

# jQuery html5Validate基于HTML5表单验证插件

这篇文章发布于 2012年12月20日, 星期四, 13:43, 归类于 [jQuery相关](#)。阅读 248242 次, 今日 27 次 [110 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com>  
本文地址: <http://www.zhangxinxu.com/wordpress/?p=2857>

更新于2018-05-08

经过不断锤炼, LuLu UI已经正式开源啦: <https://github.com/yued-fe/lulu>

其中表单验证文档参见: <https://l-ui.com/content/apis/validate.html>

可以模块化调用, 也可以作为插件直接直连使用, 文档底部有单独使用的demo演示。

多年数个千万PV项目检验与打磨, 体验和细节都非常值得称道。

下面这些以前写的更新提示可以忽略啦~

更新于2016-07-08

又有不少人问了, 开源还需要时日。不过大家现在是可以使用的, 下面是最简单使用, 需要seajs:

```
<script src="http://qidian.gting.com/lulu/theme/modern/js/plugin/sea.js"></script>
<script>
seajs.config({
  'base': 'http://qidian.gting.com/lulu/theme/modern/js'
}).use(['common/ui/validate'], function(Validate) {
  new validate($('#form'), callback, options);
});
```

然后, 我也把文档给独立出来了, 您可以狠狠地点击这里: [海量用户产品验证过的表单验证组件demo兼文档](#)

别看写得很多, 其实基本使用很简单的。后面的文档只要是面向复杂定制的。

如果上面的使用还是觉得不够简单, 可以试试直接使用下面的JS地址:

```
<script src="http://qidian.gting.com/c=/lulu/theme/modern/js/common/ui/validate.js,/lulu/
theme/modern/js/common/ui/ErrorTip.js,/lulu/theme/modern/js/common/ui/Follow.js"></script>
```

可以直接使用包装器方法, 例如:

```
$(form).validate(callback, options);
```

更新于2016-02-25

前面提到的新版目前线上已经可以访问:

[http://mp.gting.cn/old\\_mp/assets/js/common/ui/Validate.js](http://mp.gting.cn/old_mp/assets/js/common/ui/Validate.js)

[demo体验狠狠地点击这里](#)

解决了之前组件如下不足:

1. 自定义验证顺序有问题;

2. 即时验证支持没内置；
3. 不能随意定制验证规则；

等等。

今年会开源，大家稍安~

更新于2014年12月18日 0:35

本插件已经迁移至Github上更新与维护: <https://github.com/zhangxinxu/html5Validate>

## 一、前言

前3篇文章实际都是为本文做铺垫的，如果以下棋表示，前三篇是普通走棋，本篇是将军！

目前市面上有不少表单验证插件，看似强大，实在糟糕！

总结下，有以下一些问题：

### 1. 过多干预

包括：改变了表单元素UI, 为表单元素绑定过多事件等



### 2. 布局等限制

包括：需要特定结构的布局，需要特定的类名或者ID

```
<td><input type="text" value="" name="
<td><div class="Validform-checktip"></div></td></tr></table>
```

### 3. 学习成本

包括：N多元作者自定义的属性，或者自定义的特定的数据结构

```
fields: {
  '#yourName': {
    required: true,
    message: '名字是必须的'
```

### 4. 可用性

当JS出现错误的时候而无法正常运作的时候，验证就是聋子的耳朵——摆设，即使在现代浏览器下也是如此。

面向未来的表单验证

### 1. 验证驱动

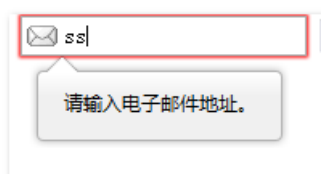
验证信息HTML驱动，例如HTML5中 `required` , `pattern` , `multiple` 等

### 2. 验证形式

非即时响应，submit验证，如Chrome浏览器的处理；或者弱即时响应，如Firefox浏览器仅仅红色外发光。

### 3. 验证交互

浮动形式，尖角指示。



换言之，所谓面向未来的表单验证，是遵循W3C HTML5规范的表单验证，我们可以从目前领先的浏览器中看到大致雏形。而本文所有展示的html5Validate表单验证插件，就是基于这个未来设计的。

