



# 数据可视化 Python Flask框架表单

李春芳

理工学部 计算机学院

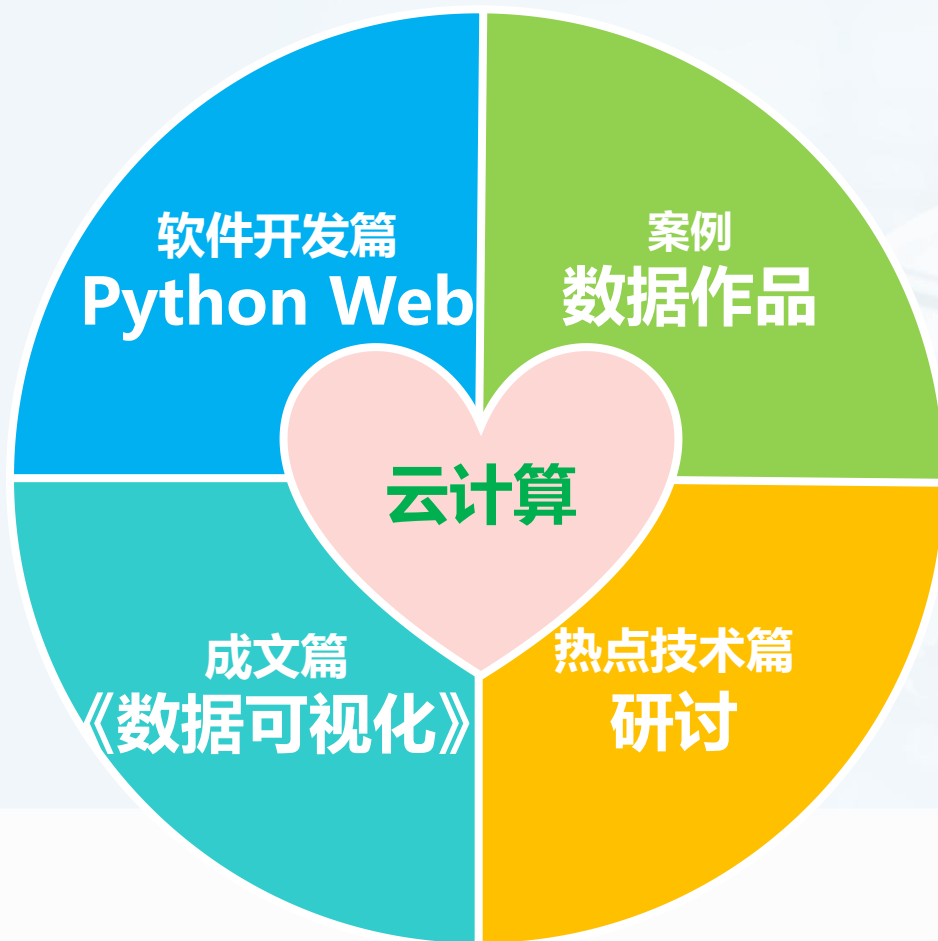
2017-7-6



中国传媒大学  
COMMUNICATION UNIVERSITY OF CHINA

# 实践课程概述

课程目标：软件开发中的数据可视化



# 第四章 表单

*Flask-wtf*

# Flask-WTF处理表单

是WTForms的包的包装，方便集成到Flask中

Pip install flask-wtf

http://127.0.0.1:5000/

网页无法访问 × The world's leading software × Flasky × +

Flasky Home

Hello, CS!

What is your name?

