**WTForms介绍和基本使用：**

**1.WTForms介绍：**

这个插件库主要有两个作用。

第一个是做表单验证，将用户提交上来的数据进行验证是否符合系统要求。

第二个是做模版渲染。 （了解即可）

官网：<https://wtforms.readthedocs.io/en/latest/index.html>

Flask-WTF是简化了WTForms操作的一个第三方库。WTForms表单的两个主要功能是验证用户提交数据的合法性以及渲染模板。而Flask-WTF还包括一些其他的功能：CSRF保护，文件上传等。

安装Flask-WTF默认也会安装WTForms，因此使用以下命令来安装Flask-WTF和WTForms:

|  |
| --- |
| pip install flask-wtf |

**2.WTForms做表单验证的基本使用：**

1. 自定义一个表单类，继承自wtforms.Form类。

2. 定义好需要验证的字段，字段的名字必须和模版中那些需要验证的input标签的name属性值保持一致。

3. 在需要验证的字段上，需要指定好具体的数据类型。

4. 在相关的字段上，指定验证器。

5. 以后在视图函数中，只需要使用这个表单类的对象，并且把需要验证的数据，也就是request.form传给这个表单类，再调用表单类对象.validate()方法进行，如果返回True，那么代表用户输入的数据都是符合格式要求的，Flase则代表用户输入的数据是有问题的。如果验证失败了，那么可以通过表单类对象.errors来获取具体的错误信息。

**如注册页面register.html：**

|  |
| --- |
| <!DOCTYPE **html**>  <**html lang="en"**>  <**head**>  <**meta charset="UTF-8"**>  <**title**>某系统注册页面</**title**>  </**head**>  <**body**>  <**form action="/register/" method="post"**>  <**table**>  <**tr**>  <**th**>用户名：</**th**>  <**td**><**input type="text" name="uname"**></**td**>  </**tr**>  <**tr**>  <**th**>密码：</**th**>  <**td**><**input type="password" name="pwd"**></**td**>  </**tr**>  <**tr**>  <**th**>确认密码：</**th**>  <**td**><**input type="password" name="pwd2"**></**td**>  </**tr**>  <**tr**>  <**td**></**td**>  <**td**><**input type="submit" value="注册"**></**td**>  </**tr**>  </**table**>  </**form**>  </**body**>  </**html**> |

**如app.py文件：**

|  |
| --- |
| **from** flask **import** Flask,request,render\_template  **from** wtforms **import** Form,StringField  **from** wtforms.validators **import** Length,EqualTo  app = Flask(\_\_name\_\_)  **class** RegisterForm(Form):  uname =StringField(validators=[Length(min=2,max=15,message=**'用户名长度必须在2-15之间'**)])  pwd = StringField(validators=[Length(min=6,max=12)])  pwd2 = StringField(validators=[Length(min=6,max=12),EqualTo(**"pwd"**)])  @app.route(**'/'**)  **def** hello\_world():  **return** render\_template(**"register.html"**)  @app.route(**'/register/'**,methods=[**'GET'**,**'POST'**])  **def** register():  **if** request.method == **'GET'**:  **return** render\_template(**'register.html'**)  **else**:  form = RegisterForm(request.form)  **if** form.validate(): *#验证 要么ok 要么no*  **return "验证通过"**  **else**:  print(form.errors)  **return "数据验证通不过"**  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |