

# jQuery 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，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。

该插件是由 Jörn Zaefferer 编写和维护的，他是 jQuery 团队的一名成员，是 jQuery UI 团队的主要开发人员，是 QUnit 的维护人员。该插件在 2006 年 jQuery 早期的时候就已经开始出现，并一直更新至今。目前版本是 **1.14.0**。

访问 [jQuery Validate 官网](#)，下载最新版的 jQuery Validate 插件。

菜鸟教程提供的 1.14.0 版本下载地址：<http://static.runoob.com/download/jquery-validation-1.14.0.zip>

## 导入 js 库（使用菜鸟教程提供的CDN）

```
<script src="http://static.runoob.com/assets/jquery-validation-1.14.0/lib/jquery.js"></script>
<script src="http://static.runoob.com/assets/jquery-validation-1.14.0/dist/jquery.validate.min.js"></script>
```

## 默认校验规则

| 序号 | 规则                 | 描述  |
|----|--------------------|---|
| 1  | required:true      | 必须输入的字段。  |
| 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。日期校验 ie6 出错，慎用。                             |
| 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。                      |

## 默认提示

```
messages: {
  required: "This field is required.",
  remote: "Please fix this field.",
  email: "Please enter a valid email address.",
  url: "Please enter a valid URL.",
  date: "Please enter a valid date.",
  dateISO: "Please enter a valid date ( ISO ).",
  number: "Please enter a valid number.",
  digits: "Please enter only digits.",
  creditcard: "Please enter a valid credit card number.",
  equalTo: "Please enter the same value again.",
  maxlength: $.validator.format( "Please enter no more than {0} characters." ),
  minlength: $.validator.format( "Please enter at least {0} characters." ),
  rangelength: $.validator.format( "Please enter a value between {0} and {1} characters long." ),
  range: $.validator.format( "Please enter a value between {0} and {1}." ),
  max: $.validator.format( "Please enter a value less than or equal to {0}." ),
  min: $.validator.format( "Please enter a value greater than or equal to {0}." )
}
```

jQuery Validate提供了中文信息提示包，位于下载包的 dist/localization/messages\_zh.js，内容如下：

```
(function( factory ) {
  if ( typeof define === "function" && define.amd ) {
    define( ["jquery", "../jquery.validate"], factory );
  } else {
    factory( jQuery );
  }
})(function( $ ) {

  /*
   * Translated default messages for the jQuery validation plugin.
   * Locale: ZH (Chinese, 中文 (Zhōngwén), 汉语, 漢語)
  */
  $.validator.messages = {
    required: "必填",
    remote: "请改正此栏内容。",
    email: "请输入正确的电子邮件地址。",
    url: "请输入正确的URL。",
    date: "请输入正确的日期。",
    dateISO: "请输入正确的日期 ( ISO )。",
    number: "请输入正确的数字。",
    digits: "只能输入数字。",
    creditcard: "请输入正确的信用卡号码。",
    equalTo: "请输入跟前面一样的内容。",
    maxlength: $.validator.format( "最多只能输入{0}个字符。" ),
    minlength: $.validator.format( "最少要输入{0}个字符。" ),
    rangelength: $.validator.format( "请输入长度在{0}到{1}之间的字符串。" ),
    range: $.validator.format( "请输入介于{0}和{1}之间的值。" ),
    max: $.validator.format( "请输入不大于{0}的值。" ),
    min: $.validator.format( "请输入不小于{0}的值。" )
  }
});
```

```
*/  
$.extend($.validator.messages, {  
    required: "这是必填字段",  
    remote: "请修正此字段",  
    email: "请输入有效的电子邮件地址",  
    url: "请输入有效的网址",  
    date: "请输入有效的日期",  
    dateISO: "请输入有效的日期 (YYYY-MM-DD)",  
    number: "请输入有效的数字",  
    digits: "只能输入数字",  
    creditcard: "请输入有效的信用卡号码",  
    equalTo: "你的输入不相同",  
    extension: "请输入有效的后缀",  
    maxlength: $.validator.format("最多可以输入 {0} 个字符"),  
    minlength: $.validator.format("最少要输入 {0} 个字符"),  
    rangelength: $.validator.format("请输入长度在 {0} 到 {1} 之间的字符串"),  
    range: $.validator.format("请输入范围在 {0} 到 {1} 之间的数值"),  
    max: $.validator.format("请输入不大于 {0} 的数值"),  
    min: $.validator.format("请输入不小于 {0} 的数值")  
});  
  
});
```