## 二、html5Validate概述

html5Validate插件的验证机制、交互形式甚至形式与Chrome浏览器HTML5表单内置验证走的很近。在使用的时候，就是写写原生的HTML5表单代码。我只想说：走阳光大道和过独木桥的感觉是完全不一样的。

举个简单例子，一个邮箱验证，HTML5代码表示应该是下面这个样子：

```
<input type="email" required>
```

好巧的是，使用html5Validate进行表单验证的时候，也是使用上面这段HTML代码！🤔

类似下面的绑定：

```
$("form").html5Validate();
```

于是，您在提交表单的时候会（在各个浏览器下）看到这样子的提示：



跟Chrome浏览器下的提示文字近似：



html5Validate支持我所知的HTML5验证相关的东西，如 `type="email"`，`type="number"`，`type="tel"`，`type="url"`，`step`，`min`，`max`，`required`，`pattern`，`multiple` 等，并有一些本地化扩展，如增加了 `type="zipcode"` 邮编等，支持 `type="hidden"` 的完整验证（HTML5中是忽略的），支持多type共存，例如 `type="email|tel"`，可以让文本框输入邮箱或者手机号码。

`type="date"`，`type="hour"`，`type="password"` 等因为不同网站规则不一样，因此，没有放在html5Validate中，不过，您可以很简单地进行扩展，使您的项目支持之，这个后面会介绍（参见 part 9-4）。

为了满足实际开发需求，额外增加了四个自定义属性值：`data-key`，`data-target`，`data-min`，`data-max`。具体何用，下面会详细讲解。

支持自动的全角转半角。

**注意：** `type="submit"`，`type="reset"`，`type="file"`，`type="image"` 以及 `disabled` 的表单元素没有验证性可言，因此，下面所说的表单元素，并不包括他们。

兼容性

html5Validate通过jQuery1.4+测试，支持正常IE6-IE10浏览器，FireFox, Chrome等现代浏览器。

### 三、demo、使用以及资源下载

以下灰色文字内容可忽略，已过时~

您可以狠狠地点击这里：[html5Validate表单验证jQuery插件测试demo](#)

下载

未压缩版JS：[jquery-html5Validate.js](#)

压缩版JS：[jquery-html5Validate-min.js](#)

zip源文件打包下载（含测试demo）（右键-[目标|源文件]另存为）：[jquery-html5Validate.zip](#)

更新于2014年12月18日 0:35

本插件作为jQuery插件迁移计划先头兵，已经迁移至Github上更新与维护：<https://github.com/zhangxinxu/html5Validate>

资源下载、或者使用遇到什么问题，可以去Github上提问，我会及时更新与建设，也欢迎大家一起参与。

#### 使用

1. 引用jQuery框架，示意：

```
<script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
```

2. 引用html5Validate插件，示意：

```
<script src="http://www.zhangxinxu.com/study/js/mini/jquery-html5Validate-min.js"></script>
```

3. 绑定调用：

```
$.html5Validate(callback, options);
```

示意，假设测试表单 `id` 为 `html5Form`，则有：

```
$("#html5Form").html5Validate(function() {  
    // 全部验证通过，该干嘛干嘛~~  
});
```

//xxx: 还有一些实际应用的例子将会在本文后半部分展示。

### 四、可选参数以及详解

可选参数见下表：

参数名称
默认值
简单释义
novalidate
true
布尔型。是否取消现代浏览器的内置验证
submitEnabled
true
布尔型。表单中禁用的提交按钮是否使之可用

labelDrive
true
布尔型。是否优先使用label标签中的文字作为提示关键词
validate
function() { return true; }
包含返回值的函数，插件自带验证以外的其他验证

## 1. 参数novalidate

既然html5Validate插件验证与原生HTML5验证公用一套HTML代码，显然就存在共存的问题。

如果没有设计这个参数 `novalidate` 或者 `novalidate` 值为 `false`，则浏览器内置表单验证优先，全部pass之后才轮到插件进行验证，也就是说，两者是没有冲突的。其实，原本我是不打算设计这个参数的，因为，我觉得浏览器内置的验证以及交互效果很赞的，但是，IE10的出现让我速速打消了这个念头，因为，丫丑得我根本无法直视——大红的粗框框！！



