



Jquery插件validate

jQuery Validate 插件为表单提供了强大的验证功能，让客户端表单验证变得更简单

jQuery Validate 插件为表单提供了强大的验证功能，让客户端表单验证变得更简单，同时提供了大量的定制选项，满足应用程序各种需求。

该插件捆绑了一套有用的验证方法，包括 **URL** 和电子邮件验证，同时提供了一个用来编写用户自定义方法的 **API**。所有的捆绑方法默认使用英语作为错误信息，且已翻译成其他 **37** 种语言。



➤ 官方网址

<http://jqueryvalidation.org/>

➤ 文件夹说明

dist

|—— localization

| |—— 37 种语言包 验证提示中文

|—— additional-methods.js 扩展的自定义验证文件

|—— additional-methods.min.js 压缩版 扩展的自定义验证文件

|—— jquery.validate.js 验证主文件

|—— jquery.validate.min.js 压缩版 验证主文件

➤引入文件

<!-- 依赖 jQuery -->

<script src="jquery.js"></script>

<!-- 验证文件 -->

<script src="jquery.validate.js"></script>

<!-- [验证提示汉语包 **若自定义消息可不引入] -->

<script src="messages_zh.js"></script>

<script>

\$(function(){

 //选择指定的表单 form 执行validate() 方法

 \$('.myform').validate();

})

</script>

validate 自带规则说明

- 1.required:true 必输字段
- 2.remote:"check.php" 使用ajax方法调用check.php验证输入值
- 3.email:true 必须输入正确格式的电子邮件
- 4.url:true 必须输入正确格式的网址
- 5.date:true 必须输入正确格式的日期
- 6.dateISO:true 必须输入正确格式的日期(ISO),
例如: 2009-06-23, 1998/01/22 只验证格式, 不验证有效性
- 7.number:true 必须输入合法的数字(负数, 小数)
- 8.digits:true 必须输入整数
- 9.creditcard: 必须输入合法的信用卡号
- 10.equalTo:"#field" 输入值必须和#field相同
- 11.accept: 输入拥有合法后缀名的字符串 (上传文件的后缀)
- 12.maxlength:5 输入长度最多是5的字符串(汉字算一个字符)
- 13.minlength:10 输入长度最小是10的字符串(汉字算一个字符)
- 14.rangelength:[5,10] 输入长度必须介于5和10之间的字符串(汉字算一个字符)
- 15.range:[5,10] 输入值必须介于 5 和 10 之间
- 16.max:5 输入值不能大于5
- 17.min:10 输入值不能小于10

validate验证插件 标签添加属性方式

将校验规则写到控件中 (不推荐)

➤ 必选 文本框

用户: `<input type="text" name="username" required minlength=5 maxlength=10>`

➤ 必选 密码

密码: `<input type="password" id="pwd" name="pwd" required minlength=5 maxlength=10>`

重复密码: `<input type="password" name="pwd2" equalTo:"#pwd">`

➤ 必选 单选按钮

男: `<input type="radio" value="man" name="sex" required>`

女: `<input type="radio" value="woman" name="sex" required>`

➤ 必选 多选按钮

我同意: `<input type="checkbox" class="checkbox" name="agree" required>`

➤ 必选 多选按钮 最少选择2个

音乐: `<input type="checkbox" value="music" name="fav[]" required minlength="2">`

电影: `<input type="checkbox" value="film" name="fav[]">`

跑步: `<input type="checkbox" value="run" name="fav[]">`

validate验证插件 js添加方式

➤ `$("#myform").validate({})`

参数 `{}` 接受一个对象

属性有:

{

`rules:{}` || string, //给指定表单元素 添加规则

`messages:{}`, //给表单元素制定规则指定提示信息

`success:function(label){}`, //字段验证通过执行函数

`errorPlacement:function(error, element){}`,

//指定错误标签位置

`errorElement: String` //自定义错误标签名称

}

validate验证插件 js添加方式

将校验规则写到js代码中

```
$(function() {  
    $("#myform").validate({  
        rules: {  
            name: "required",  
            email: {  
                required: true,  
                email: true  
            },  
            upload: {  
                required: true,  
                accept: "flv|jpg"  
            }  
        }  
    })  
})
```


validate自定义消息

```
$(function() {  
    $("#myform").validate({  
        rules: {  
            name: "required",  
            upload: {  
                required: true,  
                accept: "png|jpg"  
            }  
        },  
        messages: {  
            name: "用户名必须填写!",  
            upload: {  
                required: "必须上传照片!",  
                accept: "上传格式为png或jpg!"  
            }  
        }  
    })  
})
```

