# 1.搭建django开发环境以及创建django项目和一些基本配置,参考: [Django学习笔记2-初识django1](django学习笔记2-初识django之项目创建和静态资源配置.docx)

# 2.这里我们来学习django提供模板语法,通过一个路由作为例子来学习

## 参考,django官方文档: <https://docs.djangoproject.com/zh-hans/5.2/topics/templates/>

## 1>打开mysite包里面的urls.py,然后提交一个tpl/路由映射

|  |
| --- |
|  |

## 2>然后在mySiteApp包里面的views.py文件里面添加一个tpl函数,在这个函数里面渲染一个叫做tpl.html的模板文件

|  |
| --- |
|  |

## 3>然后我们需要在mySiteApp包里面的templates文件夹里面新建一个tpl.html模板文件.为了方便对比学习,我们使用分割视图

|  |
| --- |
|  |

## 4>然后我们访问<http://127.0.0.1:8000/tpl/>,效果如下

|  |
| --- |
|  |

### 那么问题来了,这些数据都是写死的,如果我们需要把数据传递给模板,让他动态展示,我们该如何做?

## 5>我们可以在views里面的tpl函数里面定义一些数据或者到数据库中选择一些数据,如何需要用字典的方式来传递给模板,下面是最简单的传递方式, 然后我们就可以在页面中用{{变量名称(字典的key)}}来获取内容

|  |
| --- |
|  |

### 然后就可以在页面中看到我们的信息

|  |
| --- |
|  |

### 注意:在django中代码是写在{% %}里面的,获取变量的值使用2对花括号{{ var}}

## 6>我们还可以继续添加数据,如我们添加一个列表数据,还是需要把它作为字典的key的value添加进来,然后可以在模板中使用for语句来渲染,但是在模板中是不支持for range的,因为模板不能直接使用python的内置函数

|  |
| --- |
|  |

### 注意:指令需要有开始标签和结束标签,否则会报错,比如{% for ...%}需要和{%endfor%}配对,{%if...%}需要和{%endif%}配对

### 效果如下

|  |
| --- |
|  |

### 注意模板语法是很特殊的,比如你想通过索引来获取列表的元素,你需要使用.操作符

|  |
| --- |
|  |

## 7>我们还可以添加一个布尔类型的变量,然后在模板中进行条件渲染

|  |
| --- |
|  |

### 刷新页面,效果如下

|  |
| --- |
|  |

## 8>然后我们在tpl函数里面把isGood改为True

|  |
| --- |
|  |

### 刷新页面,效果如下

|  |
| --- |
|  |

### 说明if语句是由作用的.

## 9>还可以传递字典给模板,然后可以在模板中使用.操作符来获取属性

|  |
| --- |
|  |

### 刷新页面,效果如下

|  |
| --- |
|  |

## 10>另外字典也可以用for循环来遍历,可以只遍历keys或者values,还可以遍历items也就是键值对.下面我们就来遍历键值对

|  |
| --- |
|  |

### 效果是一样的

|  |
| --- |
|  |

## 11>还有一种情况,可能是一个字典列表,这种相对比较复杂需要使用嵌套循环,当然,也可以不嵌套

|  |
| --- |
|  |

### 刷新页面,效果如下:

|  |
| --- |
|  |

## 从这里,我们可以看到django的运行流程大概是这样子的

|  |
| --- |
|  |

# 3.小案例,用爬虫爬取数据然后渲染到模板页面

## 1>我们先在mysite包里面的urls.py里面添加一个movies/路由,把它和mySiteApp包里面的views.py里面的movies函数对应起来,然后在templates文件夹里面添加一个movies.html模板

|  |
| --- |
|  |
|  |
|  |

## 2>然后我们在movies函数里面用爬虫爬取豆瓣电影第一页的数据然后渲染到movies.html页面上,mevies函数的代码如下:

|  |
| --- |
| **def** movies(req):  **import** requests  **from** bs4 **import** BeautifulSoup   headers = {  **"User-Agent"**: **"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36"** }  url = **"https://movie.douban.com/top250"** *# 这里只是爬取一页* sub\_url = url + **f"?start=0"** res = reps = requests.get(sub\_url, headers=headers).text  *# print(reps.text)* soup = BeautifulSoup(res, **"html.parser"**)  *# print(soup)* titles = soup.findAll(**"span"**, attrs={**"class"**: **"title"**})  bds = soup.findAll(**"div"**, attrs={**"class"**: **"bd"**})  chinese\_titles = []  summ = []  **for** bd **in** bds:  *# print(bd.p.text)* summ.append(bd.p.text)  **for** title **in** titles:  *# print(title.string) # 不要标签,只需要标签里面的内容  # if not title.string.\_\_contains\_\_("/"): # 过滤掉英文标题,注意到英文标题都是以 / 开头 ok  # if title.string.find("/")==-1: # 过滤掉英文标题,注意到英文标题都是以 / 开头 ok,find方法找到了返回一个不是负数的结果,-1表示找不到,我们这里就需要它找不到* **if "/" not in** title.string: *# 过滤掉英文标题,注意到英文标题都是以 / 开头 也是ok的* chinese\_titles.append(title.string)  *# print(title.string)* summ.pop(0)  *# print(summ)  # print(chinese\_titles)* movies = dict(zip(chinese\_titles, summ))  print(movies)  **return** render(req, **"movies.html"**,{**"movies"**:movies}) |

## 3>然后在movies.html模板里面渲染数据

|  |
| --- |
|  |

### 然后我们就可以访问movies/路由,效果如下

|  |
| --- |
|  |