

## 第 16 章 知问前端--验证插件

学习要点：

- 1.使用 validate.js 插件
- 2.默认验证规则
- 3.validate()方法和选项
- 4.validate.js 其他功能

主讲教师：李炎恢

官方网站：<http://www.ycku.com>

合作网站：<http://www.ibeifeng.com>

验证插件（validate.js），是一款验证常规表单数据合法性的插件。使用它，极大的解放了在表单上繁杂的验证过程，并且错误提示显示的完善也增加了用户体验。

### 一. 使用 validate.js 插件

官网下载：<http://bassistance.de/jquery-plugins/jquery-plugin-validation>

最重要的文件是 validate.js，还有两个可选的辅助文件：additional-methods.js（控件 class 方式）和 message\_zh.js（提示汉化）文件（实际使用，请使用 min 压缩版）。

第一步：引入 validate.js

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

第二步：在 JS 文件中执行

```
$('#reg').validate();
```

### 二. 默认验证规则

Validate.js 的默认验证规则的写法有两种形式：1.控件属性方式；2.JS 键值对传参方式。

默认规则列表

规则名	说明
required:true	必须输入字段
email:true	必须输入正确格式的电子邮件
url:true	必须输入正确格式的网址
date:true	必须输入正确格式的日期（IE6 验证出错）
dateISO:true	必须输入正确格式的日期(ISO)(只验证格式,不验证有效)
number:true	必须输入合法的数字(负数，小数)
digits:true	必须输入正整数
creditcard:true	必须输入合法的信用卡号，例如：5105105105105100
equalTo:"#field"	输入值必须和#field 相同

minlength:5	输入长度最小是 5 的字符串(汉字算一个字符)
maxlength:10	输入长度最多是 10 的字符串(汉字算一个字符)
rangelength:[5,10]	输入长度介于 5 和 10 之间的字符串)(汉字算一个字符)
range:[5,10]	输入值必须介于 5 和 10 之间
min:5	输入值不能小于 5
max:10	输入值不能大于 10
remote:"check.php"	使用 ajax 方法调用 check.php 验证输入值

//使用控件方式验证 “必填和不得小于两位”

```
<input type="text" class="required" minlength="2" name="user" id="user" />
```

注意：默认规则里设置布尔值的，直接写到 class 里即可实现。如果是数字或数组区间，直接使用属性=值的方式即可。而对于错误提示是因为，可以引入中文汉化文件，或直接替换掉即可。

//使用 JS 对象键值对传参方式设置

```
$('#reg').validate({
  rules : {
    user : {                                //只有一个规则的话，
      required : true,                      //直接 user : 'required',
      minlength : 2,
    },
  },
  messages : {
    user : {
      required : '帐号不得为空！',
      minlength : '帐号不得小于 2 位！',
    },
  }
});
```

注意：由于第一种形式不能设置指定的错误提示信息。我们推荐使用第二种形式，第二种形式也避免的 HTML 污染。

//所有规则演示

```
$('#reg').validate({
  rules : {
    email : {
      email : true,
    },
    url : {
```

```
        url : true,
    },
    date : {
        date : true,
    },
    dateIOS : {
        dateIOS : true,
    },
    number : {
        number : true,
    }
    digits : {
        digits : true,
    },
    creditcard : {
        creditcard : true,
    },
    min : {
        min : 5,
    },
    range : {
        range : [5, 10],
    },
    ranglength : {
        ranglength : [5,10],
    },
    notpass : {
        equalTo : '#pass',
    }
    },
});
```

//这里的 To 是大写的 T

### 三. validate()的方法和选项

除了默认的验证规则外，validate.js 还提供了大量的其他属性和方法供开发者使用。比如，调试属性，错误提示位置一系列比较有用的选项。

//jQuery.format 格式化错误提示

```
$('#reg').validate({
    rules : {
        user : {
            required : true,
            minlength : 5,
            ranglength : [5,10]
        },
    },
    messages : {
```

```
        user : {
            required : '帐号不得为空！',
            minlength : jQuery.format('帐号不得小于{0}位！'),
            ranglength : jQuery.format('帐号限制在{0}-{1}位！'),
        },
    },
});

//开启调试模式禁止提交
$('#reg').validate({
    debug : true,
});

//设置调试模式为默认，可以禁止多个表单提交
$.validator.setDefaults({
    debug : true,
});

//使用其他方式代替默认提交
submitHandler : function (form) {
    //可以执行 ajax 提交，并且无须 debug:true 阻止提交了
},

//忽略某个字段验证
ignore : '#pass',

//群组错误提示
groups : {
    myerror : 'user pass',
},

//显示群组的错误提示
focusInvalid : false,
errorPlacement : function (error, element) {
    $.each(error, function (index, value) {
        $('myerror').html($('myerror').html() + $(value).html());
    })
},

//群组错误提示，分开
groups : {
    error_user : 'user',
    error_pass : 'pass'
},
```