你可以将该本地化信息文件 dist/localization/messages\_zh.js 引入到页面：

```
<script src="http://static.runoob.com/assets/jquery-validation-1.14.0/dist/localization/messages_zh.js">  
</script>
```

## 使用方式

### 1、将校验规则写到控件中

```
<script src="http://static.runoob.com/assets/jquery-validation-1.14.0/lib/jquery.js"></script>  
<script src="http://static.runoob.com/assets/jquery-validation-1.14.0/dist/jquery.validate.min.js"></script>  
<script src="http://static.runoob.com/assets/jquery-validation-1.14.0/dist/localization/messages_zh.js">  
</script>  
<script>  
$.validator.setDefaults({  
    submitHandler: function() {  
        alert("提交事件!");  
    }  
});  
$(document).ready(function() {  
    $("#commentForm").validate();  
});  
</script>
```

```
<form class="cmxform" id="commentForm" method="get" action="">
  <fieldset>
    <legend>输入您的名字, 邮箱, URL, 备注。</legend>
    <p>
      <label for="cname">Name (必需, 最小两个字母)</label>
      <input id="cname" name="name" minlength="2" type="text" required>
    </p>
    <p>
      <label for="cemail">E-Mail (必需)</label>
      <input id="cemail" type="email" name="email" required>
    </p>
    <p>
      <label for="curl">URL (可选)</label>
      <input id="curl" type="url" name="url">
    </p>
    <p>
      <label for="ccomment">备注 (必需)</label>
      <textarea id="ccomment" name="comment" required></textarea>
    </p>
    <p>
      <input class="submit" type="submit" value="Submit">
    </p>
  </fieldset>
</form>
```

[尝试一下 »](#)

## 2、将校验规则写到 js 代码中

```
$(function() {
  // 在键盘按下并释放及提交后验证提交表单
  $("#signupForm").validate({
    rules: {
      firstname: "required",
      lastname: "required",
      username: {
        required: true,
        minlength: 2
      },
      password: {
        required: true,
        minlength: 5
      },
      confirm_password: {
        required: true,
        minlength: 5,

```

```

        equalTo: "#password"
    },
    email: {
        required: true,
        email: true
    },
    topic: {
        required: "#newsletter:checked",
        minlength: 2
    },
    agree: "required"
},
messages: {
    firstname: "请输入您的名字",
    lastname: "请输入您的姓氏",
    username: {
        required: "请输入用户名",
        minlength: "用户名必需由两个字母组成"
    },
    password: {
        required: "请输入密码",
        minlength: "密码长度不能小于 5 个字母"
    },
    confirm_password: {
        required: "请输入密码",
        minlength: "密码长度不能小于 5 个字母",
        equalTo: "两次密码输入不一致"
    },
    email: "请输入一个正确的邮箱",
    agree: "请接受我们的声明",
    topic: "请选择两个主题"
}
})
});

```

messages 处，如果某个控件没有 message，将调用默认的信息

```

<form class="cmxform" id="signupForm" method="get" action="">
  <fieldset>
    <legend>验证完整的表单</legend>
    <p>
      <label for="firstname">名字</label>
      <input id="firstname" name="firstname" type="text">
    </p>
    <p>
      <label for="lastname">姓氏</label>
      <input id="lastname" name="lastname" type="text">
    </p>

