

validate

1、引入jQuery和jquery.validate.js

```
<script src="../../static/jquery-3.5.1.min.js"></script>
<script src="../../static/jquery.validate.js"></script>
```

2、HTML设置

```
<form action="server.jsp" method="post" id="form1">
  <table>
    <tr>
      <td>用户名</td>
      <td><input type="text" name="username"/></td>
    </tr>
    <tr>
      <td>密码</td>
      <td><input type="password" name="password" id="pwd"/></td>
    </tr>
    <tr>
      <td>确认密码</td>
      <td><input type="password" name="repwd"/></td>
    </tr>
    <tr>
      <td>电子邮箱</td>
      <td><input type="text" name="mail"/></td>
    </tr>
    <tr>
      <td>手机</td>
      <td><input type="text" name="mobile"/></td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit" value="开始注册"/>
      </td>
    </tr>
  </table>
</form>
```

3、调用validate

```
var myValidate=$('#form1').validate({
  // 验证规则
  rules:{
    username:{
      required:true,
      minlength:2,
      maxlength:10,
      remote:{
        url:"http://127.0.0.1:8000/checkusername",
        type:"get"
      }
    },
  },
});
```

```

        mail:{
            required:true,
            email:true
        },
        password:{
            required:true
        },
        repwd:{
            required:true,
            equalTo:"#pwd"
        },
        mobile:{
            required:true,
            mobile:true
        }
    },
    // 错误提示信息
    messages:{
        username:{
            required:'用户名不能为空',
            minlength:'用户名长度至少需要2个字符',
            maxlength:'用户名长度最多10个字符',
            remote:'用户已被占用'
        },
        mail:{
            required:'邮箱不能为空',
            email:'邮箱格式不正确'
        },
        password:{
            required:'密码不能为空'
        },
        repwd:{
            required:'确认密码不能为空',
            equalTo:'两次密码输入不一致'
        },
        mobile:{
            required:'手机号不能为空'
        }
    },
    //提示样式
    //设置提示标签
    errorElement:'div',
    //配置成功提示
    success:function(div){
        div.addClass('ok').html('验证通过');
    },
    validClass:'ok'
})

// 自定义手机号码验证
jQuery.validator.addMethod("mobile", function(value, element) {
    var mobileReg = /^1[34578][0-9]{9}$/;
    return this.optional(element) || (mobileReg.test(value));
}, "手机号码格式不正确");

// 提交时验证
$('#form1').submit(function(){
    //在提交之前判断验证是否全部通过

```

```
        if(!myValidate.form()){  
            return false;  
        }  
    })  
})
```

验证规则：

验证规则	说明
required:true	必填
remote:{url:"",type:"get"}	服务端校验
minlength:5	最小长度
maxlength:18	最大长度
rangelength:[5,18]	长度范围
min:1	最小值
max:100	最大值
range:[1,100]	取值范围
email:true	邮箱格式
url:true	url地址格式
number:true	数字
equalTo:"#pwd"	与id为pwd的元素值相等

自定义验证规则

```
// 邮政编码验证  
jQuery.validator.addMethod("postCode", function(value, element) {  
    var codeReg= /^[0-9]{6}$/;  
    return this.optional(element) || (codeReg.test(value));  
}, "请正确填写您的邮政编码");  
  
// 手机号码验证  
jQuery.validator.addMethod("mobile", function(value, element) {  
    var mobileReg = /^[1-9][0-9]{9}$/;  
    return this.optional(element) || (mobileReg.test(value));  
}, "请正确填写您的手机号码");
```

自定义提示效果

```
$('#form1').validate({  
    //设置提示标签  
    errorElement: 'div',  
    //配置成功提示  
    success: function(div){  
        div.addClass('ok').html('验证通过');  
    },  
    validClass: 'ok'  
})
```

```
table td {  
    position: relative;  
    line-height: 45px;  
    padding-left: 10px;  
}  
  
table td:first-child {  
    width: 5em;  
    text-align-last: justify;  
}  
  
table input {  
    padding: 5px;  
    border: 1px solid #ccc;  
    border-radius: 2px;  
    outline: none;  
}  
  
div.error,  
div.ok {  
    position: absolute;  
    top: 30px;  
    padding-left: 20px;  
    font-size: 12px;  
    font-weight: bold;  
}  
  
div.error {  
    background: url(../static/images/error.png) 5px 17px no-repeat;  
    color: red;  
}  
  
div.ok {  
    background: url(../static/images/ok.png) 5px 17px no-repeat;  
    color: green;  
}
```

提交表单时进行验证：

```

$('#form1').submit(function() {
    //在表单提交之前判断验证是否全部通过
    // console.log(myValidate.form())
    if (!myValidate.form()) {
        return false;
    }
})

```

远程验证

```

// 验证规则
$('#form1').validate({
    //验证规则
    rules: {
        username: {
            required: true,
            remote: {
                url: "http://127.0.0.1:8000/checkUsername",
                type: "get"
            }
        },
    },
    //错误提示信息
    messages: {
        username: {
            required: '用户名不能为空',
            remote: '用户已被占用'
        }
    }
})

```

```

// 服务端异步响应
// 验证通过响应字符串型的"true"
// 验证不通过响应字符串型的"false"
//假设数据库中的用户信息为
var userData = [
    { username: '张三', password: '123' },
    { username: '李四', password: '123' },
    { username: '王五', password: '123' },
];
app.get("/checkUsername", function(request, response) {
    //接收get请求的数据
    var data = request.query;
    // 查询数据库，判断用户名是否存在
    var result = "true";
    for (var i = 0; i < userData.length; i++) {
        if (userData[i].username == data.username) {
            result = "false";
            break;
        }
    }
    // 响应结果给浏览器
    response.send(result);
})

```

用 户 名	<input type="text" value="张三"/>
	✖ 用户已被占用
密 码	<input type="password" value="..."/>
	✔ 验证通过
确 认 密 码	<input type="password" value="..."/>
	✔ 验证通过
电 子 邮 箱	<input type="text"/>
	✖ 邮箱不能为空
手 机	<input type="text"/>
	✖ 手机号不能为空
<input type="button" value="开始注册"/>	