

晨测

1. 创建一个python3.5.2的虚拟环境
2. 启动django服务，并且能够使用127.0.0.1:9000 访问
3. url输入 127.0.0.1:8000/index/2019/hello发出请求，能够返回 2019年说了一句hello，写出路由视图
4. 画出MVT模型，并说明mvt都代表什么

回顾

创建虚拟环境

```
conda create -n MINGZI python=3.5.2
```

激活虚拟环境

```
activate DjangoPath
```

安装django

```
pip install django==2.2.1
```

创建工程

启动项目

pycharm打开并编写代码

路由

模板

1. 调用一个html页面

```
from django.template import Template,Context
def gethtml(request):
    html="""
    <html>
        <head>
        </head>
        <body>
        <h1>我是一个h1标签</h1>
        <h2>我是{{ name }}</h2>
        <a href =
    "https://baike.baidu.com/item/%E5%BE%B7%E6%80%80%E6%81%A9%C2%B7%E9%9F%A6%E5%BE%B7/5457042?fromtitle=%E9%9F%A6%E5%BE%B7&fromid=64575&fr=aladdin">
```

```

<img src='https://timgsa.baidu.com/timg?
image&quality=80&size=b9999_10000&sec=1568003957396&di=a9ef056162060cc8d0bb4e830
d80f1a4&imgtype=0&src=http%3A%2F%2F03imgmini.eastday.com%2Fmobile%2F20180917%2F6
2855583b6b5e607bdd7c8d54c34c8dc_wmk.jpeg' title = '这是韦德' alt='这个是韦德'>
</a>
<p>{{ content }}</p>
</body>
</html>
"""

#1. 构建模板结构
tempalte_obj = Template(html)
#2. 创建渲染模板
params = dict(name="韦德大爷",content="好几个总冠军")
content_obj = Context(params)
#3. 进行数据渲染
result = tempalte_obj.render(content_obj)
#4. 返回结果
# return HttpResponse(html)
return HttpResponse(result)

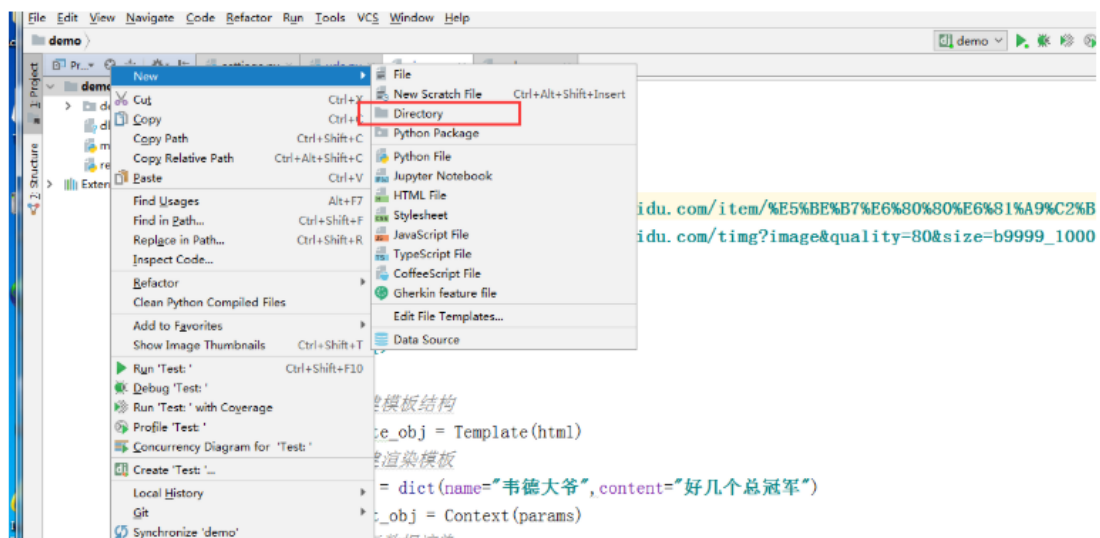
```

