# validation插件

validation是基于JQuery的插件, 实现了表单的验证功能.

validation.js官网地址：

[http://bassistance.de/jquery-plugins/jquery-plugin-validation](http://bassistance.de/jquery-plugins/jquery-plugin-validation" \t "https://blog.csdn.net/sanbingyutuoniao123/article/details/_blank)

## JQuery.validation下载地址

<https://github.com/jquery-validation/jquery-validation/releases/tag/1.19.1>

此案例是下载的1.19.1版本, 想下载其他版本, 通过官网可以找到其他的版本.

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

导入中文支持js文件

<script src="js/messages\_zh.js"></script>

默认校验规则

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 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

默认的提示

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).",  
    dateDE: "Bitte geben Sie ein g眉ltiges Datum ein.",  
    number: "Please enter a valid number.",  
    numberDE: "Bitte geben Sie eine Nummer ein.",  
    digits: "Please enter only digits",  
    creditcard: "Please enter a valid credit card number.",  
    equalTo: "Please enter the same value again.",  
    accept: "Please enter a value with a valid extension.",  
    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}.")  
},

如需要修改，可在js代码中加入：

jQuery.extend(jQuery.validator.messages, {  
  required: "必选字段",  
  remote: "请修正该字段",  
  email: "请输入正确格式的电子邮件",  
  url: "请输入合法的网址",  
  date: "请输入合法的日期",  
  dateISO: "请输入合法的日期 (ISO).",  
  number: "请输入合法的数字",  
  digits: "只能输入整数",  
  creditcard: "请输入合法的信用卡号",  
  equalTo: "请再次输入相同的值",  
  accept: "请输入拥有合法后缀名的字符串",  
  maxlength: jQuery.validator.format("请输入一个 长度最多是 {0} 的字符串"),  
  minlength: jQuery.validator.format("请输入一个 长度最少是 {0} 的字符串"),  
  rangelength: jQuery.validator.format("请输入 一个长度介于 {0} 和 {1} 之间的字符串"),  
  range: jQuery.validator.format("请输入一个介于 {0} 和 {1} 之间的值"),  
  max: jQuery.validator.format("请输入一个最大为{0} 的值"),  
  min: jQuery.validator.format("请输入一个最小为{0} 的值")  
});

推荐做法，将此文件放入messages\_cn.js中，在页面中引入  
<script src="../js/messages\_cn.js" type="text/javascript"></script>

四、使用方式  
1.将校验规则写到控件中

<script src="../js/jquery.js" type="text/javascript"></script>  
<script src="../js/jquery.validate.js" type="text/javascript"></script>  
<script src="./js/jquery.metadata.js" type="text/javascript"></script>  
  
$().ready(function() {  
 $("#signupForm").validate();  
});  
  
  
<form id="signupForm" method="get" action="">  
    <p>  
        <label for="firstname">Firstname</label>  
        <input id="firstname" name="firstname" class="required" />  
    </p>  
 <p>  
  <label for="email">E-Mail</label>  
  <input id="email" name="email" class="required email" />  
 </p>  
 <p>  
  <label for="password">Password</label>  
  <input id="password" name="password" type="password" class="{required:true,minlength:5}" />  
 </p>  
 <p>  
  <label for="confirm\_password">确认密码</label>  
  <input id="confirm\_password" name="confirm\_password" type="password" class="{required:true,minlength:5,equalTo:'#password'}" />  
 </p>  
    <p>  
        <input class="submit" type="submit" value="Submit"/>  
    </p>  
</form>

使用class="{}"的方式，必须引入包：jquery.metadata.js

可以使用如下的方法，修改提示内容：  
class="{required:true,minlength:5,messages:{required:'请输入内容'}}"

在使用equalTo关键字时，后面的内容必须加上引号，如下代码：  
class="{required:true,minlength:5,equalTo:'#password'}"