➤ 查看下jQuery.Validate源代码，在265行有提示消息的定义方式，我们就可以进行修改这边的消息改成中文的方式，或者使用自定义了一个中文的消息

jQuery.Validate.message_cn.js，然后使jQuery.extend来覆盖jQuery.Validate自身的消息

```
$.extend($.validator.messages, {  
    required:"此字段必填!",  
    remote: "请修正此字段",  
    url: "请输入有效的网址",  
    .....  
});
```

validate自定义验证规则

`jQuery.validator.addMethod(name,method,message)`

接收3个参数: **name**-验证规则名

method-验证规则实现函数

message-验证不通过显示的错误 消息

method需要注意的是, 这个**method**实现函数接收2个元素:

value: 检测的对象的值

element: 检测的对象

➤自定义验证手机号码

```
$.validator.addMethod("telephone", function(value, element) {  
    var tel =  
    /^(130|131|132|133|134|135|136|137|138|139|150|153|157|158|159|  
    180|187|188|189)\d{8}$/;  
    return tel.test(value) || this.optional(element);  
}, "请输入正确的手机号码");
```

➤自定义验证用户名规则

```
$.validator.addMethod("users", function(value, element) {  
    var tel = /^[a-zA-Z][a-z0-9A-Z_-]{6,16}$/;  
    return tel.test(value) || this.optional(element);  
}, "以大小写字母开头，6-16位字符字母、数字和 _");
```

➤使用方式

```
<input type="text" users="true" name="username">
```

```
<style type="text/css">
label.error {
background: url(images/error.png) no-repeat 0px 0px;
padding-left: 40px;
font-size:12px;
}
input.error{
border: dashed 1px red;
}
</style>
```


➤ success:callback

JS 每个字段验证通过执行函数

```
$("#myform").validate({  
  success:function(label){  
    label.addClass('valid'); //给label元素添加class valid  
  }  
}
```

➤ CSS

```
label.valid{  
  background:url(images/checked.gif) no-repeat 0 0;  
  padding:0 10px;  
  color:#666;  
}
```

➤errorPlacement: Callback

默认: 把错误信息放在验证的元素后面

指明错误放置的位置, 默认情况是:

`error.appendTo(element.parent());`即把错误信息放在验证的元素后面

```
$("#myform").validate({  
    errorPlacement:function(error, element){  
        error.appendTo(element.parent());  
    }  
})
```

➤ `errorElement`: String Default: "label"

用什么标签标记错误，默认的是label你可以改成em

把自定义标签改成 em

```
$("#myform").validate({  
  errorElement: "em"  
})
```

➤remote: URL

使用**ajax**方式进行验证，默认会提交当前验证的值到远程地址，如果需要提交其他的值，可以使用**data**选项

remote: "check.php"

```
remote: {  
    url: "check.php",           //后台处理程序  
    type: "post",              //数据发送方式  
    dataType: "json",          //接受数据格式  
    data: {                     //要传递的数据  
        username: function() {  
            return $("#username").val();  
        }  
    }  
}
```

远程地址只能输出 "true" 或 "false"，不能有其它输出

➤ 重置表单

```
<input type="reset" value="reset" id="reset">
```

```
var validator = $("#myform").validate({  
    submitHandler:function(form){//提交表单  
        alert("submitted");  
        form.submit();  
    }  
});  
$("#reset").click(function() {//重置表单  
    validator.resetForm();  
});
```


➤提交表单

```
<input type="submit" value="提交">
var validator = $("#myform").validate({
    submitHandler:function(form){//提交表单
        alert("submitted");
        form.submit();
    }
});
$("#reset").click(function() {//重置表单
    validator.resetForm();
});
```



谢谢观看...

SHANXI **UNIQUE**
TECHNOLOGY CO.,LTD.