这种方式虽然能够实现调用一个html页面，并且能够将数据渲染上去，但是页面依然嵌套在python代码中，编写不方便。因此 不常用。

模板标签系统介绍

在做web开发，要给用户提供一个页面，页面包括静态页面+数据，两者结合起来就是完整的可视化的页面，django的模板系统支持这种功能，首先需要写一个静态页面（结构html，样式css，行为js）然后通过python的模板语法将数据渲染上去。

1. 创建一个templates目录



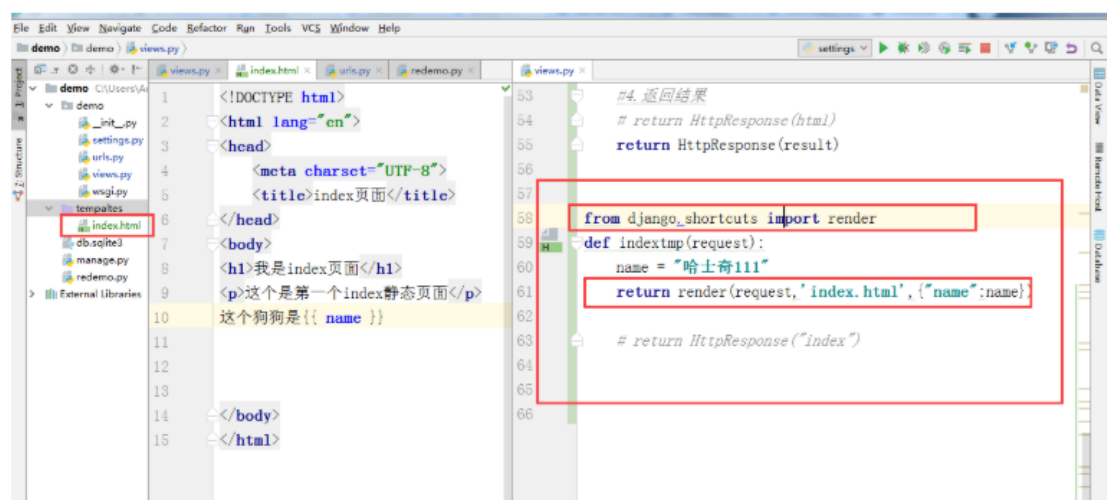
2. 配置模板



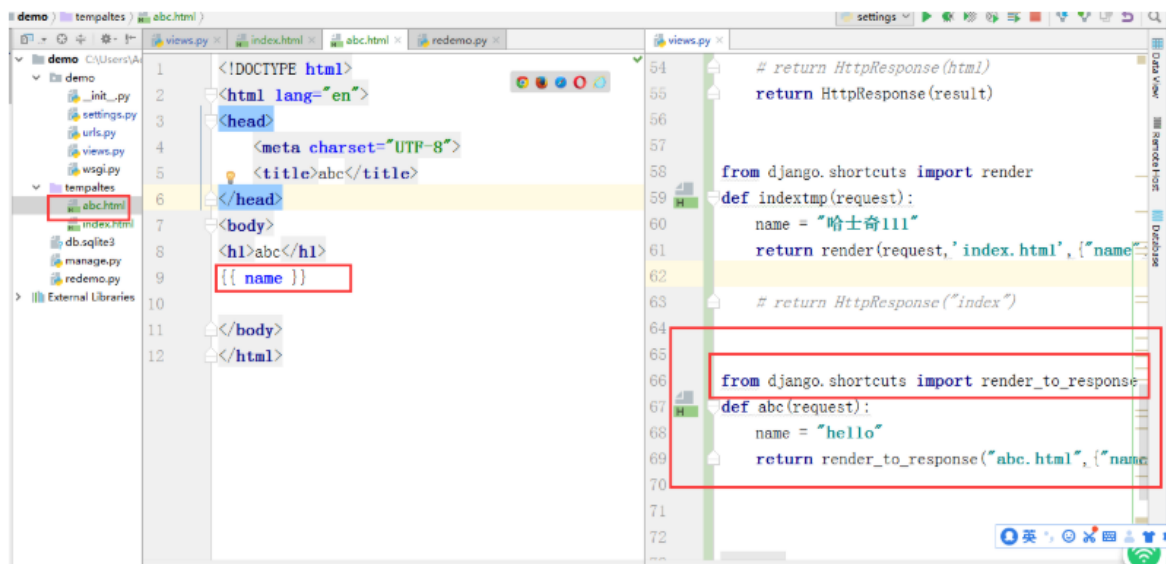
配置项介绍



1. 调用模板第一种方法



2. 调用模板第二种方法



3. 调用模板的第三种方法

```
9 # return render_to_response("abc.html", {"name": name})
10
11
12 from django.template.loader import get_template
