecommendation(request):
    movie_list = []
    for i in range(11):
        movie = {
            'id': i,
            'title': '电影'+str(i)
        }
        movie_list.append(movie);
    context= {
        'movie_list': movie_list,
    }
    return render(request, 'doubanRecommendation.html', context)
```

Django创建页面

与创建名称保持一致

□创建一个静态页面

- ❖ moviereommendation/views.py编写处理Http 请求的视图函数
- ❖ templates下面创建一个html模板“doubanRecommendation.html”



```
<!DOCTYPE html>
<html language="zh-cn">
  <head>
    <link href="/static/css/style.css" rel="stylesheet" type="text/css" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>豆瓣电影数据列表</title>
  </head>
  <body>
    <h1>豆瓣电影</h1>
    {% for movie in movie_list %}
      <p>{{ movie.title }}</p>
    {% endfor %}
  </body>
</html>
```

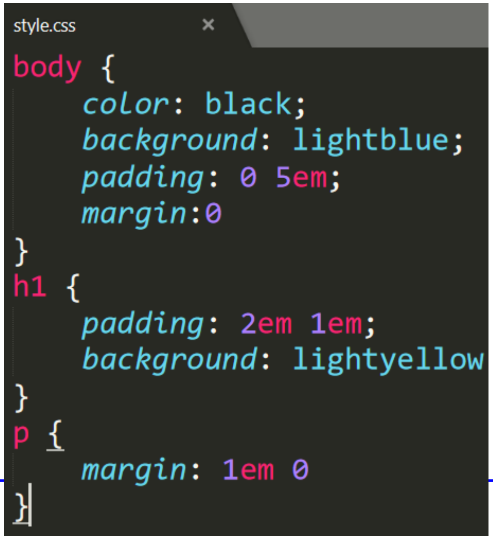
Django创建页面

与创建名称保持一致

□创建一个静态页面

- ❖ movierecommendation/views.py编写处理Http 请求的视图函数
- ❖ templates下面创建一个html模板“doubanRecommendation.html”
- ❖ static下面添加css/style.css文件

将html中所有css写到该文件

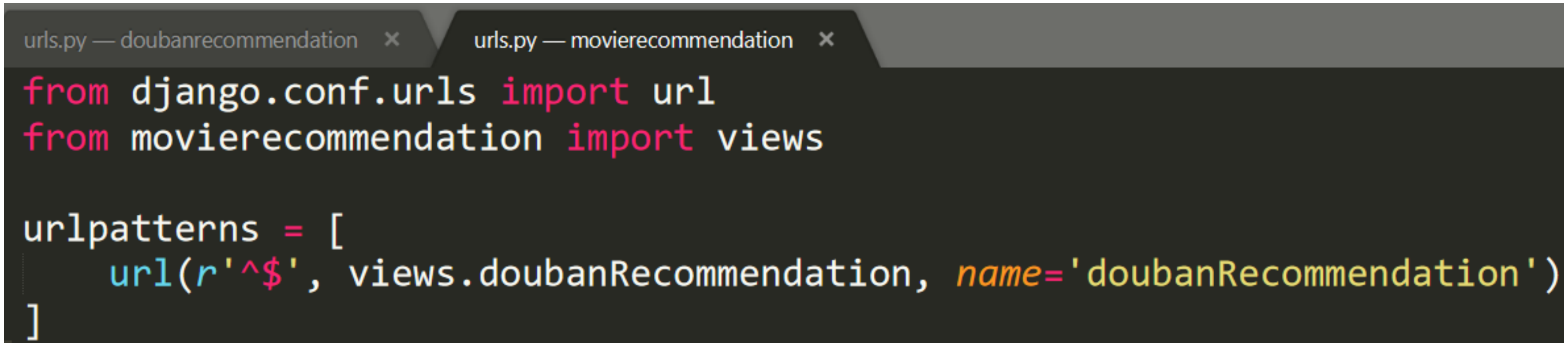

style.css ×

```css
body {
    color: black;
    background: lightblue;
    padding: 0 5em;
    margin:0
}
h1 {
    padding: 2em 1em;
    background: lightyellow
}
p {
    margin: 1em 0
}
```

Django创建页面

与创建名称保持一致

- ❑ 创建一个静态页面
  - ❖ movierecommendation/views.py编写处理Http 请求的视图函数
- ❑ 创建页面链接
  - ❖ 创建文件movierecommendation/urls.py，绑定 URL与视图函数


urls.py — doubanrecommendation ×
urls.py — movierecommendation ×

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

urlpatterns = [
    url(r'^$', views.doubanRecommendation, name='doubanRecommendation')
]
```

Django创建页面

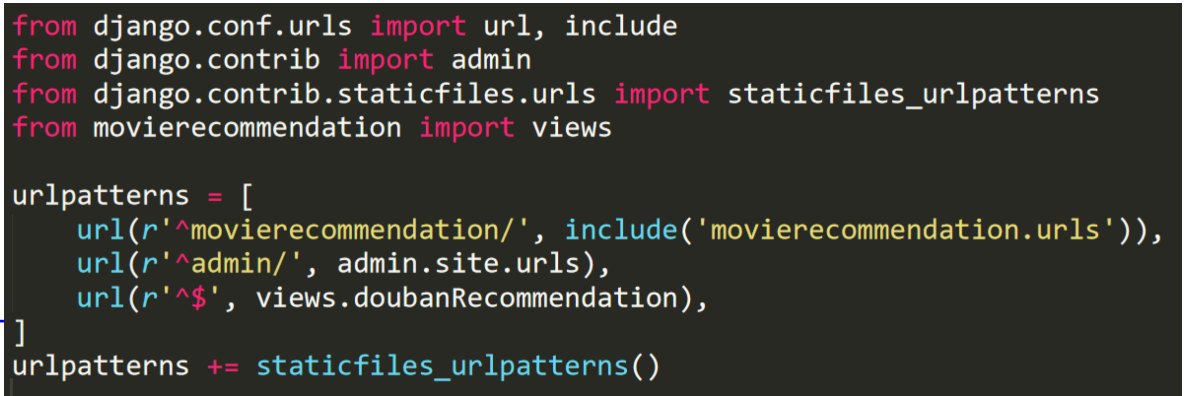
□创建一个静态页面

❖ movierecommendation/views.py编写处理Http 请求的视图函数

□创建页面链接

❖ 创建文件movierecommendation/urls.py，绑定 URL与视图函数

❖ doubanrecommendation/urls.py中添加新的url pattern

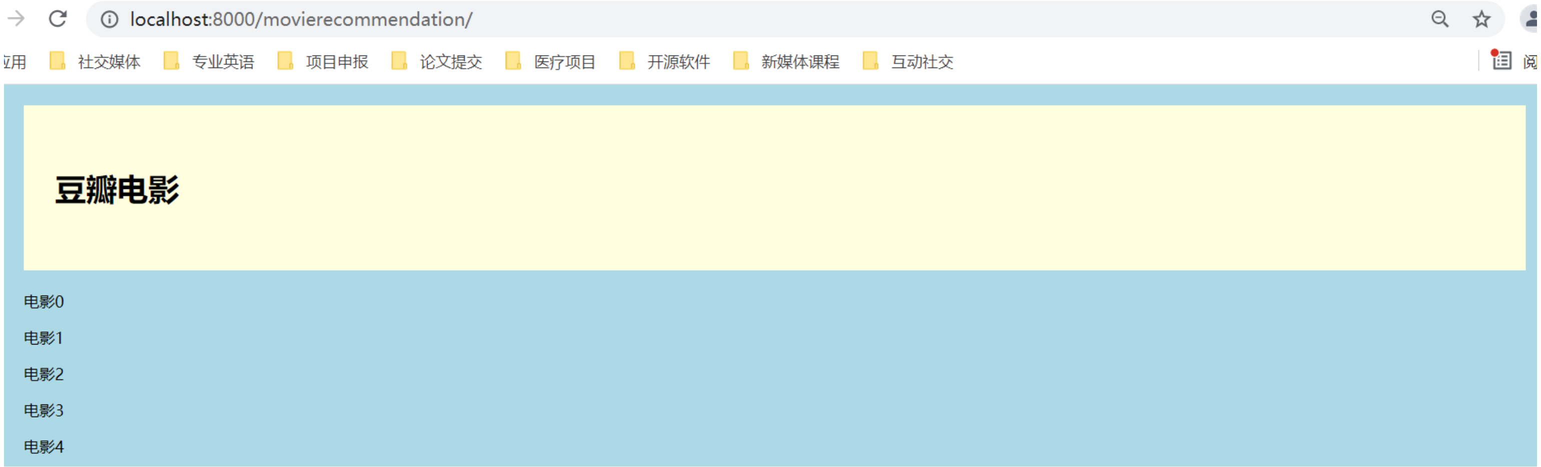


```
from django.conf.urls import url, include
from django.contrib import admin
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from movierecommendation import views

urlpatterns = [
    url(r'^movierecommendation/', include('movierecommendation.urls')),
    url(r'^admin/', admin.site.urls),
    url(r'^$', views.doubanRecommendation),
]
urlpatterns += staticfiles_urlpatterns()
```