`novalidate` 为HTML5表单内置的属性（W3C草案），可以让现代浏览器（IE10+, FireFox, Chrome, Opera等）默认不对表单做验证（忽略 `required`，`pattern` 等）。html5Validate插件默认 `novalidate: true` 也就是说，其默认对包装器元素添加了 `novalidate="novalidate"`，以阻止浏览器的默认验证，下图代码为JS生成之后所截：

```

<input type="text" value="1.1" novalidate="novalidate" />
<input type="text" value="2" novalidate="novalidate" />

```

这就是为什么测试demo在FireFox等浏览器下不出现内置提示效果的原因。

可能您有这样的需求，您希望除了IE10以外的现代浏览器都使用浏览器自带的验证规则以及提示效果，怎么办，可以像下面这样？

```

var isIe10 = typeof document.msHidden !== "undefined";
$("form").html5Validate($.noop, {
    novalidate: isIe10? false: true
});

```

## 2. 参数submitEnabled

这个参数的设计是很有必要的。

很多时候，我们的表单是Ajax提交的，当页面加载不是很及时的时候——如表单元素呈现，但验证脚本未绑定——，用户回车一个输入框，悲剧来了，默认表单的 `submit` 被触发了，而不是ajax提交.....

我不清楚其他团队是怎么处理的，我处理这类问题是表单元素的提交按钮默认 `disabled`，在表单验证脚本绑定OK之后，去除 `disabled` 属性，使一切正常。

`submitEnabled` 参数的作用就是让表单中的禁用的submit性质的按钮可用。

例如，测试demo中，默认按钮是 `disabled` 的，这样，就算JS因网络等原因被block了，用户也无法提交表单，避免造成更加不好的体验。

```
type="submit" class="submit" value="提交测试" disabled />
```

等验证事件初始化完毕，自然，`disabled` 的按钮被 `enabled` 了。

对于原生 `submit()` 的表单或者可以 `submit()` 的表单，此参数酱油，您可以无需 `disabled` 提交按钮。

### 3. 参数labelDrive

顾名思义，`label` 驱动。该参数针对错误提示框中的文字内容。

这是与浏览器内置提示文字不一样的地方，优先`label`标签文字提示。什么意思？

`html5Validate` 插件有两套文字提示机制。一种是`label`标签驱动，其次是内置固定提示文字，如“请填写此字段”。

//xxx: 对于单复选框以及按钮，由于点击它们对应`label`标签会触发选中或者按钮事件，因此，对于这些元素，`labelDrive`参数是完全的酱油 – 提示文字与`label`毫无关系。对于下拉框而言，其在隐藏状态下，提示文字可以从`label`获取；非隐藏状态也是酱油。

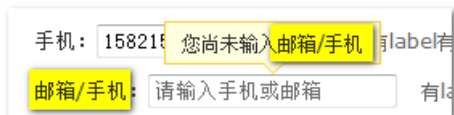
何为 `label` 标签驱动？举个例子：

```
<label for="email">邮箱: </label><input type="email" id="email" required>
```

上面HTML的提示文字就是：



如果对应`label`标签中的文字是“邮箱/手机”，则提示文字是：



也就是说“`label` 标签驱动”就是把 `label` 标签中的文字作为提示关键字，这样子的提示更智能，同时也不会增加额外的HTML代码。

该文字如何获取？

- ①找到与表单元素id对应的`label`元素( `for` 关联)，如果没有则应用内置提示文字规则；
- ②获取 `label` 标签中的文字，过滤星号(\*)，中文冒号(:)和英文冒号(:)，剩下的文字作为提示关键字出现在提示框中，如果输入框内容为空则提示“您尚未输入\*\*\*”；格式不准确则提示“您输入的\*\*\*格式不准确”。
- ③如果 `label` 标签文字与表单元素的 `placeholder` 值一样，则此 `label` 标签酱油。
- ④如果有多个`label`元素，符合要求的提示文字会累加。
- ⑤如果最后得到的 `label` 标签文字为空，同样应用泛泛的提示文字（如“请填写此字段”或“内容格式不符合要求”）。

#### 小技巧

你可以隐藏`label`标签（如demo中两个拖选）；或部分隐藏，如demo中评论部分：

```
<label for="comment">内容: </label><textarea id="comment" rows="5" required data-min="5" da
```

```
ta-max="100">
```

