

jQuery 表单检验插件 validity

作者：kevin G 时间：2009-4-10

1 不需要编写 JavaScript 代码，只要按手册约定给表单元素设置好相关属性，即可实现自动验证，如：

```
<script src="jquery.js"></script>
<script src="jquery.validity.js"></script>
<form id="myform">
<input type="text" require="true" datatype="require" msg="不能为空" />
</form>
<script> $('#myform').checkForm(); </script>
```

如此，即可实现验证效果了，当文本框失败焦点（onblur）的时候，如果文本框为空，则会在其其后提示“不能为空”。

2 目前支持以下条件验证：

不能为空 require

电子邮件地址 email

电话号码（支持分机号） phone

手机号码 mobile

url 地址 url

身份证号码 idCard

货币 currency

正整数 number

邮编 zip

IP 地址 ip

QQ 号码（新版的 QQ 帐号可使用 Email，建议使用正则自定义验证） qq

正负整数 integer

小数 double

英文字母 english

纯汉字 chinese

用户名（数字及字母组合的字符串） userName

安全字符串（不能包含！#<>?* \$+ - =@% . * { } () | \ / ~ ^ 等字符的串） isSafe

输入长度限制（即输入长度必须在一定范围内） limit / limitB

日期 date

密码重复匹配 repeat

整数或小数必须在一定范围内（比如年龄必须在 18 至 60 岁之间） range

数值比较（比如年龄必须大于 18，但没有上限 / 至多购买 100 件等等，不能为 0，必须为 1 等等，相当于只有一边范围的） compare

复选框必须选中 m 到 n 个 group

下拉框必须选中 compare/require

Ajax 服务器端检验 ajax

自定义正则表达式 custom

多条件组合验证 require|limit

3 兼容 IE6/IE7/IE8/FireFox/Safari/Opera 主流浏览器

如何使用

validity 基本 jquery 扩展，所以，你得在你的 web 页面中加载 jquery 库及 validity 库，在 <head> 与 </head> 之间加入以下代码：

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

接下来，就是定义你的表单

```
<form action="">
<input type="text" require="true" datatype="require" msg=" 不能为空 " />
<input type="submit" value=" 提交 " />
</form>
然后，在这个表单后面加上验证语句：
<script>
$('form').checkForm(); //表单验证
</script>
```

这样，就实现了最基本的验证 —— 不能为空 。现在，当文本框失去焦点，或点击提交的时候，，如果值为空，则阻止提交表单，很简单吧？

但要注意的是 上面的文本框的属性中多了几项：`require` / `datatype` / `msg` 本插件就是依靠这几个自定义的属性来达到验证效果的，要达到验证，这三项是必须的

属性	值	说明
require	true/false	指明表单项是否需要检验，设置为 true 证明表单项不能为空，且要符合 datatype 属性指定的 规则。设置为 false 说明表单项可为空，但是如果值非空，也要通过 datatype 指定的规则。
datatype	非常多，不能全列出	指定表单要符合的验证规则，多个规则可以组合起来，用竖线分隔 如 require email limit
msg	一个字符串	指定检验失败时的提示信息 组合验证时，用竖线分割
success	true/false	成功提示信息是否显示
min	范围下限	具体含义与 datatype 的值有关
max	范围上限	具体含义与 datatype 的值有关
compare	比较表达式	指定要比较的表达式 当 datatype 值为 compare 时有效
regexp	正则表达式	当 datatype 为 custom 时，指定验证的正则表达式
url	一个 url 地址	ajax 验证时，指定 ajax 请求的 url 地址 可指定 method 属性，指定 ajax 请求的方式（get 或 post）
msgid	显示提示信息的 DOM 元素的 id	提示信息将会在此指定 id 的 DOM 元素中显示，而不会直接追加到表单元素的尾端

datatype 属性

要实现强大的验证，完全取决于 datatype 属性的设置，下面详细列出目前支持的其属性与值。

datatype 属性值	说明
require	不能为空
email	必须是合法的电子邮件地址
phone	必须是合法的固定电话号码（
mobile	必须为合法的手机号码
url	必须为 url 地址
idCard	合法的身份证号码
curreny	货币类型，等价于小数
number	正整数（包含
zip	合法邮编
ip	Ip
qq	qq
integer	正负整数
double	小数
english	英文字母
chinese	纯汉字
userName	用户名（字母及数字构成的字符串，
isSafe	安全的字符串（ 不能包含 ！#<>?* \$ + - =@ % . * { } () \ / ~ ^ 等字符的串 ）
limit	长度限定在指定的范围里， gbk 编码下一个汉字视为两个字符， utf-8 编码中，一个汉字视为 3