```

```
<p>
  <label for="username">用户名</label>
  <input id="username" name="username" type="text">
</p>
<p>
  <label for="password">密码</label>
  <input id="password" name="password" type="password">
</p>
<p>
  <label for="confirm_password">验证密码</label>
  <input id="confirm_password" name="confirm_password" type="password">
</p>
<p>
  <label for="email">Email</label>
  <input id="email" name="email" type="email">
</p>
<p>
  <label for="agree">请同意我们的声明</label>
  <input type="checkbox" class="checkbox" id="agree" name="agree">
</p>
<p>
  <label for="newsletter">我乐意接收新信息</label>
  <input type="checkbox" class="checkbox" id="newsletter" name="newsletter">
</p>
<fieldset id="newsletter_topics">
  <legend>主题（至少选择两个） - 注意：如果没有勾选“我乐意接收新信息”以下选项会隐藏，但我们这里作为演示让它可见</legend>
  <label for="topic_marketflash">
    <input type="checkbox" id="topic_marketflash" value="marketflash" name="topic">Marketflash
  </label>
  <label for="topic_fuzz">
    <input type="checkbox" id="topic_fuzz" value="fuzz" name="topic">Latest fuzz
  </label>
  <label for="topic_digester">
    <input type="checkbox" id="topic_digester" value="digester" name="topic">Mailing list digester
  </label>
  <label for="topic" class="error">Please select at least two topics you'd like to receive.</label>
</fieldset>
<p>
  <input class="submit" type="submit" value="提交">
</p>
</fieldset>
</form>
```

[尝试一下 »](#)

**required: true** 值是必须的。

**required: "#aa:checked"** 表达式的值为真，则需要验证。

**required: function(){} 返回为真**，表示需要验证。

后边两种常用于，表单中需要同时填或不填的元素。

## 常用方法及注意问题

### 1、用其他方式替代默认的 SUBMIT

```
$(document).ready(function() {  
    $("#signupForm").validate({  
        submitHandler: function(form){  
            alert("提交事件!");  
            form.submit();  
        }  
    });  
});
```

使用 ajax 方式

```
$(".selector").validate({  
    submitHandler: function(form)  
    {  
        $(form).ajaxSubmit();  
    }  
});
```

可以设置 validate 的默认值，写法如下：

```
$.validator.defaults({  
    submitHandler: function(form) { alert("提交事件!");form.submit(); }  
});
```

如果想提交表单, 需要使用 form.submit(), 而不要使用 \$(form).submit()。

### 2、debug，只验证不提交表单

如果这个参数为true，那么表单不会提交，只进行检查，调试时十分方便。

```
$(document).ready(function() {  
    $("#signupForm").validate({  
        debug:true  
    });  
});
```

如果一个页面中有多个表单都想设置成为 debug，则使用：

```
$.validator.setDefaults({  
    debug: true  
})
```

### 3、ignore：忽略某些元素不验证

```
ignore: ".ignore"
```

### 4、更改错误信息显示的位置

```
errorPlacement: Callback
```

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。

```
errorPlacement: function(error, element) {  
    error.appendTo(element.parent());  
}
```

### 实例

<p>将错误信息放在 label 元素后并使用 span 元素包裹它</p>

```
<form method="get" class="cmxform" id="form1" action="">  
    <fieldset>  
        <legend>Login Form</legend>  
        <p>  
            <label for="user">Username</label>  
            <input id="user" name="user" required minlength="3">  
        </p>  
        <p>  
            <label for="password">Password</label>  
            <input id="password" type="password" maxlength="12" name="password" required minlength="5">  
        </p>  
        <p>  
            <input class="submit" type="submit" value="Login">  
        </p>  
    </fieldset>  
</form>
```

尝试一下 »

代码的作用是：一般情况下把错误信息显示在 <td class="status"></td> 中，如果是 radio 则显示在 <td></td> 中，如果是 checkbox 则显示在内容的后面。

| 参数 | 类型 | 描述 | 默认值 |
|----|----|----|-----|
|----|----|----|-----|



|                     |          |   |         |
|---------------------|----------|---|---------|
| errorClass          | String   | 指定错误提示的 css 类名，可以自定义错误提示的样式。  | "error" |
| errorElement        | String   | 用什么标签标记错误，默认是 label，可以改成 em。  | "label" |
| errorContainer      | Selector | 显示或者隐藏验证信息，可以自动实现有错误信息出现时把容器属性变为显示，无错误时隐藏，用处不大。<br>errorContainer: "#messageBox1, #messageBox2" |         |
| errorLabelContainer | Selector | 把错误信息统一放在一个容器里面。  |         |
| wrapper             | String   | 用什么标签再把上边的 errorElement 包起来。  |         |