2.将校验规则写到js代码中

IMG_260

IMG_261

$().ready(function() {  
 $("#signupForm").validate({  
        rules: {  
   firstname: "required",  
   email: {  
    required: true,  
    email: true  
   },  
   password: {  
    required: true,  
    minlength: 5  
   },  
   confirm\_password: {  
    required: true,  
    minlength: 5,  
    equalTo: "#password"  
   }  
  },  
        messages: {  
   firstname: "请输入姓名",  
   email: {  
    required: "请输入Email地址",  
    email: "请输入正确的email地址"  
   },  
   password: {  
    required: "请输入密码",  
    minlength: jQuery.format("密码不能小于{0}个字 符")  
   },  
   confirm\_password: {  
    required: "请输入确认密码",  
    minlength: "确认密码不能小于5个字符",  
    equalTo: "两次输入密码不一致不一致"  
   }  
  }  
    });  
});

IMG_262

IMG_263

//messages处，如果某个控件没有message，将调用默认的信息

IMG_264

IMG_265

<form id="signupForm" method="get" action="">  
    <p>  
        <label for="firstname">Firstname</label>  
        <input id="firstname" name="firstname" />  
    </p>  
 <p>  
  <label for="email">E-Mail</label>  
  <input id="email" name="email" />  
 </p>  
 <p>  
  <label for="password">Password</label>  
  <input id="password" name="password" type="password" />  
 </p>  
 <p>  
  <label for="confirm\_password">确认密码</label>  
  <input id="confirm\_password" name="confirm\_password" type="password" />  
 </p>  
    <p>  
        <input class="submit" type="submit" value="Submit"/>  
    </p>  
</form>

IMG_266

IMG_267

required:true 必须有值  
required:"#aa:checked"表达式的值为真，则需要验证  
required:function(){}返回为真，表时需要验证  
后边两种常用于，表单中需要同时填或不填的元素

五、常用方法及注意问题  
1.用其他方式替代默认的SUBMIT  
$().ready(function() {  
 $("#signupForm").validate({  
        submitHandler:function(form){  
            alert("submitted");     
            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.setDefaults({  
 submitHandler: function(form) { alert("submitted!");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

$().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，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

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

//示例：

IMG_268

IMG_269

<tr>  
    <td class="label"><label id="lfirstname" for="firstname">First Name</label></td>  
    <td class="field"><input id="firstname" name="firstname" type="text" value="" maxlength="100" /></td>  
    <td class="status"></td>  
</tr>  
<tr>  
    <td style="padding-right: 5px;">  
        <input id="dateformat\_eu" name="dateformat" type="radio" value="0" />  
        <label id="ldateformat\_eu" for="dateformat\_eu">14/02/07</label>  
    </td>  
    <td style="padding-left: 5px;">  
        <input id="dateformat\_am" name="dateformat" type="radio" value="1"  />  
        <label id="ldateformat\_am" for="dateformat\_am">02/14/07</label>  
    </td>  
    <td></td>  
</tr>  
<tr>  
    <td class="label">&nbsp;</td>  
    <td class="field" colspan="2">  
        <div id="termswrap">  
            <input id="terms" type="checkbox" name="terms" />  
            <label id="lterms" for="terms">I have read and accept the Terms of Use.</label>  
        </div>  
    </td>  
</tr>  
  
errorPlacement: function(error, element) {  
    if ( element.is(":radio") )  
        error.appendTo( element.parent().next().next() );  
    else if ( element.is(":checkbox") )  
        error.appendTo ( element.next() );  
    else  
        error.appendTo( element.parent().next() );  
}

IMG_270

IMG_271

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显示在<td></td>中，如果是 checkbox显示在内容的后面

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

errorElement：String  Default: "label"   
用什么标签标记错误，默认的是label你可以改成em

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  
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样式  
设置错误提示的样式，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样式

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，会当做一个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。

onsubmit：Boolean  Default: true   
提交时验证. 设置唯false就用其他方法去验证  
onfocusout：Boolean  Default: true   
失去焦点是验证(不包括checkboxes/radio buttons)  
onkeyup：Boolean  Default: true   
在keyup时验证.  
onclick：Boolean  Default: true   
在checkboxes 和 radio 点击时验证  
focusInvalid：Boolean  Default: 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  
focusCleanup：Boolean  Default: false   
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

// 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"，不能有其它输出

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");

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

2.在messages\_cn.js文件添加：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。

10radio和checkbox、select的验证

1.radio的required表示必须选中一个  
<input  type="radio" id="gender\_male" value="m" name="gender" class="{required:true}" />  
<input  type="radio" id="gender\_female" value="f" name="gender"/>