Django页面测试

❑测试新页面

- ❖命令行环境中，在工程目录下键入“python manage.py runserver”启动服务器
- ❖若成功启动，在浏览器中键入“http://localhost:8000/movierecommendation/”



豆瓣电影

电影0

电影1

电影2

电影3

电影4

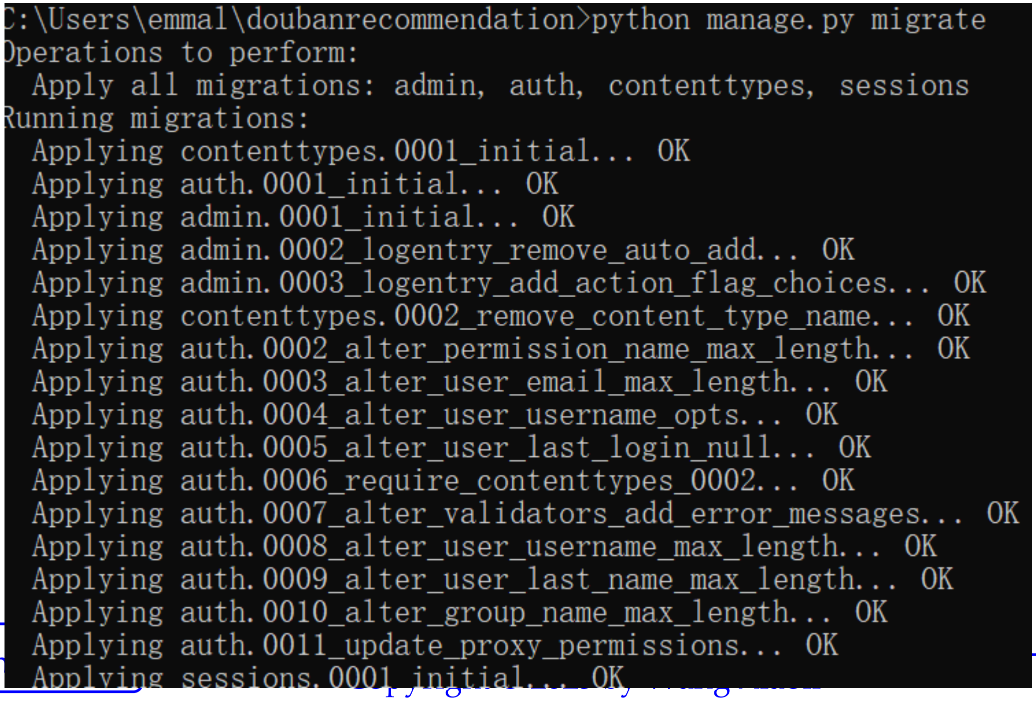
https://docs.djangoproject.com/en/2.0/intro/tutorial01/

数据存储与展示

Django后台管理系统

- ❑ Django默认包含网站后台管理系统
  - ❖ Django默认包含的应用需要导入数据表，在工程目录下，键入

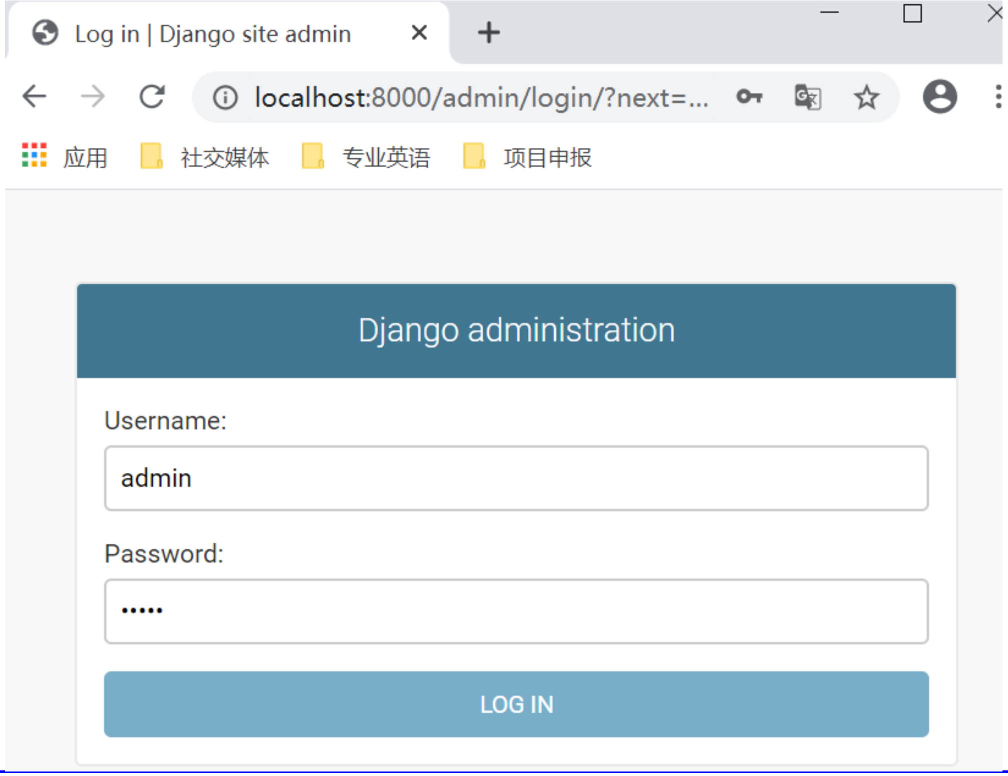
```
python manage.py migrate
```



```
C:\Users\emmal\doubanrecommendation>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 auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
```

Django后台管理系统

- 在工程目录键入“python manage.py runserver”启动服务器，浏览器输入“http://localhost:8000/admin/”测试



Log in | Django site admin
localhost:8000/admin/login/?next=...
应用
社交媒体
专业英语
项目申报
Django administration

Username:

admin

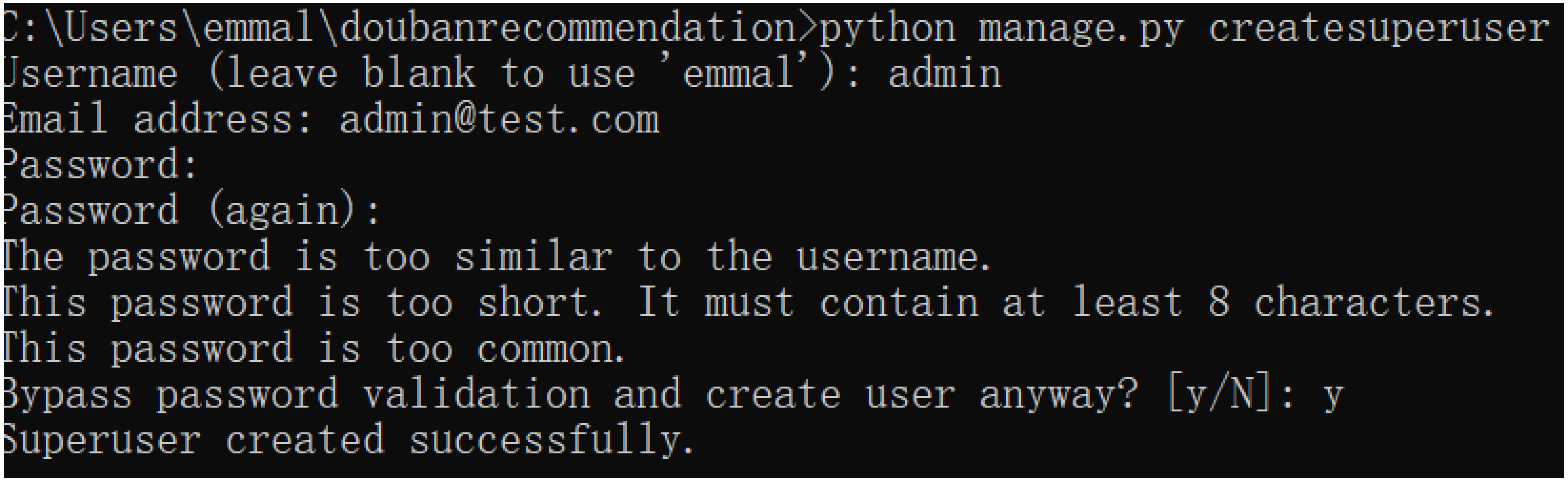
Password:

•••••

Django后台管理系统

- ❑ 使用Django自带的后台管理系统，需要在命令行先创建超级管理用户，在工程目录键入

```
python manage.py createsuperuser
```

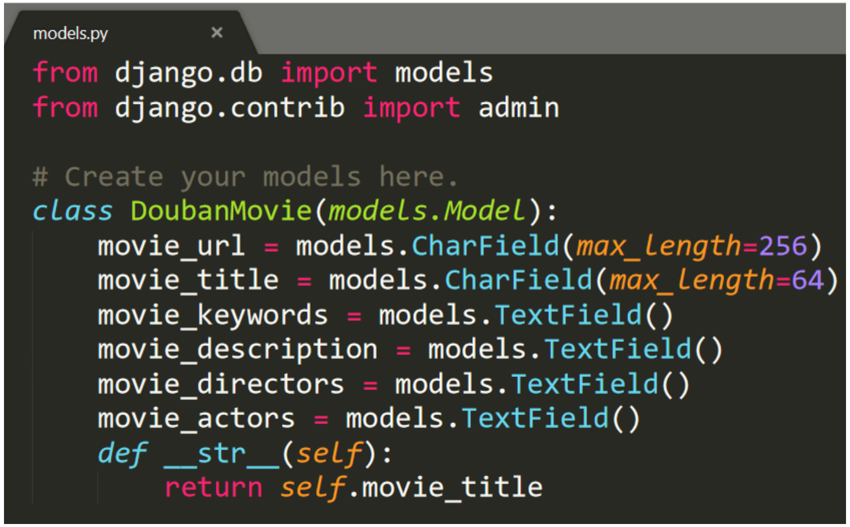


```
C:\Users\emmal\doubanrecommendation>python manage.py createsuperuser
Username (leave blank to use 'emmal'): admin
Email address: admin@test.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Django如何创建新表

❑利用models.py创建新表

python manage.py makemigrations



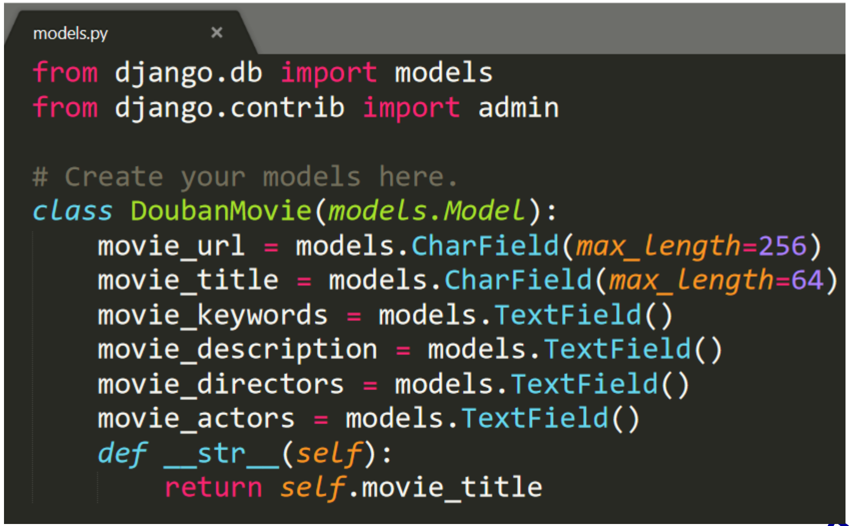
```python
from django.db import models
from django.contrib import admin

# Create your models here.
class DoubanMovie(models.Model):
    movie_url = models.CharField(max_length=256)
    movie_title = models.CharField(max_length=64)
    movie_keywords = models.TextField()
    movie_description = models.TextField()
    movie_directors = models.TextField()
    movie_actors = models.TextField()
    def __str__(self):
        return self.movie_title
```

创建类的时候可以根据已有的表进行设计，比如数据类型，长度等

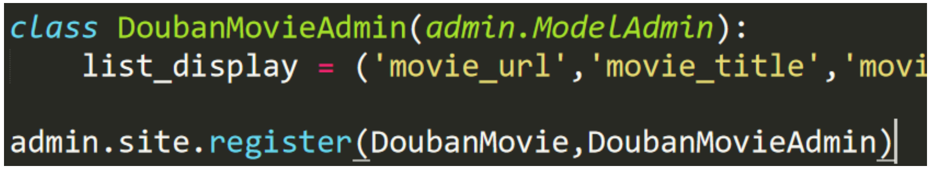
Django如何创建新表

❑利用models.py创建新表

python manage.py makemigrations



```
from django.db import models
from django.contrib import admin

# Create your models here.
class DoubanMovie(models.Model):
    movie_url = models.CharField(max_length=256)
    movie_title = models.CharField(max_length=64)
    movie_keywords = models.TextField()
    movie_description = models.TextField()
    movie_directors = models.TextField()
    movie_actors = models.TextField()
    def __str__(self):
        return self.movie_title
```



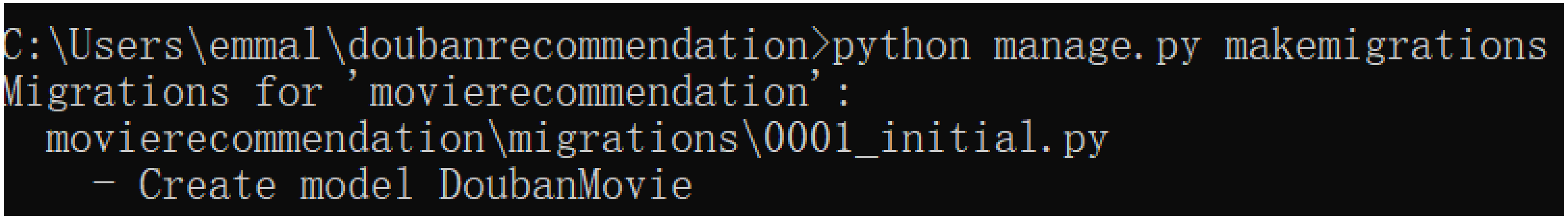
```
class DoubanMovieAdmin(admin.ModelAdmin):
    list_display = ('movie_url','movie_title','movi