Submit

# 跨站请求伪造保护

- ◆ **Flask-wtf**能保护表单免收跨站请求伪造的攻击
  - ◆ 恶意网站把请求发送到被攻击者已经登录的其他网站时就会发生**CSRF** (**Cross-Site Request Forgery**)
  - ◆ 需要一个**密钥**，**Flask-wtf**使用密钥生成加密令牌，用令牌验证请求表单数据的真伪。
- 
- ◆ `app = Flask(__name__)`
  - ◆ `app.config['SECRET_KEY'] = 'hard to guess string'`



# Flask-wtf表单类

- ◆ Web表单继承自Form类，定义了表单中的一组字段，每个字段都用对象表示
- ◆ 字段对象可附属于一个或多个验证函数
- ◆ 验证函数验证用户输入是否符合要求

- ◆ `from flask_wtf import FlaskForm`
- ◆ `from wtforms import StringField, SubmitField`
- ◆ `from wtforms.validators import Required`
- ◆ `class NameForm(FlaskForm):`
  - ◆ `name = StringField('What is your name?', validators=[Required()])`
  - ◆ `submit = SubmitField('Submit')`

```
<input class="form-control" id="name" name="name" required type="text" value="">  
<input class="btn btn-default" id="submit" name="submit" type="submit" value="Submit">
```



# WTFForms支持的HTML字段

表4-1 WTForms支持的HTML标准字段

字段类型	说 明
StringField	文本字段
TextAreaField	多行文本字段
PasswordField	密码文本字段
HiddenField	隐藏文本字段
DateTimeField	文本字段，值为 <code>datetime.date</code> 格式
DateTimeField	文本字段，值为 <code>datetime.datetime</code> 格式
IntegerField	文本字段，值为整数
DecimalField	文本字段，值为 <code>decimal.Decimal</code>
FloatField	文本字段，值为浮点数
BooleanField	复选框，值为 <code>True</code> 和 <code>False</code>
RadioField	一组单选框
SelectField	下拉列表
SelectMultipleField	下拉列表，可选择多个值
FileField	文件上传字段
SubmitField	表单提交按钮
FormField	把表单作为字段嵌入另一个表单
FieldList	一组指定类型的字段

# WTFORMS验证函数

验证函数	说 明
Email	验证电子邮件地址
EqualTo	比较两个字段的值；常用于要求输入两次密码进行确认的情况
IPAddress	验证 IPv4 网络地址
Length	验证输入字符串的长度
NumberRange	验证输入的值在数字范围内
Optional	无输入值时跳过其他验证函数
Required	确保字段中有数据
Regexp	使用正则表达式验证输入值
URL	验证 URL
AnyOf	确保输入值在可选值列表中
NoneOf	确保输入值不在可选值列表中



# 把表单渲染成HTML

- ◆ 表单字段是可以调用的，渲染成HTML
- ◆ 提交表单验证`form.validate_on_submit()`后赋值

◆ Git checkout 4a

```
◆ @app.route('/', methods=['GET', 'POST'])
◆ def index():
◆     name = None
◆     form = NameForm()
◆     if form.validate_on_submit():
◆         name = form.name.data
◆         form.name.data = ''
◆     return render_template('index.html',
form=form, name=name)
```

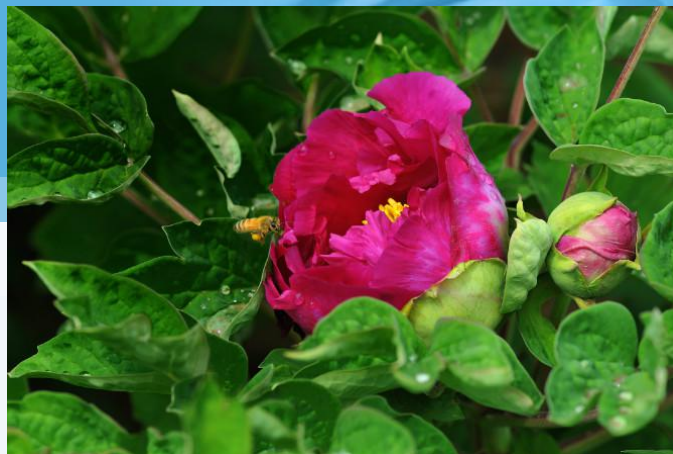
# 错误处理的模板

```
◆ @app.errorhandler(404)
◆ def page_not_found(e):
◆     return render_template('404.html'),
404
```

```
◆ @app.errorhandler(500)
◆ def internal_server_error(e):
◆     return render_template('500.html'),
500
```



# 谢谢



## 期待各位的佳作