

django 表单

- 表单
 - 概述
 - 一组HTML交互元素用于 提交信息到服务器
 - HTML, 表单
 - 表单呈现
 - `action='数据提交URL'`
 - `method='GET/POST'`
 - `{% csrf_token %}` 防止跨站请求伪造
 - 服务器数据处理
 - `request.method` 提交方法
 - `request.GET` 包含GET提交数据字典表(类似)
 - `request.POST` 包含POST提交数据字典表(类似)
 - django.forms表单处理
 - 概述
 - 准备及构造数据以备呈现
 - 为数据创建HTML表单
 - 接收并处理客户端提交的数据
 - 表单Form类
 - `django.forms.Form`
 - 构造选项
 - `auto_id = True/False` 是否为元素添加id 属性
 - `initial = {'field': 'value'}` 为元素初始化值
 - 绑定值
 - `f = xxxForm({})`
 - `.is_bound` 是否有绑定值
 - `.has_changed()` 绑定值与初始化值是否有变化
 - `.changed_data` 改变值的字段列表
 - `.cleaned_data` 清理后的实际值
 - HTML呈现
 - `.as_table()`
 - `.as_p()`
 - `.as_ul()`
 - 表单字段
 - `django.forms.Field`
 - 核心属性/方法
 - `label='标签文本'`
 - `label_suffix='标签后缀'`
 - `initial='字段初始值'`
 - `help_text='帮助文本'`
 - `required=是否必填`
 - `widget='指定HTML元素'`

- `.clean('value')` 为字段验证, 清理填充值
- 表单验证
 - 表单验证的有效及安全性
 - 验证 客户端/服务器端
- 验证逻辑
 - 表单字段类型
 - 字段属性
 - `validators=[django.core.validators.XXX()]`
 - 触发
 - `.is_valid()`
 - `.errors`
 - `.full_clean()`
 - 异常 `ValidationError`
 - 步骤
 - `Field.to_python()` 转换至对应Python类型
 - `Field.validate()`处理验证
 - `Field.run_validators()`合并错误到单一`ValidationError`
 - `Field.clean()` 合并执行前述步骤, 返回清理后数据至表单 `cleaned_data` 属性
 - `Form.clean_<fieldname>()` 特定字段值清理
 - 自定义验证
 - 自定义函数验证值
 - 不符合验证条件抛出 `django.core.exceptions.ValidationError` 异常
 - `Field(validators=[自定义验证函数,])`
- HTML端表单验证及提交
 - HTML
 - `{{ form.errors }}`
 - `{{ form.field.errors }}`
 - 服务器端
 - `if request.method == 'POST':`
 - `form = XXXForm(request.POST)`
 - `form.is_valid()`
- 表单数据处理保存
 - 常规字段处理

以上内容整理于 [幕布](#)