提示文字为“您尚未输入评论内容”，而不是“您尚未输入评论”。

#### 4. 参数validate 新增于2013-06-19

新增参数 `validate` . 实际开发的时候，各类验证需求层出不穷。例如，今天@jason就在评论中提到，是否可以指定多选就3项。这不属于HTML5规范中验证，也不是常规验证需求，在插件中添加此方法是有违插件的设计原则的，因此，插件内部是不会支持这类验证的。

在我的实际项目中，MooTools版的 `html5Validate` 插件是有一个可选参数 `validate` ，参数值为函数，就是一个额外验证的函数，专门对付一些特殊的验证需求。今天正好也为jQuery版增加下这个参数。

插件内部是如此运作的，先执行插件内置的验证，然后，再执行 `validate` 这个验证函数，如果返回值为 `true` ，则执行我们相对表单进行的回调操作。

言语苍白，demo明了。您可以狠狠地点击这里：[html5Validate新增validate参数demo](#)

demo中有两个额外的验证：一是前后密码需要一致；二是多选项至少选择3个。大致截图如下：



然后，相关JS如下：

```
$("#form").html5Validate(function() {
    alert("验证全部通过!");
    this.submit();
}, {
    validate: function() {
        // 下面这些就是额外的一些特殊验证
        if ($("#pwdCheck").val() !== ($("#pwd").val()) {
            $("#pwdCheck").testRemind("前后密码不一致");
            return false;
        } else if ($("#input[type='checkbox']:checked").length < 3) {
            $("#checkBox").testRemind("至少选择3项");
            return false;
        }
        return true;
    }
});
```

注意需要有Boolean类型返回值，验证出错返回 `false` 验证就会停止，效果达到。

## 五、四个自定义属性

HTML5的表单验证html5Validate自然支持，但是，仅仅HTML5的东西貌似不能完全支持实际的各类需求，因此，html5Validate中还有4个自定义属性(就是前面提到的 `data-key`，`data-target`，`data-min` 和 `data-max`)，帮助覆盖95%+的验证需求。

### 1. 自定义属性“data-key”和“data-target”

这两个参数是为模拟表单（或称为“自定义表单”）设计的。

何为模拟表单？如自定义的下拉框，或者星星评分选择等.....

#### ① “data-key”作用

模拟表单往往都是纯鼠标操作的，此时，类似“请填写此字段”或者“您输入的.....”就显得不准确，不准确的关键就是这里的操作动词“填写”与“输入”。“data-key”的值就表示这个关键动词，如demo中的第二个拖选：

```
<label for="serverScore" class="dn">服务评分</label>
<input type="hidden" id="serverScore" required data-key="拖选" />
```

于是，提示的时候，就是：

注意：隐藏性质表单的“data-key”是可以缺省的，对于一般的隐藏输入框，默认“data-key”值为“输入”；对于隐藏的单复选框或下拉框，默认“data-key”值为“选择”。

#### ① “data-target”作用

从可用性上讲，模拟表单都有一个对应的隐藏不可见的真实表单元素，或 `visibility:hidden` 的 `select` 下拉框，或 `hidden` 类型的 `input` 框，或者 `display:none` 的 `radio` 们。

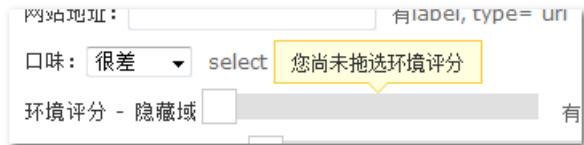
要知道，这些隐藏表单的位置是捕获不到(`display:none`)或者位置不准确的。因此，浮动提示框的



位置无法确定；就算浮动框位置确定了，指向了一个看不见的东西，用户也会很奇怪的。面对这种情况，`"data-target"` 应运而生。

`"data-target"` 可将黄色提示框的目标元素从真实表单元素转移到模拟表单元素（其值正是模拟表单元素的id(或className)）。如果 `"data-target"` 不存在，或其值对应不到元素，则 `alert` 弹出提示，如上面这个截图。