13
14 def abc(request):
15     template = get_template("abc.html")
16     name = "hello"
17     result = template.render({'name': name})
18
19
20 return HttpResponse(result)
```

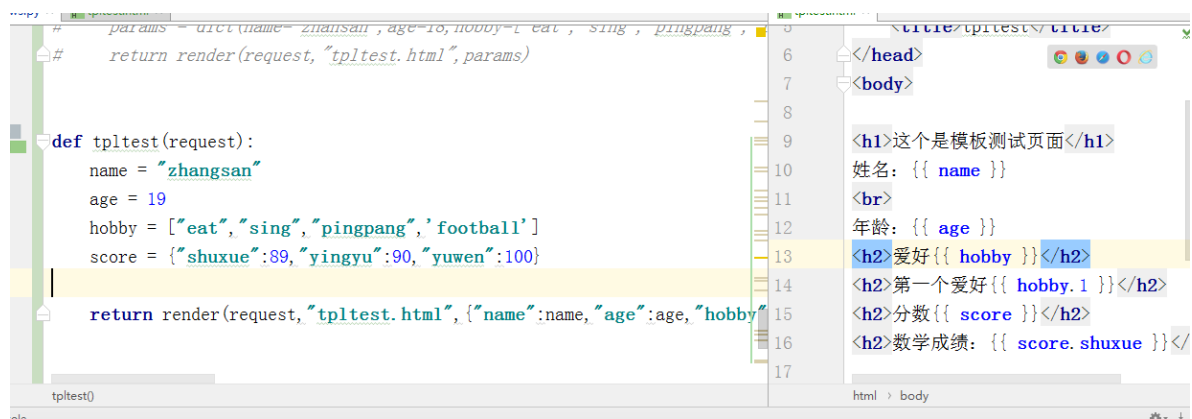
模板语法

1. 变量的使用

在模板中，如果想要使用后端传入的数据，就要用到模板语法去实现。

django提供的模板语法：变量的使用，控制语句，循环

django模板html文件中使用变量 语法：{{ }}



使用`locals()`方法，作用是将当前视图函数的所有局部变量返回



2. 控制语句

if else elif

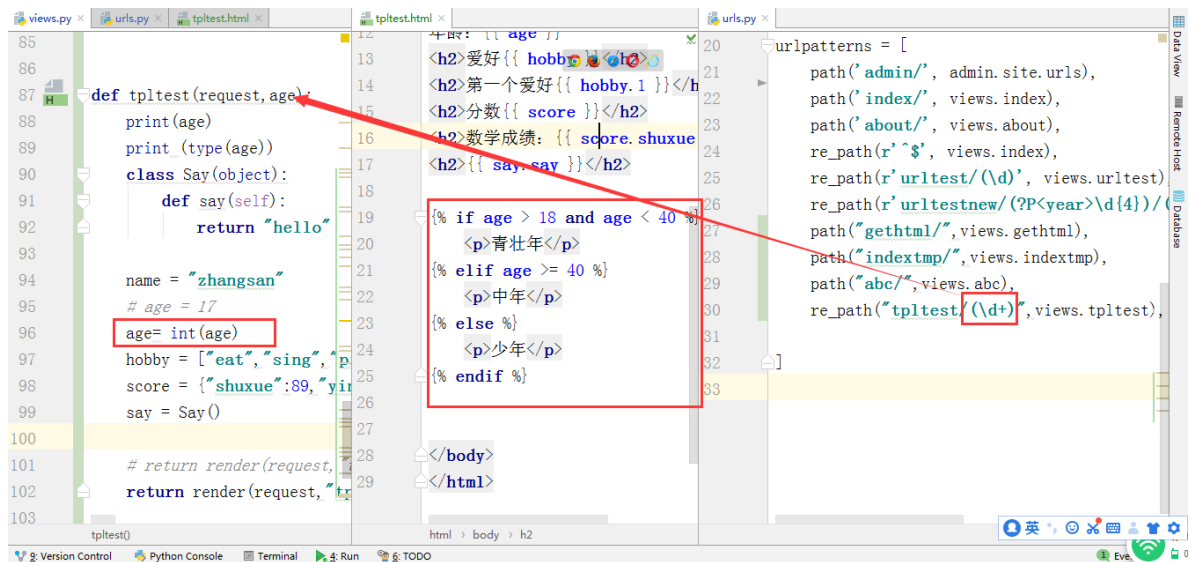
模板中使用控制语句，语法结构

```
{% if 条件 %}
```

```
{% elif 条件 %}
```