2.checkbox的required表示必须选中  
<input type="checkbox" class="checkbox" id="agree" name="agree" class="{required:true}" />

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[]" class="{required:true, 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[]" />

    3.select的required表示选中的value不能为空  
<select id="jungle" name="jungle" title="Please select something!" class="{required:true}">  
    <option value=""></option>  
    <option value="1">Buga</option>  
    <option value="2">Baga</option>  
    <option value="3">Oi</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)* | 返回*:Validator* | 验证所选的*FORM* |
| *valid()* | 返回*:Boolean* | 检查是否验证通过 |
| *rules()* | 返回*:Options* | 返回元素的验证规则 |
| *rules("add",rules)* | 返回*:Options* | 增加验证规则 |
| *rules("remove",rules)* | 返回*:Options* | 删除验证规则 |
| *removeAttrs(attributes)* | 返回*:Options* | 删除特殊属性并且返回他们 |
| *Custom selectors* | | |
| *:blank* | 返回*:Validator* | 没有值的筛选器 |
| *:filled* | 返回*:Array <Element  >* | 有值的筛选器 |
| *:unchecked* | 返回*:Array <Element  >* | 没选择的元素的筛选器 |
| *Utilities* | | |
| *jQuery.format*  *(template,argument  ,argumentN...)* | 返回*:String* | 用参数代替模板中的*{n}* |

|  |  |  |
| --- | --- | --- |
| *Validator:* | | |
| *validate*方法返回一个*Validator*对象*,*它有很多方法*,*让你能使用引发校验程序或者改变*form*的内容*. validator*对象有很多方法*,*但下面只是列出常用的 | | |
| *form()* | 返回*:Boolean* | 验证*form*返回成功还是失败 |
| *element(element)* | 返回*:Boolean* | 验证单个元素是成功还是失败 |
| *resetForm()* | 返回*:undefined* | 把前面验证的*FORM*恢复到验证前原来的状态 |
| *showErrors(errors)* | 返回*:undefined* | 显示特定的错误信息 |
|  | | |
| *Validator functions:* | | |
| *setDefaults(defaults)* | 返回*:undefined* | 改变默认的设置 |
| *addMethod(name,method,message)* | 返回*: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)* | 返回*:undefined* | 增加组合验证类型 在一个类里面用多种验证方法里比较有用 |
| *addClassRules(rules)* | 返回*:undefined* | 增加组合验证类型 在一个类里面用多种验证方法里比较有用*,*这个是一下子加多个 |

|  |  |  |
| --- | --- | --- |
| 内置验证方式： | | |
| *required()* | 返回*:Boolean* | 必填验证元素 |
| *required(dependency-expression)* | 返回*:Boolean* | 必填元素依赖于表达式的结果 |
| *required(dependency-callback)* | 返回*:Boolean* | 必填元素依赖于回调函数的结果 |
| *remote(url)* | 返回*:Boolean* | 请求远程校验。*url*通常是一个远程调用方法 |
| *minlength(length)* | 返回*:Boolean* | 设置最小长度 |
| *maxlength(length)* | 返回*:Boolean* | 设置最大长度 |
| *rangelength(range)* | 返回*:Boolean* | 设置一个长度范围*[min,max]* |
| *min(value)* | 返回*:Boolean* | 设置最大值 |
| *max(value)* | 返回*:Boolean* | 设置最小值 |
| *email()* | 返回*:Boolean* | 验证电子邮箱格式 |
| *range(range)* | 返回*:Boolean* | 设置值的范围 |
| *url()* | 返回*:Boolean* | 验证*URL*格式 |
| *date()* | 返回*: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()* | 返回*:Boolean* | 验证*ISO*类型的日期格式 |
| *dateDE()* | 返回*:Boolean* | 验证德式的日期格式（*29.04.1994 or  1.1.2006*） |
| *number()* | 返回*:Boolean* | 验证十进制数字（包括小数的） |
| *digits()* | 返回*:Boolean* | 验证整数 |
| *creditcard()* | 返回*:Boolean* | 验证信用卡号 |
| *accept(extension)* | 返回*:Boolean* | 验证相同后缀名的字符串 |
| *equalTo(other)* | 返回*:Boolean* | 验证两个输入框的内容是否相同 |
| *phoneUS()* | 返回*:Boolean* | 验证美式的电话号码 |

