

Python flask模版介绍

flask模板

在动态web程序中，视图函数返回的HTML数据往往需要根据相应的变量（比如查询参数）动态生成。

当HTML代码保存到单独的文件中时，我们没法再使用字符串格式化或拼接字符串的当时在HTML代码中插入变量，这时我们需要使用模板引擎（template engine）。借助模板引擎，我们可以再HTML文件中使用特殊的语法来标记变量，这类包含固定内容和动态部分的可重用文件称为模板（template）。

模板引擎的作用就是读取并执行模板中的特殊语法标记，并根据传入的数据将变量替换为实际值，输出最终的HTML页面，这个过程被称为渲染（rendering）。

Flask默认使用的模板引擎是jinja2，他是一个功能齐全的python模板引擎，除了设置变量，还允许我们在模板中添加if判断，执行for迭代，调整函数等，以各种方式 控制模板的输出。

对于jinja2来说，模板可以是任何格式的纯文本文件，比如HTML、XML、CSV等。

视图函数的主要作用是生成请求的响应

处理业务逻辑

返回响应内容

把业务逻辑和表现内容放在一起，会增加代码的复杂度和维护成本。

```
def hello():  
    return "hello benben"  
  
def index():  
    return render_template("index.html")  
  
def success(name):  
    return 'welcome %s' % name
```

Flask模板

视图函数的主要作用是生成请求的响应

处理业务逻辑

返回响应内容

把业务逻辑和表现内容放在一起，会增加代码的复杂度和维护成本。

使用模板：使用静态的页面html展示动态的内容

模板是一个响应文本的文件，其中占用符（变量）表示动态部分，告诉模板引擎其具体的值需要从使用的数据中获取。

使用真实值替换变量，再返回最终得到的字符串，这个过程称为“渲染”。

Flask使用Jinja2这个模板引擎来渲染模板。

如何理解静态页面html展示动态的内容，简单来说就是一个静态的页面，你在url输入“zhangsang”，页面就显示“zhangsang”

以下就是使用模版想要达到的效果，**视图函数可以看做后端，模版可以看做前段**（便于理解）



接下来了解flask模版中的两个函数

render_template

加载html文件。默认文件路径在templates目录下。

```
from flask import Flask,render_template
app = Flask(__name__)

@app.route('/')
def index():
    return render_template("index.html")

if __name__ == '__main__':
    app.run()
```

在templates目录下创建index.html文件

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  模板html展示页面
</body>
</html>
```

render_template()函数是flask函数，它从模版文件夹templates中呈现给定的模板上下文。

⚠ render_template()函数需要调用flask包

render_template

加载html文件。默认文件路径在templates目录下。

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route('/')
def index():
    return render_template("index.html")

if __name__ == '__main__':
    app.run()
```

127.0.0.1:5000

在templates目录下创建index.html文件

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  模板html展示页面
</body>
</html>
```