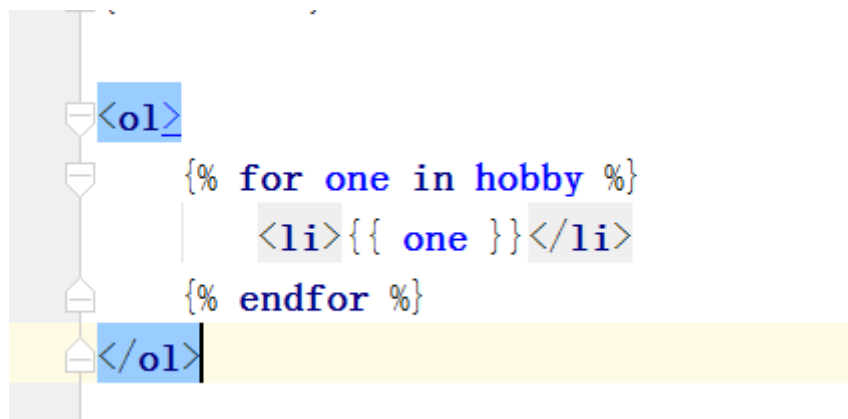
```
{% else %}
```

```
{% endif %}
```

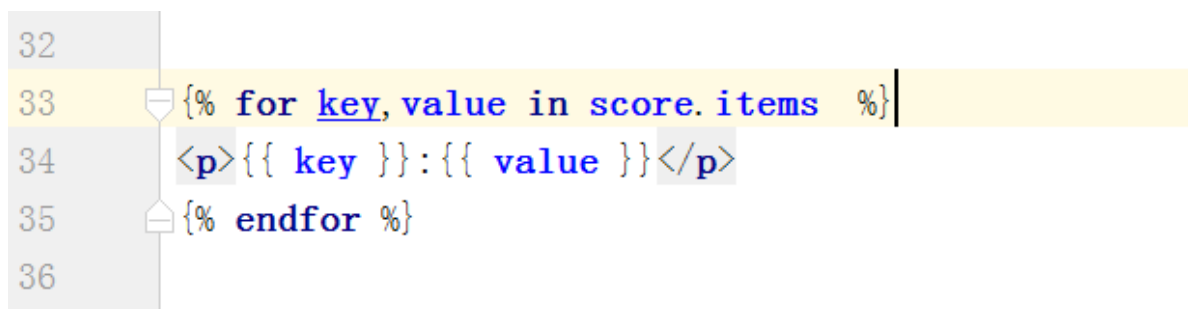


3. 循环语句

1. 遍历列表



2. 遍历字典



3. forloop

forloop是一个django自带的一个变量，作用于循环，记录循环的次数，可以对循环的内容进行控制，判断，通常和if连用

```
41      {% endtor %}
42
43  {% for one in hobby reversed %}
44      {% if forloop.first %}
45          <p style="color: red">{{ forloop.counter }}. {{ one }}</p>
46      {% elif forloop.last %}
47          <p style="color: blue">{{ forloop.counter }}. {{ one }}</p>
48      {% else %}
49          <p>{{ forloop.counter }}. {{ one }}</p>
50      {% endif %}
51
52  {% endfor %}
53
54
```