|  |  |
| --- | --- |
| *validate ()*的可选项： | |
| *debug:*进行调试模式（表单不提交）*:* | *$(".selector").validate*  *({*  *debug:true*  *})* |
| 把调试设置为默认*:* | *$.validator.setDefaults({*  *debug:true*  *})* |
| *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 | *$(".selector").validate({*  *submitHandler:function(form)  {*  *$(form).ajaxSubmit();*  *}*  *})* |
| *ignore:*  对某些元素不进行验证 | *$("#myform").validate({*  *ignore:".ignore"*  *})* |
| *rules:*  自定义规则*,key:value*的形式*,key*是要验证的元素*,value*可以是字符串或对象 | *$(".selector").validate({*  *rules:{*  *name:"required",*  *email:{*  *required:true,*  *email:true*  *}*  *}*  *})* |
| *messages:*  自定义的提示信息*key:value*的形式*key*是要验证的元素*,*值是字符串或函数 | *$(".selector").validate({*  *rules:{*  *name:"required",*  *email:{*  *required:true,*  *email:true*  *}*  *},*  *messages:{*  *name:"Name*不能为空*",*  *email:{*  *required:"E-mail*不能为空*",*  *email:"E-mail*地址不正确*"*  *}*  *}*  *})* |
| *groups:*  对一组元素的验证*,*用一个错误提示*,*用*error Placement*控制把出错信息放在哪里 | *$("#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*  *})* |
| *Onubmit Boolean*默认*:true*  是否提交时验证 | *$(".selector").validate({*  *onsubmit:false*  *})* |
| *onfocusout Boolean*默认*:true*  是否在获取焦点时验证 | *$(".selector").validate({*  *onfocusout:false*  *})* |
| *onkeyup Boolean*默认*:true*  是否在敲击键盘时验证 | *$(".selector").validate({*  *onkeyup:false*  *})* |
| *onclick Boolean*默认*:true*  是否在鼠标点击时验证（一般验证*checkbox,radiobox*） | *$(".selector").validate({*  *onclick:false*  *})* |
| *focusInvalid 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 | *$(".selector").validate({*  *focusInvalid:false*  *})* |
| *focusCleanup Boolean*默认*:false*  当未通过验证的元素获得焦点时*,*并移除错误提示（避免和*focusInvalid.*一起使用） | *$(".selector").validate({*  *focusCleanup:true*  *})* |
| *errorClass 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 | *$(".selector").validate({*  *errorClass:"invalid"*  *})* |
| *errorElement String*默认*:"label"*  使用什么标签标记错误 | *$(".selector").validate*  *errorElement:"em"*  *})* |
| *wrapper String*  使用什么标签再把上边的*errorELement*包起来 | *$(".selector").validate({*  *wrapper:"li"*  *})* |
| *errorLabelContainer Selector*  把错误信息统一放在一个容器里面 | *$("#myform").validate({*  *errorLabelContainer:"#messageBox",*  *wrapper:"li",*  *submitHandler:function() {  alert("Submitted!") }*  *})* |
| *showErrors:*  跟一个函数*,*可以显示总共有多少个未通过验证的元素 | *$(".selector").validate({*  *showErrors:function(errorMap,errorList) {*  *$("#summary").html("Your  form contains " + this.numberOfInvalids() + " errors,see details  below.");*  *this.defaultShowErrors();*  *}*  *})* |
| *errorPlacement:*  跟一个函数*,*可以自定义错误放到哪里 | *$("#myform").validate({*  *rrorPlacement:function(error,element) {   error.appendTo(element.parent("td").next("td"));*  *},*  *debug:true*    *})* |
| *success:*  要验证的元素通过验证后的动作*,*如果跟一个字符串*,*会当做一个*css*类*,*也可跟一个函数 | *$("#myform").validate({*  *success:"valid",*  *submitHandler:function()  { alert("Submitted!") }*  *})* |
| *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*built-in Validation  methods*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");*  用的时候*,*比如有个表单字段的*id="username",*则在*rules*中写  *username:{*  *af:["a","f"]*  *}*    *addMethod*的第一个参数*,*就是添加的验证方法的名子*,*这时是*af*  *addMethod*的第三个参数*,*就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*,*如果*af:"a",*那么*a*就是这个唯一的参数*,*如果多个参数*,*用在*[]*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>* |