admin.site.register(DoubanMovie,DoubanMovieAdmin)
```

创建后的类如果想要在Django后台管理系统进行管理，就需要在admin.py里头注册

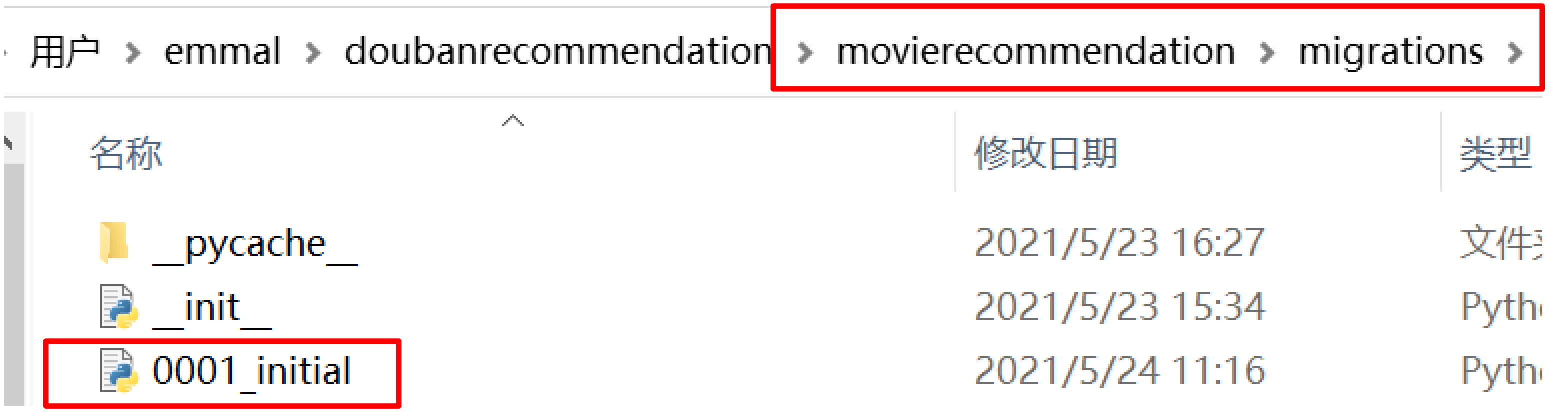
Django如何创建新表

❑利用models.py创建新表

在命令行输入python manage.py makemigrations创建迁移文件



```
C:\Users\emmal\doubanrecommendation>python manage.py makemigrations
Migrations for 'movierecommendation':
  movierecommendation\migrations\0001_initial.py
    - Create model DoubanMovie
```



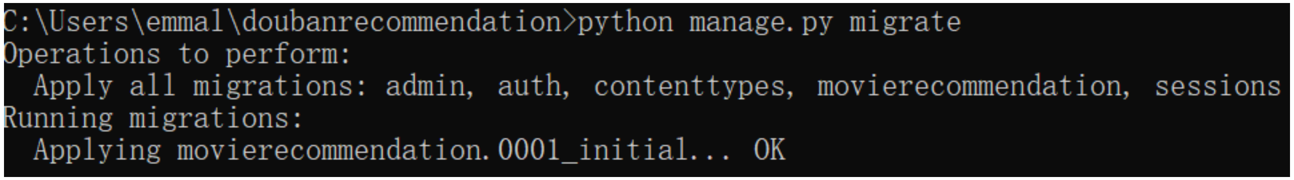
用户 › emmal › doubanrecommendation › movierecommendation › migrations ›

名称	修改日期	类型
__pycache__	2021/5/23 16:27	文件
__init__	2021/5/23 15:34	Pyth
0001_initial	2021/5/24 11:16	Pyth

Django如何创建新表

❑利用models.py创建新表

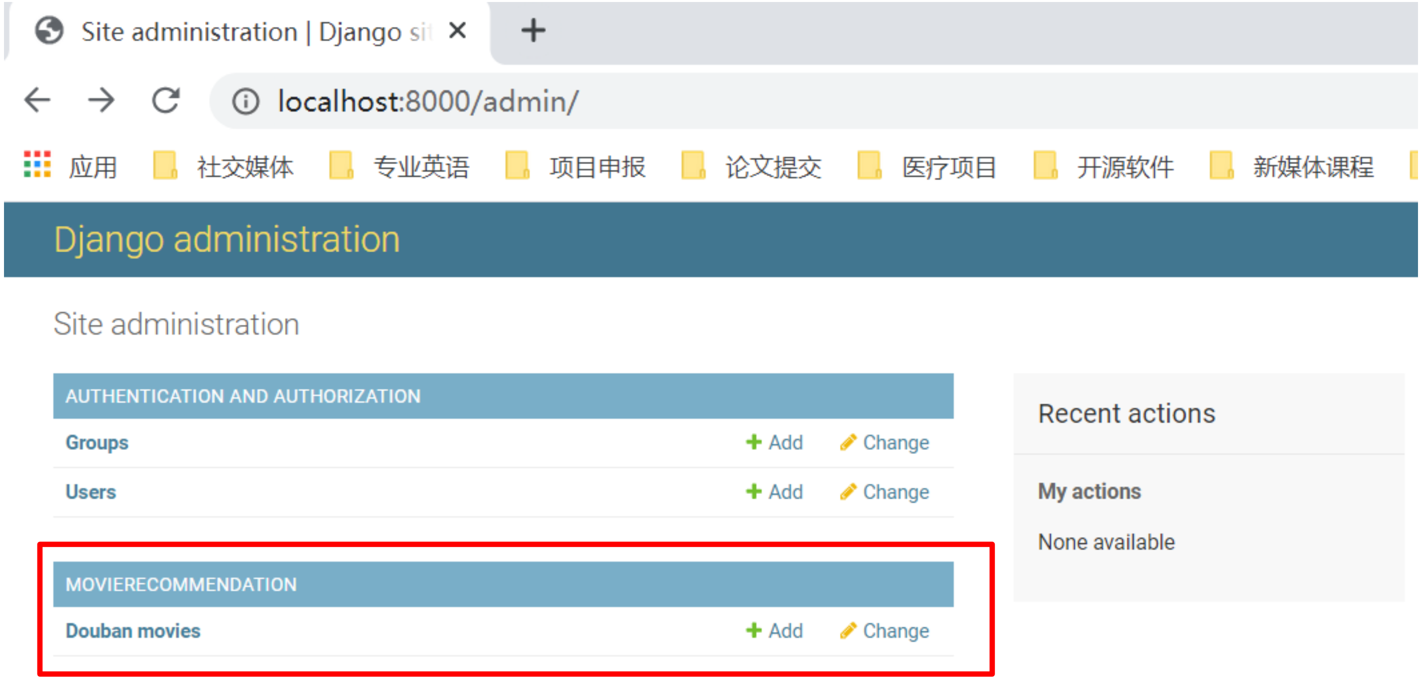
在命令行输入python manage.py migrate完成迁移文件的导入



```
C:\Users\emmal\doubanrecommendation>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, movierecommendation, sessions
Running migrations:
  Applying movierecommendation.0001_initial... OK
```

后台管理系统查看创建的新表

- ❑在工程目录键入“python manage.py runserver”启动服务器，浏览器输入http://localhost:8000/admin/登录管理员测试



Site administration | Django si ×

localhost:8000/admin/

应用 社交媒体 专业英语 项目申报 论文提交 医疗项目 开源软件 新媒体课程

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

MOVIERECOMMENDATION		
Douban movies	+ Add	Change

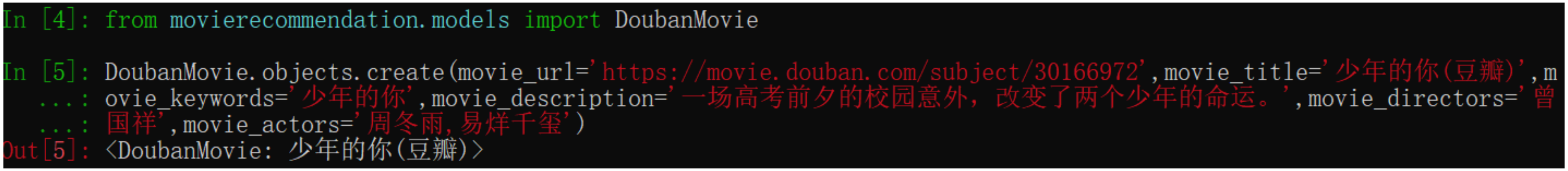
Recent actions

My actions

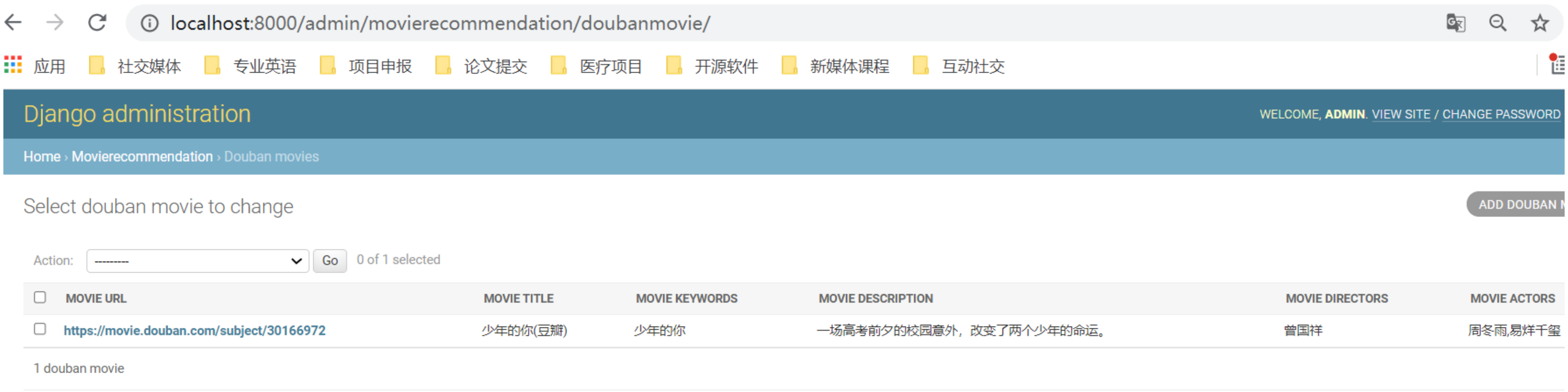
None available

Django如何插入数据

在命令行输入python manage.py shell进行数据插入