//将群组的错误指定存放位置

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

//设置错误提示的 class 名

```
errorClass : 'error_list',
```

//设置错误提示的标签

```
errorElement : 'p',
```

//统一包裹错误提示

```
errorLabelContainer : 'ol.error',  
wrapper : 'li',
```

//设置成功后加载的 class

```
success : 'success',
```

//使用方法加载 class 并添加文本

```
success : function (label) {  
    label.addClass('success').text('ok');  
},
```

//高亮显示有错误的元素，变色式

```
highlight: function(element, errorClass) {  
    $(element).fadeOut(function() {  
        $(element).fadeIn()  
    })  
},
```

//高亮显示有错误的元素，变色式

```
highlight: function(element, errorClass) {  
    $(element).css('border', '1px solid red');  
},
```

//成功的元素移出错误高亮

```
unhighlight : function (element, errorClass) {  
    $(element).css('border', '1px solid #ccc');  
},
```

//表单提交时获取信息

```
invalidHandler : function (event, validator) {  
    var errors = validator.numberOfInvalids();
```

```
if (errors) {
    $('#myerror').html('您有' + errors + '个表单元素填写非法! ');
}
},

//获取错误提示句柄，不用提交及时获取值
showErrors : function (errorMap, errorList) {
    var errors = this.numberOfInvalids();
    if (errors) {
        $('#myerror').html('您有' + errors + '个表单元素填写非法! ');
    } else {
        $('#myerror').hide();
    }
    this.defaultShowErrors();           //执行默认错误
},

//获取错误提示句柄，errorList
showErrors : function (errorMap, errorList) {
    alert(errorList[0].message);         //得到错误信息
    alert(errorList[0].element);        //当前错误的表单元素
},
```

#### 四. validate.js 其他功能

使用 remote:url, 可以对表单进行 ajax 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。

//使用 ajax 验证

```
rules : {
    user : {
        required : true,
        minlength : 2,
        remote : 'user.php',
    },
},
```

//user.php 内容

```
<?php
    if ($_GET['user'] == 'bnbbs') {
        echo 'false';
    } else {
        echo 'true';
    }
?>
```

注意：远程地址只能输出'true'或'false'，不能输出其他值。

//同时传递多个值到远程端

```
pass : {
    required : true,
    minlength : 6,
    remote : {
        url : 'user.php',
        type : 'POST',
        dataType : 'json',
        data : {
            user : function () {
                return $('#user').val();
            },
        },
    },
},
```

//user.php 内容

```
<?php
    if ($_POST['user'] != 'bnbbs' || $_POST['pass'] != '123456') {
        echo 'false';
    } else {
        echo 'true';
    }
?>
```

validate.js 提供了一些事件触发的默认值，这些值呢，大部分建议是不用更改的。

//取消提交验证

**onsubmit : false,** //默认是 true

注意：设置为 false 会导致直接传统提交，不会实现验证功能，一般是用于 keyup/click/blur 验证提交。

//设置鼠标离开不触发验证

**onfocusout : false,** //默认为 true

//设置键盘按下弹起不触发验证

**onkeyup : false,** //默认为 true

注意：只要设置了，在测试的浏览器不管是 false 还是 true 都不触发了。

//设置点击 checkbox 和 radio 点击不触发验证

**onclick : false,** //默认为 true

//设置错误提示后，无法获取焦点

**focusInvalid : false,** //默认为 true

```
//提示错误时，隐藏错误提示，不能和 focusInvalid 一起用，冲突
focusCleanup : true,                                //默认为 false
```

如果表单元素设置了 title 值，且 messages 为默认，就会读取 title 值的错误信息，我们可以通过 ignoreTitle : true，设置为 true，屏蔽这一个功能。

```
ignoreTitle : true,                                //默认为 false
```

```
//判断表单所验证的元素是否全部有效
alert($('#reg').valid());                            //全部有效返回 true
```

Validate.js 提供了可以单独验证每个表单元素的 rules 方法，不但提供了 add 增加验证，还提供了 remove 删除验证的功能。

```
//给 user 增加一个表单验证
```

```
$('#user').rules('add', {
    required : true,
    minlength : 2,
    messages : {
        required : '帐号不得为空！',
        minlength : jQuery.format('帐号不得小于{0}位！'),
    }
});
```

```
//删除 user 的所有验证规则
```

```
$('#user').rules('remove');
```

```
//删除 user 的指定验证规则
```

```
$('#user').rules('remove', 'minlength min max');
```

```
//添加自定义验证
```

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

```
//调用自定义验证
```

```
rules : {
    code : {
        required : true,
        code : true,
    }
},
```



# 感谢收看本次教程！

本课程是由北风网([ibEIFENG.COM](http://ibEIFENG.COM))

瓢城 Web 俱乐部([Ycku.com](http://Ycku.com))联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！