如果 `"data-target"` 对应元素存在，则就会像下面这样提示（demo中第一个拖选）- 乾坤大挪移：



HTML代码如下：

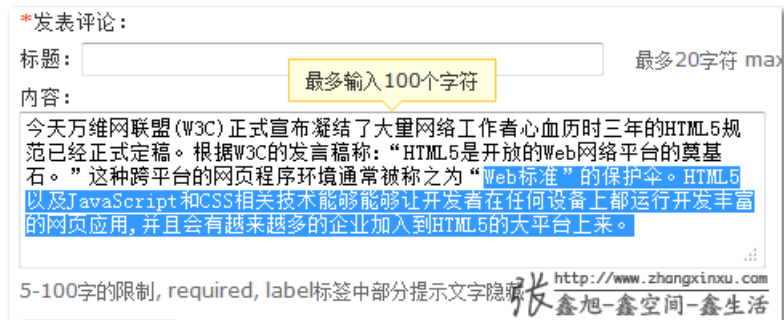
```
<span id="hiddenRemind" class="bar_bg"><i class="bar_btn" data-rel="envirScore"></i></span>
<label for="envirScore" class="dn">环境评分</label>
<input type="range" id="envirScore" class="dn" required data-key="拖选" data-target="hiddenRemind" />
```

## 2. 自定义属性“data-max”和“data-min”

这两个自定义属性要好理解多了，一般用在文本域上或昵称文本框上，用做最多字符个数和最小字符个数限制。例如demo最后文本域：

```
<textarea id="comment" rows="5" data-max="100" data-min="5" required></textarea>
```

表示，评论内容需要5-100个字符。无论字符不足或者是超出，都会出现提示，其中，当内容超出的时候，html5Validate会自动帮你选中溢出部分的文字，例如下面截图：



# 六、一些验证类型的说明

## 1. type=number, 整数与小数

`type=number` 文本框（数值文本框）的验证相对复杂点，这里有必要说下。数值文本框有 `step`, `min`, `max` 原生属性。`min`, `max` 表示数值最小值和最大值，`step` 缺省表示输入值必须整数，`step` 为 `0.1` 表示输入值最多1位小数（`.00` 结尾除外）。

具体验证规则参见之前下的一步棋：[HTML5 number类型文本框step属性的验证机制](#)

## 数值文本框的UI

Chrome, Opera等浏览器下，数值文本框会有上下数值增减箭头；很多固执的设计师是无法容忍这个的：“the ui is so terrible!”



因此，实际上，不少团队的数值类型框还是老旧的 `type="text"` 以避免浏览器自带的上下箭头。OK，冲突来了，`html5Validate`需要HTML5代码做验证，实际实现不能使用HTML5的东西，怎么办？

很简单，在类型后面加个空格，例如，不是设置 `type="number"`，而是设置 `type="number "` 或者 `type="number|"`。//zxx: 管道符用来支持多type公用

`html5Validate`会自动过滤最后的空格，因此，相关的验证没有任何影响。

### 数值文本框与multiple

数值文本框不支持 `multiple`，此规则与浏览器一致。`email` / `tel` 等类型支持 `multiple`，多个邮箱或手机号使用逗号(,)分隔。

### 2. 不得不提的type=range

`type=range` 与 `type=number` 是近亲，均有 `step`，`min`，`max` 原生属性。然而，目前以及今后很长的一段时间，在web项目上，就验证而言 `type=range` 几乎没有用武之地，原因是UI限制。何解？

如果浏览器支持 `type=range`，则用户完全是拖选操作（不能输入），几乎不存在范围溢出的情况，验证也几乎就是摆设；如果浏览器不支持 `type=range`，其UI就是个普通输入框，其内部验证机制与 `type=number` 无异，如果撇开语义化以及可访问性不谈，是不是可以直接使用 `type=number`？

因此，个人看来，`type=range` 只能作为隐藏的表单元素——为自定义表单元素服务，如demo中第一个拖选。也正因为这个原因，`html5Validate`并未重视 `type=range`。

### 3. type=radio以及 type=checkbox的验证规则

无论是单选框还是复选框，只有是否必选的验证(`required`)。

对于同一组单选框（name值一致），只要有一个单选框设置 `required` 就可以了（为了性能，也建议这么做）。单选框的提示文字固定，取自Chrome浏览器，为“请选择一个选项”。

对于复选框，哪个必选，就哪个需要设置 `required`。其提示文字也是固定的，也是参考的Chrome浏览器，为“如果要继续，请选中此框”。不过，个人举得，复选框必选设置使非常让人蛋疼菊紧的。

