Python Flask从入门到不放弃

按照惯例我们会在这里插入一张图片作为封面

高清无码pdf见

链接：https://pan.baidu.com/s/1Dpg3G44Ytp5EwGg9CuoI3g

提取码：gc22

前情回顾

上次我们使用Flask中的路由来创建了两个可以被同时访问的页面

可是,一个完整的网站当然不能只返回给用户一句"HelloWord"

灵魂拷问

怎样才能返回一个好看的页面呢?

难道要这么写么?

@app.route('/')

def hello\_world():

    return """

        <html>

        <head>

          <meta charset="UTF-8">

          <title>这是一个页面</title>

        </head>

        <body>

          <h1>Hello</h1>

          <h2>难道我们要以这样的方式来返回页面么?</h2>

        </body>

        </html>

        """

虽然这个方法是可取的,但是如果我有一个上百上千行的html代码的话,你还打算这样做吗?

我们在网络上去访问一个地址时,通常情况下他会给我们返回一个带有各种信息的html文档,因为我们的程序是动态的,他可能会根据不同的情况展示不同的状态.比如说不同用户登录之后展示出来各自不同的信息,所以页面需要在用户访问的时候通过查找的数据来自动生成.

我们可以把一些带有变量和逻辑运算的html或者其他格式的文件叫做模板

程序将这些变量的内容替换和逻辑计算的过程叫做渲染

来完成这个工作的程序叫做模板引擎

在flask中,可以使用诸多的模板引擎,但是flask默认使用的是jinja2模板引擎

如何编写模板?

根据flask的默认设置,Flask会在程序实例所在模块的同级目录的templates文件夹中去寻找模板,这也是上次我们要搞明白为什么要有\_\_name\_\_的原因

目前我们的程序存在app.py文件中,所有我们需要在app.py的同级目录来创建templates目录

为了操作方便,下面的操作会在pycharm软件中完成

现在有了模板目录,我们再来创建模板文件比如index.html

保存之后,我们就拥有了一个html模板

如何将模板展示到页面上?

我们需要从flask模块中导入render\_template函数

使用render\_template来渲染我们的模板文件并且返回页面

@app.route('/')

def hello\_world():

    return render\_template("index.html")

重新运行服务器进行测试

模板已经成功的渲染到了浏览器中

但是,这里还是展示的静态数据,那我们如何传递数据到模板中呢?

不要着急,我们先来看一下模板的基本语法是如何构成的

{{ ... }} 用来标记变量。

{% ... %} 用来标记语句，比如 if 语句，for 语句等。

{# ... #} 用来写注释。

先来看看一个变量是怎么跑到页面上去的

<body>

    {{ name }}

</body>

这里我们就在模板中声明了一个变量他的名字叫name

那我们如何把这个变量的值传递过去呢?

在render\_template函数后面传递参数

@app.route('/')

def hello\_world():

    return render\_template("index.html", name="万能北极熊")

重启服务器查看

很显然,这个变量已经成功的传递过去了

你也可以直接把对应类型的数据写到模板中

<body>

    {{ name }}<br>

    {{ 18 }} 岁<br>

    坐标: {{ "山东省" }}<br>

    {{ ["甚","至","一","个","列","表"] }}<br>

</body>

我们再来看看其他语法是怎么使用的

{% ... %}

你可以在这里面写if或者for语句

if

<body>

    {% if True %}

        {{ "真" }}

    {% endif %}

    <br/>

    {% if False %}

        {{ "假" }}

    {% endif %}

</body>

输出结果

可以看到,当if为真的时候就会执行if和endif中间的语句

你也可以在中间写else或者elif

for

for循环主要的作用就是渲染多个一样格式的内容,比如一个名字的列表

<body>

    <br/>

    {% for name in ["大熊","熊二","小熊"] %}

        {{ name }}<br>

    {% endfor %}

</body>

穿插在for和endfor中间的html标签也会被重复的渲染

{# ... #}

这个标签中间的内容是注释内容,既不会被模板引擎所渲染也不会包含在html标签中

<body>

    <br/>

    {% for name in ["大熊","熊二","小熊"] %}

        {{ name }}<br>

    {% endfor %}

    <!-- HTML的注释 -->

    {# 这写的是注释 #}

</body>

过滤器

为了方便对变量进行处理，Jinja2 提供了一些过滤器，语法形式如下：

{{  变量|过滤器  }}

怎样来使用过滤器?

