

使用文档

1. 表单验证插件用于哪些地方

- 1、用户登录
- 2、用户注册
- 3、产品管理系统，发布产品，修改产品
- 4、客户管理系统，客户数据的录入，客户信息的修改。
- 5、其他系统信息的录入与编辑。

总结：只要有表单的地方，就可能会用到表单数据的验证。

2. 使用方法

插件版本：v1.15.1 需要jQuery版本：1.4.4+

2.1. 表单制作

首先需要制作一个表单，在这里老师就制作一个比较全的表单，包含了普通文本框，日期，数字，单选，复选，下拉，文件，多行文本，以后大家验证其他的表单可以使用同样的方法，特殊需求课进行自定义验证，举一反三，表单的代码位于素材文件夹下面，表单的界面如下（蓝色的文字是验证要求）：

The screenshot shows a web browser window displaying a form titled 'validation/index.html'. The form contains various input fields and controls. A green box highlights the validation requirements for several fields, which are listed in blue text:

- 姓名: 必填
- 班级: 非必填
- 个人主页: 必填
- 出生日期: 必填日期格式
- 性别: 必选
- 年龄: 必填数字
- 爱好: 至少选一个
- 电话号码: 必填数字
- 所在位置: 必须选择文件
- 头像上传: 选择文件 未选择任何文件
- 个人简介: 必填内容

Arrows point from the text '表单验证要求' (Form Validation Requirements) to the highlighted blue text. The form also includes a '提交' (Submit) button and a '重置' (Reset) button.

2.2. 默认规则

- | | |
|-----------------------|--|
| (1)required:true | 必填字段 |
| (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

2.3. 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).",  
  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}."),  
},
```

2.4. 中文提示

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>
```

用验证插件来验证此表单（代码以及需要的资源，上课的时候会给大家）：

- 1、制作一个表单，给表单取一个id
- 2、引入jquery库
- 3、引入验证控件jquery.validate.js
- 4、当验证的条件不满足的时候，提示默认是英文的，要有默认的中文提示，就引入多语言包js/messages_cn.js
- 5、写验证规则：给表单的元素取一个name，js验证代码里面增加对应name的验证规则，具体规则参考插件的文档 插件的文档一般来至官网、或者官网的demo\如果这些都没有、可以百度下 找相关文章，比较出名的插件网友一般都有总结。
- 6、写验证没有通过的提示。
- 7、当插件中内置的这些验证都不能满足我们的时候，我们可以自定义验证方法。

在html表单页面代码中引入jq库，验证插件，多语言配置文件：

```

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

```

在代码中写校验规则：

```

$(document).ready( function () {
  $( "#signupForm" ).validate({
    rules: {
      username: {
        required: true,
        email:true,
        minlength: 2
      }
    },
    messages: {
      username: {
        required: "请填写你的姓名",
        email:"电子邮件格式不正确",
        minlength: "Your username must consist of at least 2 characters"
      }
    }
  });

```

2.5. 自定义验证方法

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

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

radio和checkbox、select的验证：

常用方法及注意问题

1.用其他方式替代默认的SUBMIT

```
$(function(){
    $("#signupForm").validate({
        submitHandler:function(form){
            alert("submitted");
            form.submit();
        }
    });
});
```

可以设置validate的默认值，写法如下：

```
$.validator.defaults({
    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，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

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

如果一个页面中有多个表单，用

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

3.ignore：忽略某些元素不验证

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

4.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());  
}
```