## 七、内置插件testRemind介绍

[jquery-html5Validate.js](#)中还内置另外一个插件 – `testRemind`。就是出现那个黄色提示框框的插件。

用法如下：

```
$(...).testRemind(content, options);
```

其中，`content` 为提示的内容，可以包含HTML字符串；`options` 为可选参数。

可选参数

参见下表：

参数名称
默认值
简单释义

size
6
数值。提示框尖角的尺寸大小
align
“center”
特定字符串。提示框三角的位置，其他可选值为“left”和“right”
css
<pre>{   maxWidth: 280,   backgroundColor: "#FFFE0",   borderColor: "#F7CE39",   color: "#333",   fontSize: "12px",   padding: "5px 10px",   zIndex: 202 }</pre>
样式对象。提示框框的可变样式。

## 参数含义

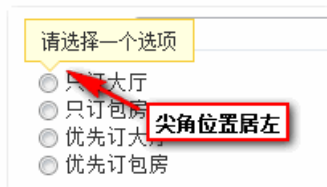
### ① 参数size

没什么好说的，提示框三角的大小。

### ② 参数align

没多少好说的，三角在框框的左边、中间还是右边。

在html5Validate中，一般的表单元素尖角都是在中间的，但是，单选框和复选框尖角实在左侧，如下图所示，这个设置目前暂不可变：



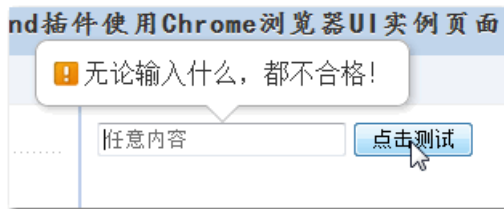
### ③ 参数css

该选项可以让你修改提示框的UI。

例如，通过如下的设置，我们可以让提示框框的效果长得跟Chrome浏览器近似：

```
$("#input").testRemind("无论输入什么，都不合格!", {
  size: 10,
  css: {
    padding: "8px 10px",
    borderColor: "#aaa",
    borderRadius: 8,
    boxShadow: "2px 2px 4px rgba(0,0,0,.2)",
    background: "#fff url(chrome-remind.png) no-repeat 10px 12px",
    backgroundColor: "#fff",
    fontSize: 16,
    textIndent: 20
  }
}).get(0).focus();
```

于是，有类似下面效果（截自FireFox 17）：



您可以狠狠地点击这里：[testRemind Chrome UI测试demo](#)

### 几乎酱油的可选参数

在实际使用的时候，testRemind方法的这几个可选参数几乎是酱油的，因为html5Validate方法并未暴露相关参数可以修改弹出框框的UI，这是我故意这样设计的。

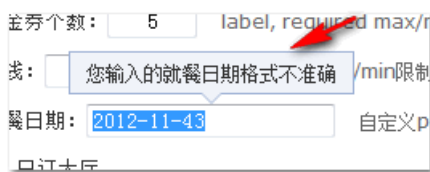
一来自己不希望html5Validate与testRemind方法有过多的耦合，因为这会让我觉得复杂，而产生不安；其次，对于一个项目而言，提示框框效果应该是整站统一，样式设置的限制未尝不是一件好事。

不过，此处小标题是“几乎酱油”，言外之意就是您还是可以通过一定的外部设置，改变提示框框的样式的。本插件JS中暴露了一个名为 `$.testRemind.css` 的对象，您可以对其加加减减来控制提示框的UI。

举个板栗，如下代码：

```
$.testRemind.css = {  
    borderColor: "#a0b3d6",  
    backgroundColor: "#f0f3f9"  
};  
$("#testForm").html5Validate();
```

则，提示框框就会变成下面这幅屏模样：



总结为：testRemind方法的CSS有限可选参数 `options` 中的css对象，如果没有，使用 `$.testRemind` 中的css对象。

注意：提示框出现后，点击提示框以外的页面任意位置，或在表单元素focus状态下触发键盘，或改变浏览器的尺寸，提示框都会因此隐藏。

### \$.testRemind.display

除了 `$.testRemind.css` 外，还有一个 `$.testRemind.display` 属性，Boolean值(true/false)，用来反映当前是否有弹框显示。关键时刻，这个还是很管用的。