```
In [4]: from movierecommendation.models import DoubanMovie

In [5]: DoubanMovie.objects.create(movie_url='https://movie.douban.com/subject/30166972',movie_title='少年的你(豆瓣)',m
   ...: ovie_keywords='少年的你',movie_description='一场高考前夕的校园意外，改变了两个少年的命运。',movie_directors='曾
   ...: 国祥',movie_actors='周冬雨,易烊千玺')
Out[5]: <DoubanMovie: 少年的你(豆瓣)>
```



localhost:8000/admin/movierecommendation/doubanmovie/
应用
社交媒体
专业英语
项目申报
论文提交
医疗项目
开源软件
新媒体课程
互动社交
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD
Home › Movierecommendation › Douban movies
ADD DOUBAN M
Action:
Go

Select douban movie to change

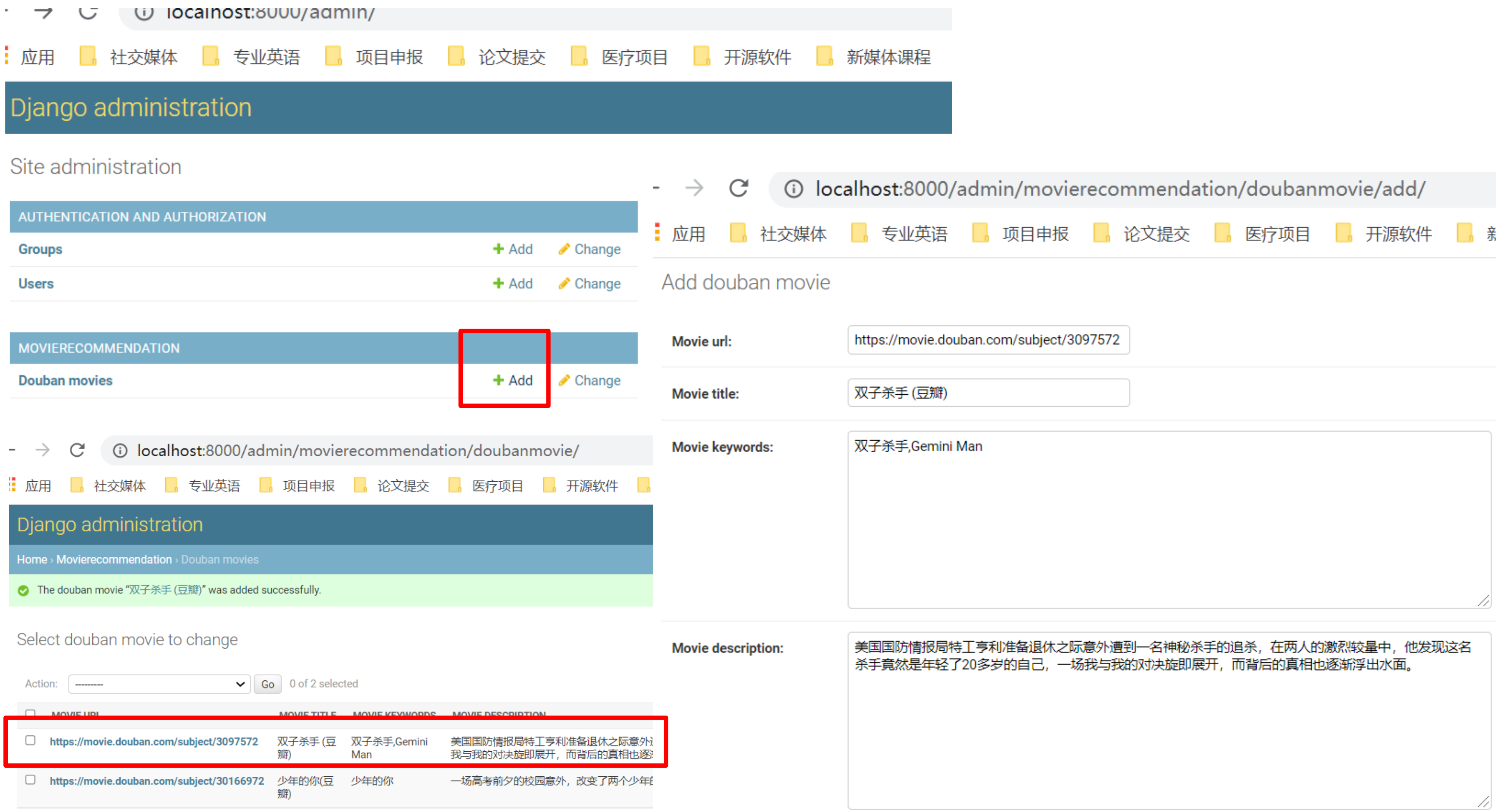
Action: --------- Go 0 of 1 selected

□	MOVIE URL	MOVIE TITLE	MOVIE KEYWORDS	MOVIE DESCRIPTION	MOVIE DIRECTORS	MOVIE ACTORS
□	https://movie.douban.com/subject/30166972	少年的你(豆瓣)	少年的你	一场高考前夕的校园意外，改变了两个少年的命运。	曾国祥	周冬雨,易烊千玺

1 douban movie

Django如何插入数据

进入Django admin表格管理页面进行手动插入



localhost:8000/admin/movierecommendation/doubanmovie/add/

Add douban movie

Movie url:	https://movie.douban.com/subject/3097572
Movie title:	双子杀手 (豆瓣)
Movie keywords:	双子杀手,Gemini Man
Movie description:	美国国防情报局特工亨利准备退休之际意外遭到一名神秘杀手的追杀，在两人的激烈较量中，他发现这名杀手竟然是年轻了20多岁的自己，一场我与我的对决旋即展开，而背后的真相也逐渐浮出水面。

localhost:8000/admin/movierecommendation/doubanmovie/

Django administration

Home › Movierecommendation › Douban movies

The douban movie “双子杀手 (豆瓣)” was added successfully.

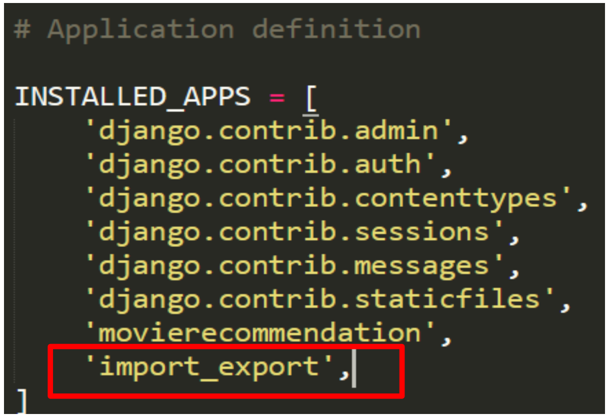
Select douban movie to change

Action: --------- Go 0 of 2 selected

☐	MOVIE URL	MOVIE TITLE	MOVIE KEYWORDS	MOVIE DESCRIPTION
☐	https://movie.douban.com/subject/3097572	双子杀手 (豆瓣)	双子杀手,Gemini Man	美国国防情报局特工亨利准备退休之际意外 我与我的对决旋即展开，而背后的真相也逐
☐	https://movie.douban.com/subject/30166972	少年的你(豆瓣)	少年的你	一场高考前夕的校园意外，改变了两个少年

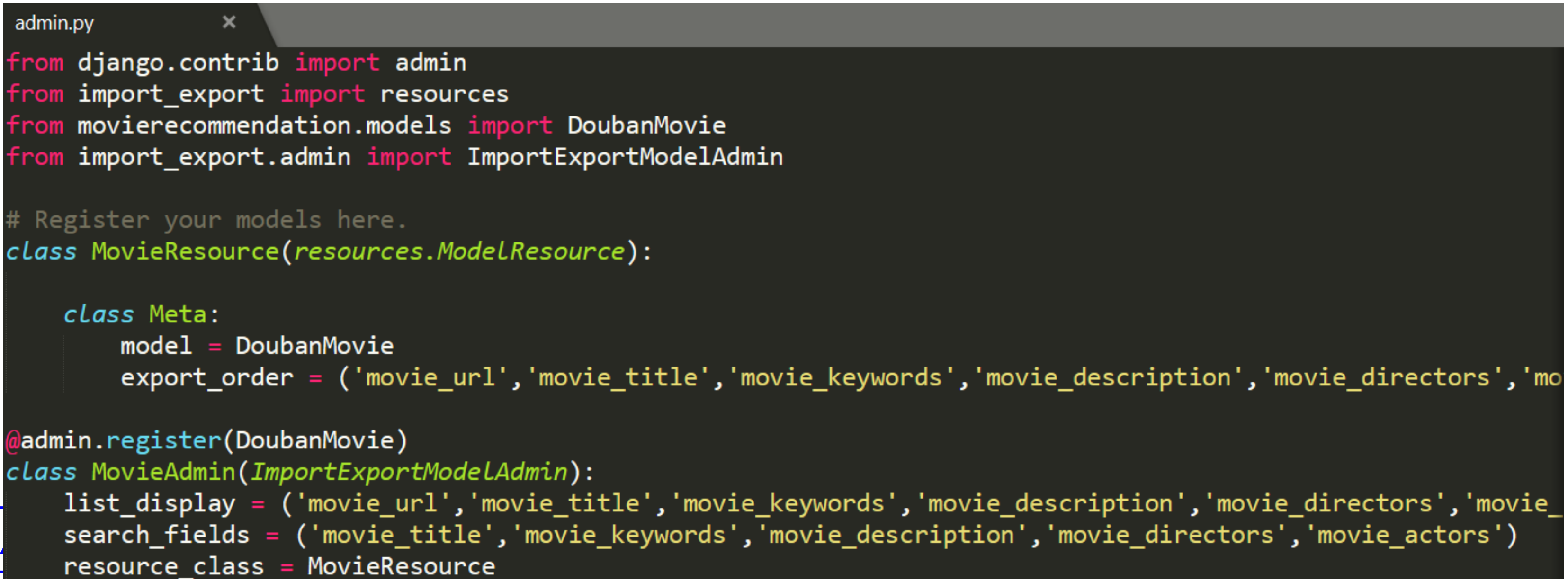
Django如何插入数据

将爬取的数据文件CSV中的数据导入后台数据库