一般这三个属性同时使用，实现在一个容器内显示所有错误提示的功能，并且没有信息时自动隐藏。

```
errorContainer: "div.error",
errorLabelContainer: $("#signupForm div.error"),
wrapper: "li"
```

## 5、更改错误信息显示的风格

设置错误提示的风格，可以增加图标显示，在该系统中已经建立了一个 validation.css，专门用于维护校验文件的风格。

```
input.error { border: 1px solid red; }
label.error {
  background:url("../demo/images/unchecked.gif") no-repeat 0px 0px;

  padding-left: 16px;

  padding-bottom: 2px;

  font-weight: bold;

  color: #EA5200;
}
label.checked {
  background:url("../demo/images/checked.gif") no-repeat 0px 0px;
}
```

## 6、每个字段验证通过执行函数

```
success: String,Callback
```

要验证的元素通过验证后的动作，如果跟一个字符串，会当作一个 css 类，也可跟一个函数。

```
success: function(label) {
    // set &nbsp; as text for IE
    label.html("&nbsp;").addClass("checked");
    //label.addClass("valid").text("Ok!")
}
```

添加 "valid" 到验证元素，在 CSS 中定义的样式 <style>label.valid {}</style>。

```
success: "valid"
```

7、验证的触发方式修改

下面的虽然是 boolean 型的，但建议除非要改为 false，否则别乱添加。

| 触发方式         | 类型      | 描述   | 默认值   |
|--------------|---------|--|-------|
| onsubmit     | Boolean | 提交时验证。设置为 false 就用其他方法去验证。                             | true  |
| onfocusout   | Boolean | 失去焦点时验证（不包括复选框/单选按钮）。                                  | true  |
| onkeyup      | Boolean | 在 keyup 时验证。   | true  |
| onclick      | Boolean | 在点击复选框和单选按钮时验证。  | true  |
| focusInvalid | Boolean | 提交表单后，未通过验证的表单（第一个或提交之前获得焦点的未通过验证的表单）会获得焦点。            | true  |
| focusCleanup | Boolean | 如果是 true 那么当未通过验证的元素获得焦点时，移除错误提示。避免和 focusInvalid 一起用。 | false |

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

8、异步验证

```
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-email.php"
```

```
remote: {
  url: "check-email.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"，不能有其他输出。

## 9、添加自定义校验

```
addMethod: name, method, message
```

自定义验证方法

```
// 中文字两个字节
jQuery.validator.addMethod("byteRangeLength", function(value, element, param) {
  var length = value.length;
  for(var i = 0; i < value.length; i++){
    if(value.charCodeAt(i) > 127){
      length++;
    }
  }
  return this.optional(element) || ( length >= param[0] && length <= param[1] );
}, $.validator.format("请确保输入的值在{0}~{1}个字节之间(一个中文字算2个字节)"));

// 邮政编码验证
jQuery.validator.addMethod("isZipCode", function(value, element) {
  var tel = /^[0-9]{6}$/;
  return this.optional(element) || (tel.test(value));
}, "请正确填写您的邮政编码");
```

**注意：**要在 additional-methods.js 文件中添加或者在 jquery.validate.js 文件中添加。建议一般写在 additional-methods.js 文件中。

**注意：**在 messages\_cn.js 文件中添加：isZipCode: "只能包括中文字、英文字母、数字和下划线"。调用前要添加对 additional-methods.js 文件的引用。

## 10、radio 和 checkbox、select 的验证

radio 的 required 表示必须选中一个。

```
<input type="radio" id="gender_male" value="m" name="gender" required />
<input type="radio" id="gender_female" value="f" name="gender"/>
```

checkbox 的 required 表示必须选中。

```
<input type="checkbox" class="checkbox" id="agree" name="agree" required />
```

checkbox 的 minlength 表示必须选中的最小个数，maxlength 表示最大的选中个数，rangelength:[2,3] 表示选中个数区间。