<body>

    {{ ["熊大","熊二","吉吉","毛毛"] | length }}

</body>

这是一个取长度的过滤器,类似于len()函数

jinja2中都有哪些过滤器?

这里列举了常用的一些过滤器

safe:      渲染时值不转义

capitialize:   把值的首字母转换成大写，其他子母转换为小写

lower:    把值转换成小写形式

upper:    把值转换成大写形式

title:    把值中每个单词的首字母都转换成大写

trim:      把值的首尾空格去掉

striptags:    渲染之前把值中所有的HTML标签都删掉

join:      拼接多个值为字符串

replace:      替换字符串的值

round:  默认对数字进行四舍五入，也可以用参数进行控制

int:       把值转换成整型

你也可以自定义一个过滤器,相关内容可以自行查找资料

了解了这些基础的内容我们来做个小实例

模仿bilibi主页上的个人信息卡片(的文字)

先来准备前端页面(奇丑无比)

<body>

    <p>

        {{ name }}

    </p>

    <p>

        硬币:{{ coin\_num }}

    </p>

    <p>

         B币:{{ B\_coin\_num }}

    </p>

    <p>

        手机绑定状态: {% if phone %} {{ phone }} {% else %} {{ "未绑定手机" }} {% endif %}

    </p>

    <p>

        邮箱绑定状态: {% if emall %} {{ emall }} {% else %} {{ "未绑定邮箱" }} {% endif %}

    </p>

    <p>

        关注:{{ Subscribe\_num }}

    </p>

    <p>

        粉丝数:{{ fans\_num }}

    </p>

    <p>

        动态: {{ dynamic\_num }}

    </p>

</body>

虽然稍微有些捡漏...

现在我们来模拟一些数据来传入这个模板

from flask import Flask

from flask import render\_template

app = Flask(\_\_name\_\_)

@app.route('/')

def hello\_world():

    user = {

        "name": "万能北极熊",

        "coin\_num": 600,

        "B\_coin\_num": 0,

        "phone": None,

        "emall": "2960053329@qq.com",

        "Subscribe\_num":140,

        "fans\_num": 9999,

        'dynamic\_num': 255

    }

    return render\_template("index.html",

                           name=user.get("name"),

                           coin\_num=user.get("coin\_num"),

                           B\_coin\_num=user.get("B\_coin\_num"),

                           phone=user.get("phone"),

                           emall=user.get("emall"),

                           Subscribe\_num=user.get("Subscribe\_num"),

                           fans\_num=user.get("fans\_num"),

                           dynamic\_num=user.get("dynamic\_num")

                           )

if \_\_name\_\_ == '\_\_main\_\_':

    app.run()

这样我们的数据就传入模板了

将参数一个个传递进去并不是一个聪明的做法,我们应该传入一个对象或者一个字典来统一调用数据

   return render\_template("index.html", user=user)

<body>

    <p>

        {{ user.get("name") }}

    </p>

    <p>

        硬币:{{ user.get("coin\_num") }}

    </p>

    <p>

         B币:{{ user.get("B\_coin\_num") }}

    </p>

    <p>

        手机绑定状态: {% if user.get("phone") %} {{ user.get("phone") }} {% else %} {{ "未绑定手机" }} {% endif %}

    </p>

    <p>

        邮箱绑定状态: {% if user.get("emall") %} {{ user.get("emall") }} {% else %} {{ "未绑定邮箱" }} {% endif %}

    </p>

    <p>

        关注:{{ user.get("Subscribe\_num") }}

    </p>

    <p>

        粉丝数:{{ user.get("fans\_num") }}

    </p>

    <p>

        动态: {{ user.get("dynamic\_num") }}

    </p>

</body>

关于模板的使用还有很多很多的东西需要你来了解,这里我们就先介绍到这里,更多新奇的内容还是要用心去发现多查阅资料

那现在,我们就先刹刹车休息一下消化一下内容

(疯狂暗示,暗示,暗示)

我们,下次见! 作者：热心的裴同学 https://www.bilibili.com/read/cv5129517/ 出处：bilibili