Django模板系统

本节我们继续使用《Django视图函数》一节中的“Hello_my_django”函数来完成相关知识的讲解。

```
from django.http import HttpResponse
def Hello_my_django(request):
    return HttpResponse('<html><body>Hello my Django</body></html>')
```

1. Django的模板系统

虽然上面的函数也能够顺利完成响应任务，但是我们可以看出这个视图函数的 HTML 代码写在了 Python 定义的函数中。我们先思考一下，如果用以上方法来定义视图函数的话，它是否具有可行性呢？

有以下两点值得我们思考：

- 我们知道前端页面需要经常改动。比如，某个电商网站到了双十一搞活动的时候，需要对前端页面做大量的修改，如果将页面放到视图函数中，那么当对前端页面修改的时候，也会使得视图函数发生变化。
- 从 MTV 设计模式的角度出发，视图层是实现业务逻辑的，在视图层编写代码和编写 HTML 页面是两项可以相互独立的工作，就像公司有开发小组和 UI 小组，它们分别负责不同类型的工作，所以我们为什么不考虑把它们分开呢？如果放到一起，就会增加视图层的复杂度，给程序员维护代码带来困难！

那么 Django 是如何实现视图函数与 HTML 代码解耦的呢？这就引出我们本节要讲解的知识——模板系统。

2. 模板系统的应用

在 Django 中我们把“模板”称之为 Template，它的存在使得 HTML 和 View 视图层实现了解耦。 在《Django MTV和MVC的区别》一文中也提到过 Template，它是设计模式中的 T 层，那么它在 Djang 中又是如何应用的呢？

其实 T 层应用是这样实现，当创建好一个 Django 项目后，我们在项目的同级目录下创建一个名为 templates 文件夹，对它进行简单的

配置后，这个文件夹将被 Django 自动识别。我们可以简单的理解为：文件夹就好比我们所说的 T 层，然而其复杂的实现过程由 Django 框架本身来实现的，所以我们无需关心内部细节。

下面我们对 Hello_my_django 函数进行一下改造，在 templates 文件中新建一个 HTML 文件，并且将此文件命名为 hello.html，然后在此文件中书写我们的 HTML 代码，如下所示：

写HTM代码：

```
<html><body>{{vaule}}</body></html>
```

写视图函数：

```
from django.shortcuts import render
def hello_my_django(request):
    return render(request,"hello.html",{"vaule":"hello my Django"})
```

看完上述代码，你可会有些不理解，这属于正常现象，因为我们还有许多的知识未涉及到，继续看我们教程，你会很快理解它。对于上述代码，我们先通俗易懂的讲解一下。

1) 模板传参

hello.html 文件中的 {{vaule}} 是一个模板的变量，视图函数必须把数据形成字典的形式才可以传递给模板，这就是“模板传参”。

2) render方法

render 是 View 层加载模板的一种方式，它封装在 django.shortcuts 模块中，render 方法使用起来非常方便，它首先加载模板，然后将加载完成的模板响应给浏览器。

本节重点讲解了模板层与视图层的组合使用，也涉及了一些其他的知识，比如“模板的传参”和“render 方法”加载模板。当然，后续章节对这些知识点还会深入讲解。