```
<input type="checkbox" class="checkbox" id="spam_email" value="email" name="spam[]" required minlength="2" />
<input type="checkbox" class="checkbox" id="spam_phone" value="phone" name="spam[]" />
<input type="checkbox" class="checkbox" id="spam_mail" value="mail" name="spam[]" />
```

select 的 required 表示选中的 value 不能为空。

```
<select id="jungle" name="jungle" title="Please select something!" required>
  <option value=""></option>
  <option value="1">Buga</option>
  <option value="2">Baga</option>
  <option value="3">0i</option>
</select>
```

select 的 minlength 表示选中的最小个数（可多选的 select），maxlength 表示最大的选中个数，rangelength:[2,3] 表示选中个数区间。

```
<select id="fruit" name="fruit" title="Please select at least two fruits" class="{required:true, minlength:2}" multiple="multiple">
  <option value="b">Banana</option>
  <option value="a">Apple</option>
  <option value="p">Peach</option>
  <option value="t">Turtle</option>
</select>
```

## jQuery.validate 中文 API

| 名称 | 返回类型 | 描述 |
|----|------|----|
|----|------|----|

|   |                 |                |
|---|-----------------|----------------|
| validate(options)                             | Validator       | 验证所选的 FORM。    |
| valid()                                       | Boolean         | 检查是否验证通过。      |
| rules()                                       | Options         | 返回元素的验证规则。     |
| rules("add",rules)                            | Options         | 增加验证规则。        |
| rules("remove",rules)                         | Options         | 删除验证规则。        |
| removeAttrs(attributes)                       | Options         | 删除特殊属性并且返回它们。  |
| 自定义选择器  |                 |                |
| :blank  | Validator       | 没有值的筛选器。       |
| :filled                                       | Array <Element> | 有值的筛选器。        |
| :unchecked                                    | Array <Element> | 没选择的元素的筛选器。    |
| 实用工具  |                 |                |
| jQuery.format(template,argument,argumentN...) | String          | 用参数代替模板中的 {n}。 |

## Validator

validate 方法返回一个 Validator 对象。Validator 对象有很多方法可以用来引发校验程序或者改变 form 的内容，下面列出几个常用的方法。

| 名称                             | 返回类型      | 描述  |
|--------------------------------|-----------|---|
| form()                         | Boolean   | 验证 form 返回成功还是失败。                                   |
| element(element)               | Boolean   | 验证单个元素是成功还是失败。                                      |
| resetForm()                    | undefined | 把前面验证的 FORM 恢复到验证前原来的状态。                            |
| showErrors(errors)             | undefined | 显示特定的错误信息。  |
| Validator 函数                   |           |   |
| setDefaults(defaults)          | undefined | 改变默认的设置。  |
| addMethod(name,method,message) | undefined | 添加一个新的验证方法。必须包括一个独一无二的名字，一个 JAVASCRIPT 的方法和一个默认的信息。 |
| addClassRules(name,rules)      | undefined | 增加组合验证类型，在一个类里面用多种验证方法时比较有用。                        |

|                      |           |   |
|----------------------|-----------|---|
| addClassRules(rules) | undefined | 增加组合验证类型，在一个类里面用多种验证方法时比较有用。这个是同时加多个验证方法。 |
|----------------------|-----------|---|

内置验证方式

| 名称                              | 返回类型    | 描述                                       |
|---------------------------------|---------|--|
| required()                      | Boolean | 必填验证元素。                                  |
| required(dependency-expression) | Boolean | 必填元素依赖于表达式的结果。                           |
| required(dependency-callback)   | Boolean | 必填元素依赖于回调函数的结果。                          |
| remote(url)                     | Boolean | 请求远程校验。url 通常是一个远程调用方法。                  |
| minlength(length)               | Boolean | 设置最小长度。                                  |
| maxlength(length)               | Boolean | 设置最大长度。                                  |
| rangelength(range)              | Boolean | 设置一个长度范围 [min,max]。                      |
| min(value)                      | Boolean | 设置最小值。                                   |
| max(value)                      | Boolean | 设置最大值。                                   |
| email()                         | Boolean | 验证电子邮箱格式。                                |
| range(range)                    | Boolean | 设置值的范围。                                  |
| url()                           | Boolean | 验证 URL 格式。                               |
| date()                          | Boolean | 验证日期格式（类似 30/30/2008 的格式，不验证日期准确性只验证格式）。 |
| dateISO()                       | Boolean | 验证 ISO 类型的日期格式。                          |
| dateDE()                        | Boolean | 验证德式的日期格式（29.04.1994 或 1.1.2006）。        |
| number()                        | Boolean | 验证十进制数字（包括小数的）。                          |
| digits()                        | Boolean | 验证整数。                                    |
| creditcard()                    | Boolean | 验证信用卡号。                                  |
| accept(extension)               | Boolean | 验证相同后缀名的字符串。                             |

|                |         |                 |
|----------------|---------|-----------------|
| equalTo(other) | Boolean | 验证两个输入框的内容是否相同。 |
| phoneUS()      | Boolean | 验证美式的电话号码。      |

validate ()的可选项

| 描述   | 代码  |
|--|---|
| debug：进行调试模式（ 表单不提交 ）。                                 | <pre>\$(".selector").validate({   debug:true })</pre>   |
| 把调试设置为默认。  | <pre>\$.validator.setDefaults({   debug:true })</pre>   |
| submitHandler：通过验证后运行的函数，里面要加上表单提交的函数，否则表单不会提交。        | <pre>\$(".selector").validate({   submitHandler:function(form) {     \$(form).ajaxSubmit();   } })</pre>                            |
| ignore：对某些元素不进行验证。                                     | <pre>\$("#myform").validate({   ignore:".ignore" })</pre>   |
| rules：自定义规则，key:value 的形式，key 是要验证的元素，value 可以是字符串或对象。 | <pre>\$(".selector").validate({   rules:{     name:"required",     email:{       required:true,       email:true     }   } })</pre> |

messages：自定义的提示信息，key:value 的形式，key 是要验证的元素，value 可以是字符串或函数。