	个 为了达到些规则的验证效果，你还应该为表单设置 min 及 max 属性，如 <code><input require="true" d max="15" msg=" 必须是 3 到 15 个字符 " /></code> 这里的 min max 指的是字符串的长度范围，而不是值的大小范围
limitB	一个汉字视为两个字符的检验，和 limit 规则类似，也需要指定 min 及 max 属性
date	日期格式（也可以指定 format 属性，即可日期格式）如 <code><input require="true" datatype="date" format="ymd" msg=" 日期非法 " /></code>
repeat	必须和指定的表单值一致，常常用来做密码确认，针对此检验，你必须设置表单项的 to 属性，其值为要匹配的表单元素的 name 属性的值，如： <code><input require="true" datatype="limit" min="6" max="16" msg=" 密码应该在 6 至 16 位之间 " to="pwd"/></code> 则说明 此项的值必须和 name 为 “pwd” 的元素的值一致
range	值必须在一定范围内，也应该设置 min max 属性，但是 min/max 在里指的是值大小的范围，而不是与 limit 的 min/max 的含义是不一样的
compare	数值比较 必须额外设置一个 compare 属性，指定要比较的表达式 如 <code><input require="true" datatype="compare" compare=">100" msg=" 请输入大于 100 的数字"</code> 类似的，你还可以将 compare 设置为 “!=0”， “>=123”， “<=123”， “==123”
group	复选框 / 单选框必须选中 m-n 个 必须设置 min 属性，表示复选框必须至少选中 m 个，最多选中 n 个，当然，对于单选框， min==max==1 即可 max 属性如果不指定，则不做 “最多” 限定
compare/require	下拉框必须选中，也可以使用这两个值来限定
custom	自定义正则检验 当使用自定义正则时，你必须指定 regexp 属性，regexp 为你指定的正则表达式，如 <code><input datatype="custom" require="true" regexp=".+V\$" msg=" 必须以 / 结束 " /></code>
ajax	ajax 检验，必须指定 url 属性，即 ajax 请求的地址，如 <code><input type="text" name="username" require="true" datatype="ajax" url="/api.php?action=check" /></code> 那么，构造的最终请求 URL 为 “/api.php?action=check&username=XXX”（其中 XXX 即文本框的值） 可指定 method 属性，指定 ajax 请求的方式（get 或 post）如 method=“post” 那么，如果约定 ajax 检验成功与否呢？ 如果服务器端返回字符串 “success”，则证明检验成功，否则， ajax 检验失败，如果设置了 msg，则提示 msg。否则直接显示服务器返回的信息。

组合验证

何为组合验证？ 组合验证即为普通验证规则的叠加，只有全部满足所有的各个规则，验证方可通过。

如

```
<input type="text" name="username" require="true" datatype="limit|english|ajax" min="3" max="8" msg=" 用户名应该在 3 到 8 位英文字母组合|必须为英文字母|用户名已经被占用 " url="/ajax.php"/>
```

上面的意思是：必须为 3 到 8 位，且为英文字母，并且要通过服务器 ajax 检验。注意， msg 与 datatype 均用竖线分隔，即表示了每个验证规则

失败后的相应的提示信息。那么，如果上面的 msg=“ 请使用其它用户名 ”，那么无论何种验证失败，均提示 “ 请使用其它用户名 ”。

在实际情况中，请灵活运行组合验证，会达到意想不到的效果的！

msgid 属性

msgid 属性指定 提示信息显示到指定 id 的 DOM 元素中去。如果不指定 msgid，则会在表单项的后面追加显示信息。

验证成功与失败时的图示

要达到更好的提示图示，你应该定义以下几个 C S S 类

- .error-info 验证失败后提示信息的样式
- .error-Input 验证失败后 ， 表单项本身的样式
- .suc-info 验证成功后， 提示信息的样式（往往是一个表示正确的图标）

方法：

\$(document). checkForm(); 表单验证
\$(document). Check_submit() 按 input 排列的顺序来显示错误信息 适用于表单顶部显示错误信息

验证成功与失败时的图示代码修改

```
Line 209:
jQuery("<em class=\"form-info "+validity.errorTip+\">&nbsp;&nbsp;&nbsp;</em>").html(errmsg).insertAfter(element);
Line 218:
jQuery("<em class=\"form-info "+validity.validTip+\">&nbsp;&nbsp;&nbsp;</em>").insertAfter(element);
Line 243:
jQuery("<em class=\"form-info "+this['errorTip']+\">&nbsp;&nbsp;&nbsp;</em>").html(errmsg).insertAfter(element);
Line 318:
jQuery("<em class=\"form-info "+validity.validTip+\">&nbsp;&nbsp;&nbsp;</em>").insertAfter(obj);
```

演示 CSS

<div><div>.form-info /*基本样式*/</div><div>background-image:url(..images/ico.gif);</div><div>background-repeat:no-repeat;</div><div>background-position:1000px 1000px;</div><div>font-style:normal;</div><div>line-height:25px;</div><div>padding:0 5px 0 18px;</div><div>position:absolute;</div><div>right:0;</div><div>width:160px;</div></div>	<div><div>.error-info /*错误样式*/</div><div>background-color:#ffebe8;</div><div>background-position:3px -26px;</div><div>border:1px #dd3c10 solid;</div><div>color:#d90000;</div></div>
<div><div>.suc-info /*成功样式*/</div><div>background-color;;</div><div>background-position:3px 0;</div><div>color:#3c8036;</div></div>	<div><div>.error-input /*错误 input 样式*/</div><div>background-color;;</div><div>border:1px #dd3c10 solid;</div></div>

备注

<div><div>图 片</div><div>ico.gif</div></div>	<div><div></div><div></div></div>
---	-----------------------------------