使用方法：

```
<div class="panel-body">  
  <form id="signupForm" method="post" class="form-horizontal" action="">  
    <div class="form-group">  
      <label class="col-sm-4 control-label" for="username">用户名</label>  
      <div class="col-sm-5">  
        <input type="text" class="form-control" id="username" name="username" placeholder="用户名" />  
      </div>  
    </div>  
  
    <div class="form-group">  
      <label class="col-sm-4 control-label" for="psw">密码</label>  
      <div class="col-sm-5">  
        <input type="password" class="form-control" id="psw" name="psw" placeholder="密码" />  
      </div>  
    </div>  
  
    <div class="form-group">  
      <label class="col-sm-4 control-label" for="psw_c">确认密码</label>  
      <div class="col-sm-5">  
        <input type="password" class="form-control" id="psw_c" name="psw_c" placeholder="再次输入密码" />  
      </div>  
    </div>  
  
    <div class="form-group">  
      <label class="col-sm-4 control-label" for="age">年龄</label>  
      <div class="col-sm-5">  
        <input type="text" class="form-control" id="age" name="age" placeholder="年龄" />  
      </div>  
    </div>  
  </div>  
  
  <div class="form-group">  
    <label class="col-sm-4 control-label" for="email">Email</label>  
    <div class="col-sm-5">  
      <input type="text" class="form-control" id="email" name="email" placeholder="Email" />  
    </div>  
  </div>  
  
  <div class="form-group">  
    <div class="col-sm-5 col-sm-offset-4">  
      <div class="checkbox">  
        <input type="checkbox" id="agree" name="agree" value="agree" /> 同意协议  
      </div>  
    </div>  
  </div>  
  
  <div class="form-group">  
    <div class="col-sm-9 col-sm-offset-4">  
      <button type="submit" class="btn btn-primary" name="signup" value="Sign up">提交</button>  
    </div>  
  </div>  
</div>
```

```

<script type="text/javascript">
$.validator.setDefaults( {
  submitHandler: function () {
    alert( "提交成功!" );
  }
});

$(document).ready( function () {
  $( "#signupForm" ).validate( {
    rules: {
      username: "required",
      username: {
        required: true,
        minlength: 2
      },
      psw: {
        required: true,
        minlength: 5
      },
      psw_c: {
        required: true,
        minlength: 5,
        equalTo: "#psw"
      },
      age: {
        required: true,
        maxlength: 3,
        minlength: 1,
        number: true,
        digits: true,
        range: [5, 10]
      },
      email: {
        required: true,
        email: true
      },
      agree: "required"
    },
    messages: {
      username: {
        required: "请输入用户名",
        minlength: "用户名至少2位数"
      },
      psw: {
        required: "请输入密码",
        minlength: "密码长度至少5位"
      },
      psw_c: {
        required: "请再次输入密码",
        minlength: "密码长度至少5位",
        equalTo: "两次输入的密码不一致"
      },
      age: {
        required: "请输入年龄",
        minlength: "长度至少1位",
        maxlength: "最多三位",
        number: "必须是数字",
        digits: "必须是整数",
        range: "5到10之间的数字"
      },
      email: "邮箱的格式不正确",
      agree: "必须接受协议"
    },
    errorElement: "em",
    errorPlacement: function ( error, element ) {
      // Add the 'help-block' class to the error element
      error.addClass( "help-block" );
      // console.log(error.text());
      // console.log(!($("#msg>em:contains '"+error.text()+"").length>0));
      // if(!($("#msg>em:contains '"+error.text()+"").length>0))
      // {
      //   error.appendTo($("#msg"));
      // }

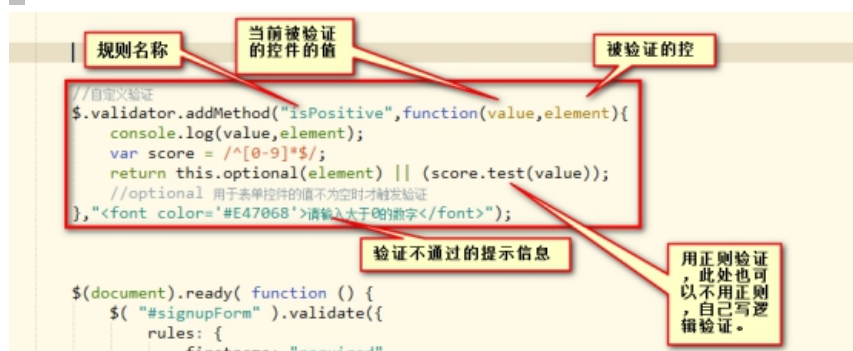
      if ( element.prop( "type" ) === "checkbox" ) {
        error.insertAfter( element.parent( "label" ) );
      } else {
        error.insertAfter( element );
      }
    },
    highlight: function ( element, errorClass, validClass ) {
      $( element ).parents( ".col-sm-5" ).addClass( "has-error" ).removeClass( "has-success" );
    },
    unhighlight: function ( element, errorClass, validClass ) {
      $( element ).parents( ".col-sm-5" ).addClass( "has-success" ).removeClass( "has-error" );
    }
  } );
});
</script>

```

如何自定义验证:

//自定义验证

```
$.validator.addMethod("isPositive",function(value,element){
    console.log(value,element);
    var score = /^[0-9]*$/;
    return this.optional(element) || (score.test(value));
    //optional 用于表单控件的值不为空时才触发验证
}, "<font color='#E47068'>请输入大于0的数字</font>");
```



```
$(document).ready( function () {
    $( "#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
            },
            age:{
                required:true,
                isPositive:true
            },
            agree: "required"
        },
    });
});
```

调用自定义规则, 非常的简单, 只需要在对应的字段后面加上 规则名称: true