WTForms笔记：
这个库一般有两个作用。第一个就是做表单验证，把用户提交上来的数据进行验证是否合法。第二个就是做模版渲染。

做表单验证：
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. 以后在视图中，就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，以后调用form.validate()方法，如果返回True，那么代表用户输入的数据都是合法的，否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form.errors来获取具体的错误信息。
示例代码如下：
ReistForm类的代码：
```
class RegistForm(Form):
    username = StringField(validators=[Length(min=3,max=10,message='用户名长度必须在3到10位之间')])
    password = StringField(validators=[Length(min=6,max=10)])
    password_repeat = StringField(validators=[Length(min=6,max=10),EqualTo("password")])
```

视图函数中的代码：
```
form = RegistForm(request.form)
if form.validate():
    return "success"
else:
    print(form.errors)
    return "fail"
```

常用的验证器：
数据发送过来，经过表单验证，因此需要验证器来进行验证，以下对一些常用的内置验证器进行讲解：

1. Email：验证上传的数据是否为邮箱。
2. EqualTo：验证上传的数据是否和另外一个字段相等，常用的就是密码和确认密码两个字段是否相等。
3. InputRequir：原始数据的需要验证。如果不是特殊情况，应该使用InputRequired。
3. Length：长度限制，有min和max两个值进行限制。
4. NumberRange：数字的区间，有min和max两个值限制，如果处在这两个数字之间则满足。
5. Regexp：自定义正则表达式。
6. URL：必须要是URL的形式。
7. UUID：验证UUID。

自定义验证器：
如果想要对表单中的某个字段进行更细化的验证，那么可以针对这个字段进行单独的验证。步骤如下：
1. 定义一个方法，方法的名字规则是：validate_字段名(self,filed)。
2. 在方法中，使用field.data可以获取到这个字段的具体的值。
3. 如果数据满足条件，那么可以什么都不做。如果验证失败，那么应该抛出一个wtforms.validators.ValidationError的异常，并且把验证失败的信息传到这个异常类中。
示例代码：
```
# 注册表单类
class RegisterForm(Form):
    username = StringField(validators=[Length(min=3, max=10, message=u'用户名长度为3-10个字符')])
    password1 = StringField(validators=[Length(min=3, max=10, message=u'密码长度为3-10个字符')])
    password2 = StringField(validators=[EqualTo('password1', message=u'两次输入密码不同')])
    code = StringField(validators=[Length(min=0, max=10, message=u'注册码长度为3-10个字符')])

    def validate_code(self, filed):
        if filed.data != 'L!EA6OJQ%8':
            raise ValidationError(u'邀请码错误')
```

使用渲染模板

WTForms可以用来渲染模板, 但用处不大, 了解即可