```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'movierecommendation',
    'import_export',
]
```

1. pip install django-import-export
2. 配置settings.py
3. python manage.py collectstatic
4. models.py建立model
5. 在admin.py创建Resource、对应的Admin

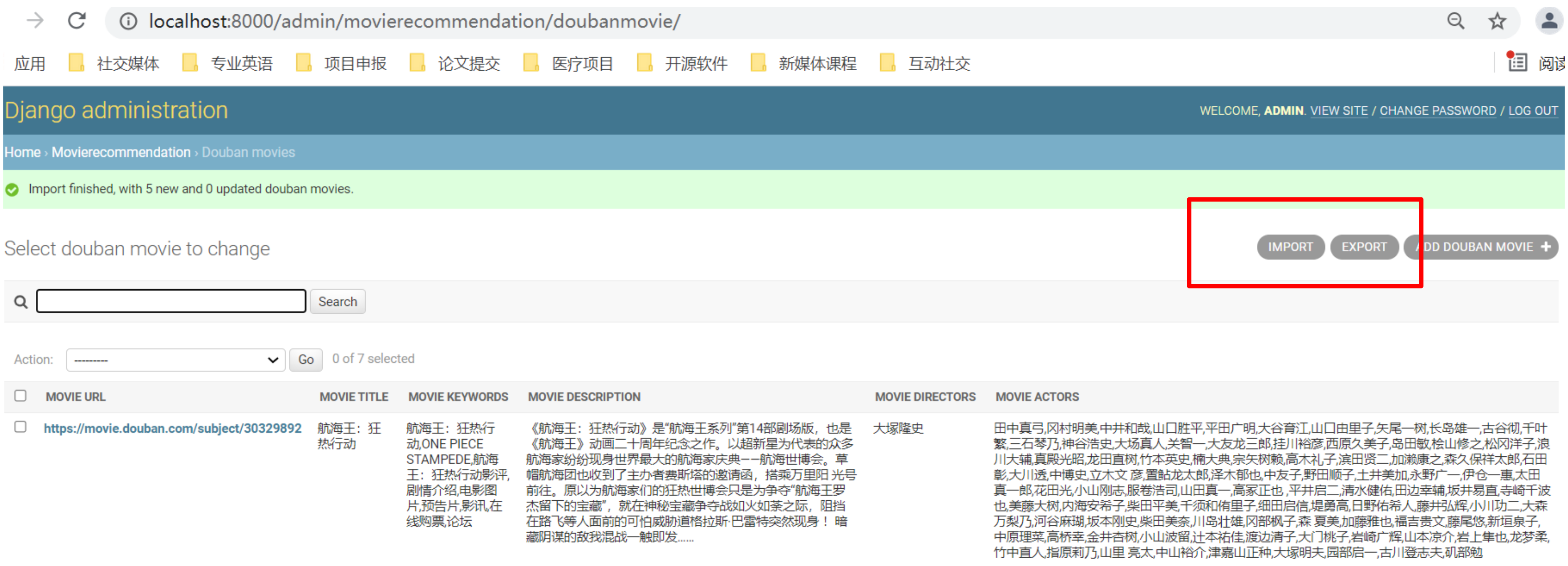


```
from django.contrib import admin
from import_export import resources
from movierecommendation.models import DoubanMovie
from import_export.admin import ImportExportModelAdmin

# Register your models here.
class MovieResource(resources.ModelResource):

    class Meta:
        model = DoubanMovie
        export_order = ('movie_url','movie_title','movie_keywords','movie_description','movie_directors','mo

@admin.register(DoubanMovie)
class MovieAdmin(ImportExportModelAdmin):
    list_display = ('movie_url','movie_title','movie_keywords','movie_description','movie_directors','movie_
    search_fields = ('movie_title','movie_keywords','movie_description','movie_directors','movie_actors')
    resource_class = MovieResource
```

Django如何插入数据

进入Django admin表格管理页面进行手动导入

注意CSV文件的格式要与数据表一致



Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Movierecommendation › Douban movies

Import finished, with 5 new and 0 updated douban movies.

Select douban movie to change

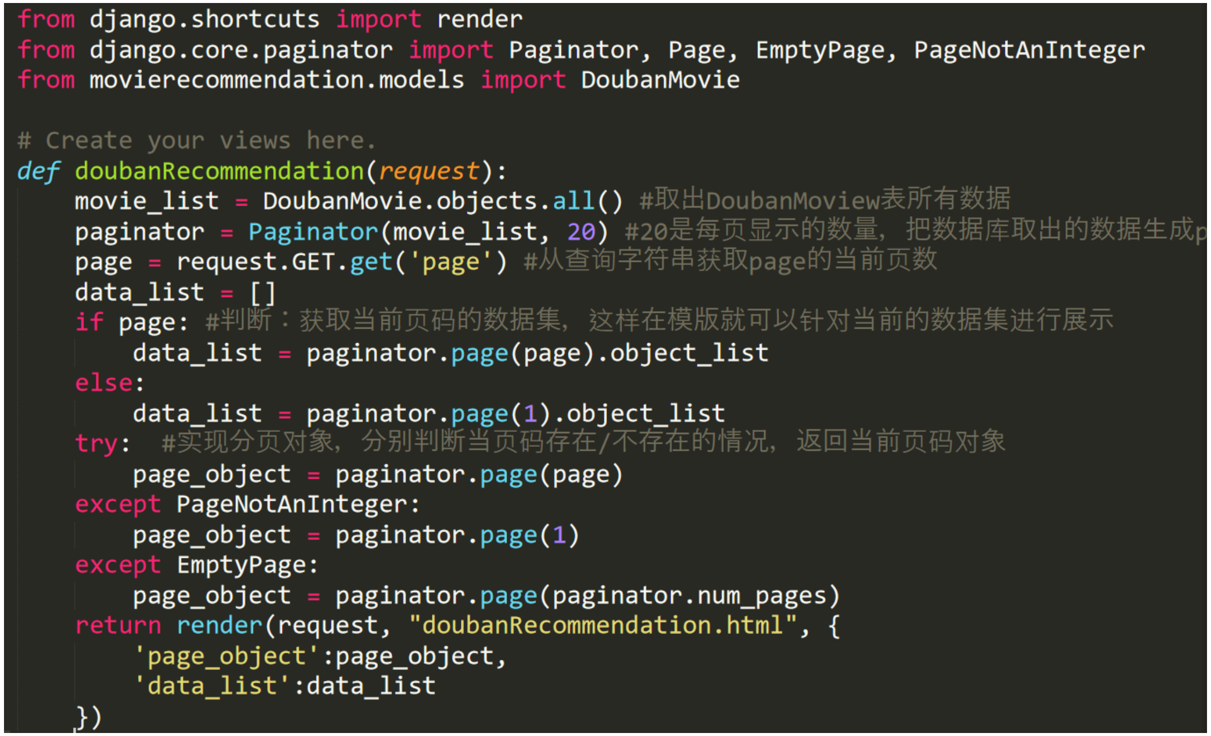
0 of 7 selected

□	MOVIE URL	MOVIE TITLE	MOVIE KEYWORDS	MOVIE DESCRIPTION	MOVIE DIRECTORS	MOVIE ACTORS
□	https://movie.douban.com/subject/30329892	航海王：狂热行动	航海王：狂热行动,ONE PIECE STAMPEDE,航海王：狂热行动影评,剧情介绍,电影图片,预告片,影讯,在线购票,论坛	《航海王：狂热行动》是"航海王系列"第14部剧场版，也是《航海王》动画二十周年纪念之作。以超新星为代表的众多航海家纷纷现身世界最大的航海家庆典——航海世博会。草帽航海团也收到了主办者费斯塔的邀请函，搭乘万里阳 光号前往。原以为航海家们的狂热世博会只是为争夺"航海王罗杰留下的宝藏"，就在神秘宝藏争夺战如火如荼之际，阻挡在路飞等人面前的可怕威胁道格拉斯·巴雷特突然现身！ 暗藏阴谋的敌我混战一触即发......	大塚隆史	田中真弓,冈村明美,中井和哉,山口胜平,平田广明,大谷育江,山口由里子,矢尾一树,长岛雄一,古谷彻,千叶繁,三石琴乃,神谷浩史,大场真人,关智一,大友龙三郎,挂川裕彦,西原久美子,岛田敏,桧山修之,松冈洋子,浪川大辅,真殿光昭,龙田直树,竹本英史,楠大典,宗矢树赖,高木礼子,滨田贤二,加濑康之,森久保祥太郎,石田彰,大川透,中博史,立木文 彦,置鮎龙太郎,泽木郁也,中友子,野田顺子,土井美加,永野广一,伊仓一惠,太田真一郎,花田光,小山刚志,服卷浩司,山田真一,高冢正也 ,平井启二,清水健佑,田边幸辅,坂井易直,寺崎千波也,美藤大树,内海安希子,柴田平美,千须和侑里子,细田启信,堤勇高,日野佑希人,藤井弘辉,小川功二,大森万梨乃,河谷麻瑚,坂本刚史,柴田美奈,川岛壮雄,冈部枫子,森 夏美,加藤雅也,福吉贵文,藤尾悠,新垣泉子,中原理菜,高桥幸,金井杏树,小山波留,辻本祐佳,渡边清子,大门桃子,岩崎广辉,山本凉介,岩上隼也,龙梦柔,竹中直人,指原莉乃,山里 亮太,中山裕介,津嘉山正种,大塚明夫,园部启一,古川登志夫,矶部勉

Django如何查询并展示数据

❑ 前端页面和数据库表的数据交互

❖ views.py编写处理 Http 请求的视图函数，将数据库中的数据进行返回

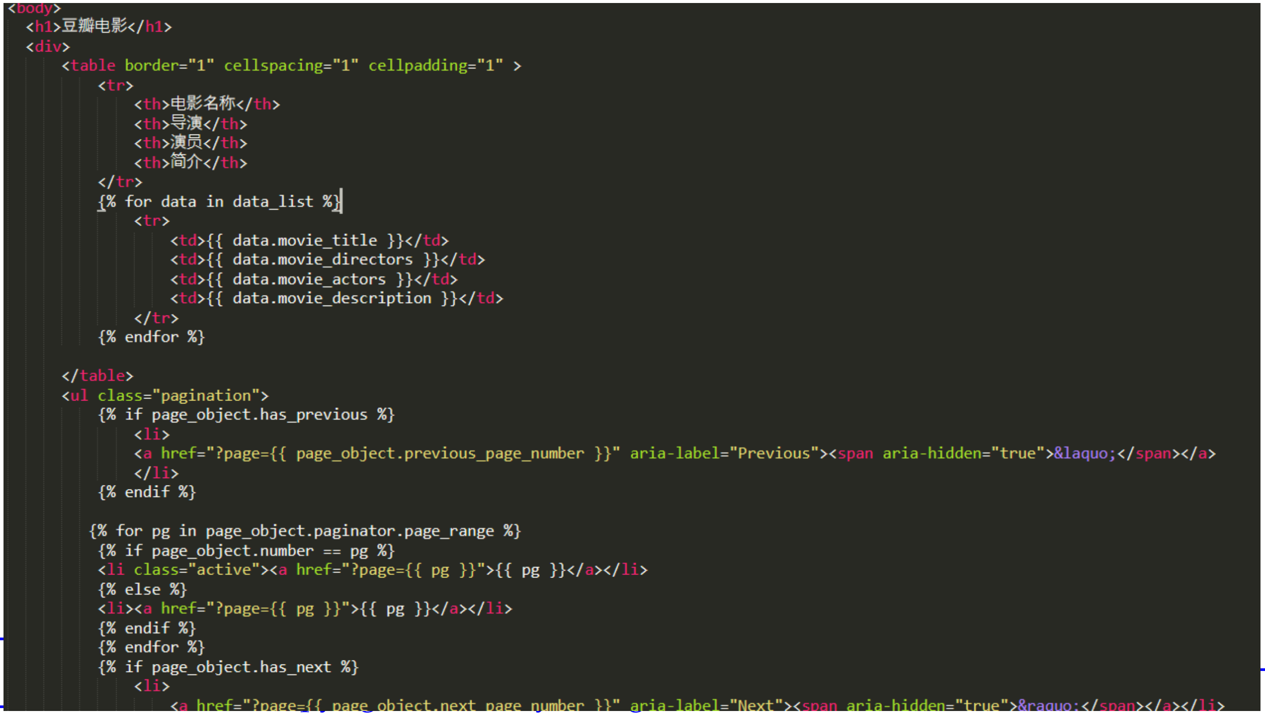