```
$(".selector").validate({
  rules:{
    name:"required",
    email:{
      required:true,
      email:true
    }
  },
  messages:{
    name:"Name不能为空",
    email:{
      required:"E-mail不能为空",
      email:"E-mail地址不正确"
    }
  }
})
```

groups：对一组元素的验证，用一个错误提示，用 errorPlacement 控制把出错信息放在哪里。

```
$("#myform").validate({
  groups:{
    username:"fname
    lname"
  },
  errorPlacement:function(error,element) {
    if (element.attr("name") == "fname" || element.attr("name") =
    = "lname")
      error.insertAfter("#lastname");
    else
      error.insertAfter(element);
  },
  debug:true
})
```

OnSubmit：类型 Boolean，默认 true，指定是否提交时验证。

```
$(".selector").validate({
  onsubmit:false
})
```

onfocusout：类型 Boolean，默认 true，指定是否在获取焦点时验证。

```
$(".selector").validate({
  onfocusout:false
})
```



onkeyup : 类型 Boolean , 默认 true , 指定是否在敲击键盘时验证。

```
$(".selector").validate({  
    onkeyup:false  
})
```

onclick : 类型 Boolean , 默认 true , 指定是否在鼠标点击时验证 ( 一般验证 checkbox、 radiobox ) 。

```
$(".selector").validate({  
    onclick:false  
})
```

focusInvalid : 类型 Boolean , 默认 true。提交表单后, 未通过验证的表单 ( 第一个或提交之前获得焦点的未通过验证的表单 ) 会获得焦点。

```
$(".selector").validate({  
    focusInvalid:false  
})
```

focusCleanup : 类型 Boolean , 默认 false。当未通过验证的元素获得焦点时, 移除错误提示 ( 避免和 focusInvalid 一起使用 ) 。

```
$(".selector").validate({  
    focusCleanup:true  
})
```

errorClass : 类型 String , 默认 "error"。指定错误提示的 css 类名, 可以自定义错误提示的样式。