转载自：[http://blog.sina.com.cn/s/blog\_608475eb0100h3h1.html](http://blog.sina.com.cn/s/blog_608475eb0100h3h1.html" \t "https://blog.csdn.net/sanbingyutuoniao123/article/details/_blank)

 转载自：[http://www.cnblogs.com/hejunrex/archive/2011/11/17/2252193.html](http://www.cnblogs.com/hejunrex/archive/2011/11/17/2252193.html" \t "https://blog.csdn.net/sanbingyutuoniao123/article/details/_blank)

关于 jquery.validate.js 表单验证插件如何在失去焦点时做验证。看手册后发现默认是在表单提交时执行验证代码。 手册中提到使用 onfocusout:false 来解决失去焦点时验证的参数功能，但是经测试无效果。 可以使用函数手工验证：

$("#signupForm").validate({

onfocusout: function(element){

$(element).valid();

}

});

# 案例

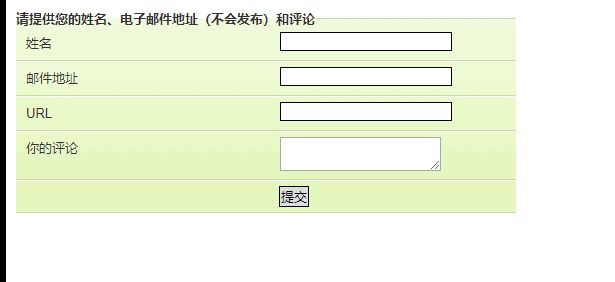
表单代码 关键:required 表示必填字段

|  |
| --- |
| <form class="cmxform" id="commentForm" method="get" action="">          <fieldset>              <legend>请提供您的姓名、电子邮件地址（不会发布）和评论</legend>              <p>                  <label for="cname">姓名</label>                  <input id="cname" name="name" minlength="2" type="text" required>              </p>              <p>                  <label for="cemail">邮件地址</label>                  <input id="cemail" type="email" name="email" required>              </p>              <p>                  <label for="curl">URL</label>                  <input id="curl" type="url" name="url">              </p>              <p>                  <label for="ccomment">你的评论</label>                  <textarea id="ccomment" name="comment" required></textarea>              </p>              <p>                  <input class="submit" type="submit" value="提交">              </p>          </fieldset>      </form> |