```
from django.shortcuts import render
from django.core.paginator import Paginator, Page, EmptyPage, PageNotAnInteger
from movierecommendation.models import DoubanMovie

# Create your views here.
def doubanRecommendation(request):
    movie_list = DoubanMovie.objects.all() #取出DoubanMoview表所有数据
    paginator = Paginator(movie_list, 20) #20是每页显示的数量，把数据库取出的数据生成p
    page = request.GET.get('page') #从查询字符串获取page的当前页数
    data_list = []
    if page: #判断：获取当前页码的数据集，这样在模版就可以针对当前的数据集进行展示
        data_list = paginator.page(page).object_list
    else:
        data_list = paginator.page(1).object_list
    try:  #实现分页对象，分别判断当前页码存在/不存在的情况，返回当前页码对象
        page_object = paginator.page(page)
    except PageNotAnInteger:
        page_object = paginator.page(1)
    except EmptyPage:
        page_object = paginator.page(paginator.num_pages)
    return render(request, "doubanRecommendation.html", {
        'page_object':page_object,
        'data_list':data_list
    })
```

Python快速建站

□修改页面模板

- ❖ views.py编写处理 Http 请求的视图函数
- ❖ 修改templates下面页面模板“doubanRecommendation.html”

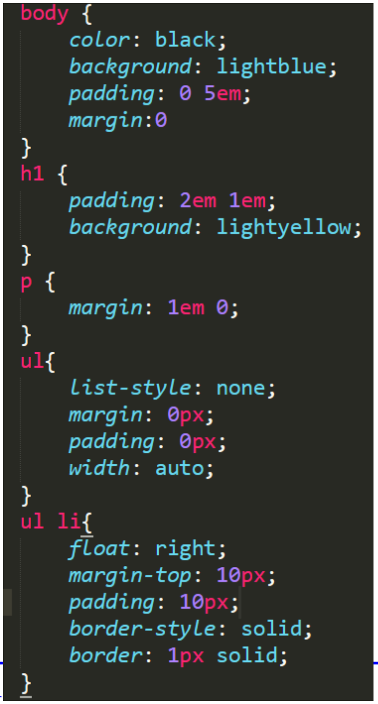


```
<body>
  <h1>豆瓣电影</h1>
  <div>
      <table border="1" cellspacing="1" cellpadding="1" >
          <tr>
              <th>电影名称</th>
              <th>导演</th>
              <th>演员</th>
              <th>简介</th>
          </tr>
          {% for data in data_list %}
              <tr>
                  <td>{{ data.movie_title }}</td>
                  <td>{{ data.movie_directors }}</td>
                  <td>{{ data.movie_actors }}</td>
                  <td>{{ data.movie_description }}</td>
              </tr>
          {% endfor %}

      </table>
      <ul class="pagination">
          {% if page_object.has_previous %}
              <li>
              <a href="?page={{ page_object.previous_page_number }}" aria-label="Previous"><span aria-hidden="true">&laquo;</span></a>
              </li>
          {% endif %}

         {% for pg in page_object.paginator.page_range %}
          {% if page_object.number == pg %}
          <li class="active"><a href="?page={{ pg }}">{{ pg }}</a></li>
          {% else %}
          <li><a href="?page={{ pg }}">{{ pg }}</a></li>
          {% endif %}
          {% endfor %}
          {% if page_object.has_next %}
              <li>
                  <a href="?page={{ page_object.next_page_number }}" aria-label="Next"><span aria-hidden="true">&raquo;</span></a></li>
```

Python快速建站

❑ 修改CSS

- ❖ views.py编写处理 Http 请求的视图函数
- ❖ 修改templates下面页面模板“doubanRecommendation.html
- ❖ 修改css文件“style.css”

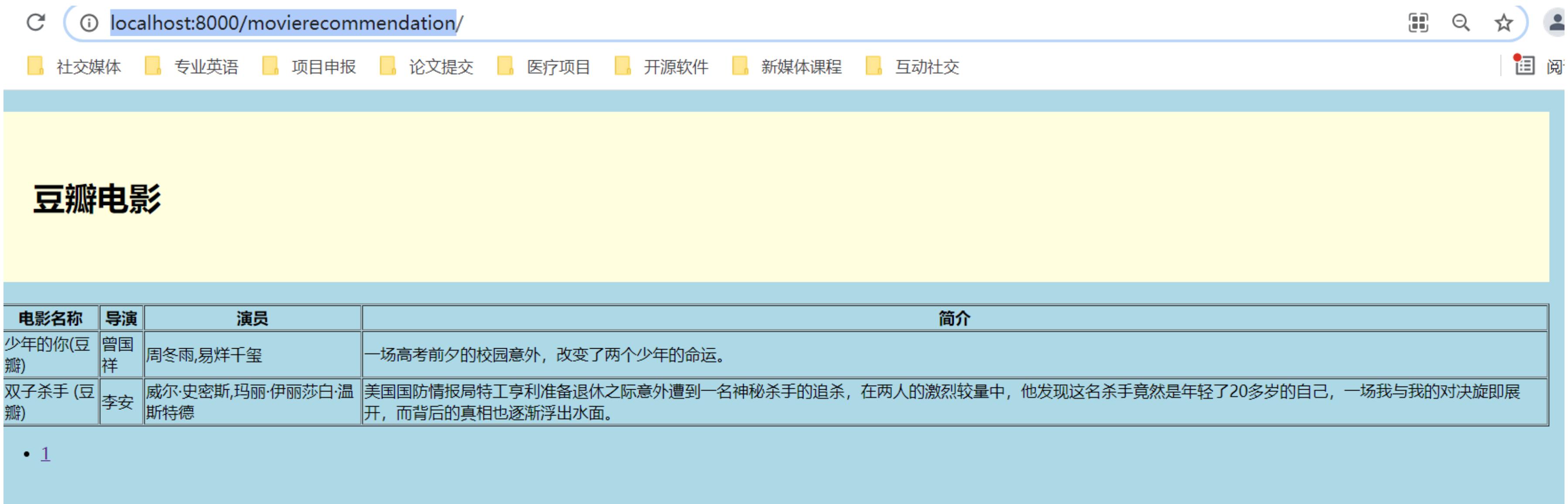


```
body {
    color: black;
    background: lightblue;
    padding: 0 5em;
    margin:0
}
h1 {
    padding: 2em 1em;
    background: lightyellow;
}
p {
    margin: 1em 0;
}
ul{
    list-style: none;
    margin: 0px;
    padding: 0px;
    width: auto;
}
ul li{
    float: right;
    margin-top: 10px;
    padding: 10px;
    border-style: solid;
    border: 1px solid;
}
```

Python快速建站

❑ 前端网站查看页面

- ❖ 浏览器输入“http://localhost:8000/movierecommendation/”



豆瓣电影

电影名称	导演	演员	简介
少年的你(豆瓣)	曾国祥	周冬雨,易烊千玺	一场高考前夕的校园意外，改变了两个少年的命运。
双子杀手 (豆瓣)	李安	威尔·史密斯,玛丽·伊丽莎白·温斯特德	美国国防情报局特工亨利准备退休之际意外遭到一名神秘杀手的追杀，在两人的激烈较量中，他发现这名杀手竟然是年轻了20多岁的自己，一场我与我的对决旋即展开，而背后的真相也逐渐浮出水面。

- 1

实验报告要求

- 请按照前面介绍的方法将给定的样例数据存进自己设计的数据表里
  - 请按照上述步骤完成豆瓣数据工程构建、数据导入和前端显示

    注意：所有命名都要以个人名字拼音作为前缀（防止被系统判定为雷同作业）

  - 下课之前每个人把报告各自上传到课程中心平台(请把“工程源代码+数据文件”打包命名为文件命名“学号+姓名+实验一.zip”)
- 本次实验不做为作业成绩，作为实验报告成绩的一部分

谢谢！

下一节实验内容：数据爬取