```
$(".selector").validate({  
    errorClass:"invalid"  
})
```

errorElement : 类型 String , 默认 "label"。指定使用什么标签标记错误。

```
$(".selector").validate  
    errorElement:"em"  
})
```

wrapper : 类型 String , 指定使用什么标签再把上边的 errorElement 包起来。

```
$(".selector").validate({  
    wrapper:"li"  
})
```

errorLabelContainer : 类型 Selector , 把错误信息统一放在一个容器里面。

```
$("#myform").validate({  
    errorLabelContainer:"#messageBox",  
    wrapper:"li",  
    submitHandler:function() {
```

|  |   |
|--|---|
|  | <pre>        alert("Submitted!")     } })</pre>   |
| showErrors：跟一个函数，可以显示总共有多少个未通过验证的元素。                 | <pre>\$(".selector").validate({     showErrors:function(errorMap,errorList) {         \$("#summary").html("Your form contains " + this.numberOfInvalids() + " errors,see details below.");         this.defaultShowErrors();     } })</pre> |
| errorPlacement：跟一个函数，可以自定义错误放到哪里。                    | <pre>\$("#myform").validate({     errorPlacement:function(error,element) {         error.appendTo(element.parent("td").next("td"));     },     debug:true })</pre>  |
| success：要验证的元素通过验证后的动作，如果跟一个字符串，会当作一个 css 类，也可跟一个函数。 | <pre>\$("#myform").validate({     success:"valid",     submitHandler:function() {         alert("Submitted!")     } })</pre>  |
| highlight：可以给未通过验证的元素加效果、闪烁等。                        |   |

## addMethod(name,method,message)方法

参数 name 是添加的方法的名字。

参数 method 是一个函数，接收三个参数 (value,element,param)。

value 是元素的值，element 是元素本身，param 是参数。

我们可以用 addMethod 来添加除内置的 Validation 方法之外的验证方法。比如有一个字段，只能输一个字母，范围是 a-f，写法如下：

```
$.validator.addMethod("af",function(value,element,params){
    if(value.length>1){
```

```
        return false;
    }
    if(value>=params[0] && value<=params[1]){
        return true;
    }else{
        return false;
    }
}, "必须是一个字母,且a-f");
```

如果有个表单字段的 name="username", 则在 rules 中写：

```
username:{
    af:["a","f"]
}
```

addMethod 的第一个参数，是添加的验证方法的名字，这时是 af。

addMethod 的第三个参数，是自定义的错误提示，这里的提示为:"必须是一个字母,且a-f"。

addMethod 的第二个参数，是一个函数，这个比较重要，决定了用这个验证方法时的写法。

如果只有一个参数，直接写，比如 af:"a"，那么 a 就是这个唯一的参数，如果多个参数，则写在 [] 里，用逗号分开。

## meta String 方式

```
$("#myform").validate({

    meta:"validate",

    submitHandler:function() {
    alert("Submitted!") }

})
```

```
<script type="text/javascript"
src="js/jquery.metadata.js"></script>

<script type="text/javascript"
src="js/jquery.validate.js"></script>

<form id="myform">

    <input type="text"
name="email" class="{validate:{ required:true,email:true }}" />

    <input type="submit"
value="Submit" />
```

```
</form>
```

# 实例演示

## 虚构的实例

- 错误消息容器
- 自定义消息作为元素数据
- radio ( 单选按钮 )、checkbox ( 复选按钮 ) 和 select ( 下拉框 )
- 与表单 ( Form ) 插件的交互 ( AJAX 提交 )
- 自定义方法和消息显示
- 动态表单
- 使用 jQuery UI Themeroller 定义表单样式
- TinyMCE - 一个轻量级的基于浏览器的所见即所得编辑器
- 文件输入框
- jQuery Mobile 表单验证

## 现实世界的实例

- Milk 注册表单
- Marketo 注册表单
- 房屋买卖折叠面板表单
- 远程 CAPTCHA ( 验证码 ) 验证

## 实例下载

点击下载

官方实例

← jQuery 属性

jQuery Accordion →

 点我分享笔记