完整代码

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="utf-8">      <title>jQuery validation plug-in - main demo</title>      <link rel="stylesheet" href="css/screen.css">      <script src="../lib/jquery.js"></script>      <script src="../dist/jquery.validate.js"></script>      <script src="js/messages\_zh.js"></script>      <script>      $.validator.setDefaults({          submitHandler: function() {              alert("submitted!");          }      });        $().ready(function() {          // validate the comment form when it is submitted          $("#commentForm").validate();            // validate signup form on keyup and submit          $("#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: "Please enter your firstname",                  lastname: "Please enter your lastname",                  username: {                      required: "Please enter a username",                      minlength: "Your username must consist of at least 2 characters"                  },                  password: {                      required: "Please provide a password",                      minlength: "Your password must be at least 5 characters long"                  },                  confirm\_password: {                      required: "Please provide a password",                      minlength: "Your password must be at least 5 characters long",                      equalTo: "Please enter the same password as above"                  },                  email: "Please enter a valid email address",                  agree: "Please accept our policy",                  topic: "Please select at least 2 topics"              }          });            // propose username by combining first- and lastname          $("#username").focus(function() {              var firstname = $("#firstname").val();              var lastname = $("#lastname").val();              if (firstname && lastname && !this.value) {                  this.value = firstname + "." + lastname;              }          });            //code to hide topic selection, disable for demo          var newsletter = $("#newsletter");          // newsletter topics are optional, hide at first          var inital = newsletter.is(":checked");          var topics = $("#newsletter\_topics")[inital ? "removeClass" : "addClass"]("gray");          var topicInputs = topics.find("input").attr("disabled", !inital);          // show when newsletter is checked          newsletter.click(function() {              topics[this.checked ? "removeClass" : "addClass"]("gray");              topicInputs.attr("disabled", !this.checked);          });      });      </script>      <style>      #commentForm {          width: 500px;      }      #commentForm label {          width: 250px;      }      #commentForm label.error, #commentForm input.submit {          margin-left: 253px;      }      #signupForm {          width: 670px;      }      #signupForm label.error {          margin-left: 10px;          width: auto;          display: inline;      }      #newsletter\_topics label.error {          display: none;          margin-left: 103px;      }      #main{      padding: 10px 10px 10px 10px;      }      </style>  </head>  <body>  <div id="main">      <form class="cmxform" id="commentForm" method="get" action="">          <fieldset>              <legend>请提供您的姓名、电子邮件地址（不会发布）和评论</legend>              <p>                  <label for="cname">姓名</label>                  <input id="cname" name="name" minlength="2" type="text" required>              </p>              <p>                  <label for="cemail">邮件地址</label>                  <input id="cemail" type="email" name="email" required>              </p>              <p>                  <label for="curl">URL</label>                  <input id="curl" type="url" name="url">              </p>              <p>                  <label for="ccomment">你的评论</label>                  <textarea id="ccomment" name="comment" required></textarea>              </p>              <p>                  <input class="submit" type="submit" value="提交">              </p>          </fieldset>      </form>      <div>  </body>  </html> |

## 效果图



# Bootstrap validation插件

## Bootstrap Validation插件下载

打开github，输入bootstrapvalidator，搜索，打开第一个。

此插件是基于Bootstrap框架的, 所以form表单的格式要符合Bootstrap框架的格式. 与Bootstrap框架相比, 仅仅是多了bootstrapValidator.js 以及bootstrapValidator.css文件.

## Bootstrap Validation插件使用

首先，在刚才布局的下面插入script标签，用jQery的加载语句作为初始。

然后用jquery的选择器获取表单元素 $('#defaultForm').bootstrapValidator(

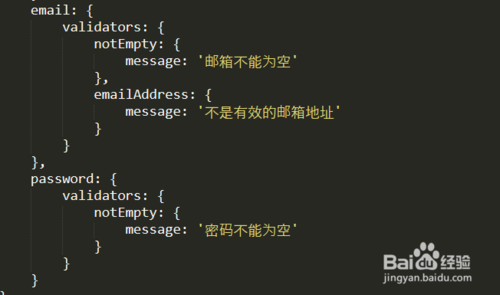
feedbackIcons：代表了不同结果时显示的图标。message是默认的提示信息。

[](http://jingyan.baidu.com/album/c843ea0bb2bf3c77931e4ad4.html?picindex=10)

接着是fields，这个里面就具体的写了username,email,password三个输入框的验证逻辑。以username为例。message代表基本的提示信息。 validators里面写了三个判别逻辑。notEmpty，当为空时处理的逻辑。stringLength输入框长度的限制，这里min，和max直接输入就可进行限制。regexp，正则表达式限制。regexp: /^[a-zA-Z0-9\_\.]+$/,这里简单写了一个只能包含“大小写字母数字下划线，点”的正则。

[](http://jingyan.baidu.com/album/c843ea0bb2bf3c77931e4ad4.html?picindex=11)

同理，这里再绑定email，和password。email比较特殊，我们可以直接用emailAddress来判断是否为有效的email地址，bootstrapvalidator已经为我们写好了逻辑。

[](http://jingyan.baidu.com/album/c843ea0bb2bf3c77931e4ad4.html?picindex=12)

# 参考文献

1. <https://www.cnblogs.com/max-hou/p/10986702.html>
2. <https://blog.csdn.net/sanbingyutuoniao123/article/details/51837113>
3. <https://my.oschina.net/enyo/blog/311566>
4. Bootstrap validation插件使用.

<https://jingyan.baidu.com/article/c843ea0bb2bf3c77931e4ad4.html>