//zxx: 下面为广告~~注意不要勿点~~嘻嘻~~



## 八、高阶使用

`$(...).html5Validate()` 方法是针对表单元素的，已经自动绑定了submit事件，并阻止了默认表单提交。然而，各类交互需求千变万化，自以为是的自动绑定可能满足不了苛刻的丈母娘，此时我们需要另外的处理，这就是这里要说的html5Validate的高阶使用。

html5Validate中有个名为\$.html5Validate的对象，其中有一些方法，如验证元素是否为空(`$.html5Validate.isEmpty(ele)`)，是否合法(`$.html5Validate.isRegex(ele, regex, params)`)等，因为我比较懒，这几个方法不介绍。这里着重要提的是下面这个方法：

```
$.html5Validate.isAllpass(elements, options);
```

顾名思义：HTML5验证 – 是否全部通过！

### 1. 参数elements

`elements` 参数必需，可以是：`form` 节点，或者jQuery包装 `form`，或者是需要验证的元素们。代码示意如下：

```
$.html5Validate.isAllpass(document.getElementsByTagName("form")[0]);
$.html5Validate.isAllpass($("#form"));
$.html5Validate.isAllpass($("#input, textarea, select"));
```

### 2. 可选参数options

`options` 可选参数目前就一个，就是 `labelDrive`，html5Validate方法中的参数 `labelDrive` 实际上是喂给 `$.html5Validate.isAllpass` 使用的。

`$.html5Validate.isAllpass` 方法与表单没有任何关系，因此，我们可以更加灵活地应用在其他一些特殊场景上，举个简单例子，通过按钮点击事件验证并触发相关操作：

```
订阅优惠信息
<input type="email" class="email" placeholder="输入邮箱，如a@b.com" required />
<a href="javascript:" class="button">订阅</a>
```

然后，类似下面的验证处理：

```
$("#button").bind("click", function() {
    var email = $(".email");
    if ($.html5Validate.isAllpass(email)) {
        // 邮箱验证通过，该干嘛干嘛~~
    }
});
```

您可以狠狠地点击这里：[\\$.html5Validate.isAllpass简单应用demo](#)



## 九、其他可用的全局方法、对象及扩展

### 1. 全半角转换DBC2SBC方法

方法名称就是 `DBC2SBC`，一个字符串参数，返回新的半角字符串，例如：

```
DBC2SBC("1 3 2 0 8 0 3 3 6 2 1"); // 结果是"13208033621"
```

## 2. 检测元素是否可见

包装器方法，名称为 `$(obj).isVisible()`，返回Boolean值，`true/false`。

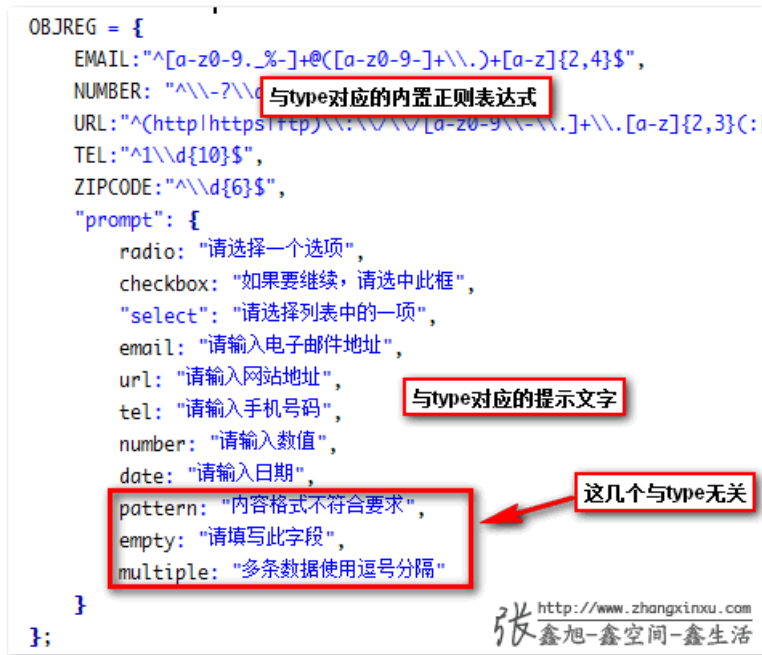
这里的不可见指的是不能被focus的隐藏，包括 `type="hidden"` 隐藏域，`display:none` 以及 `visibility:hidden`，屏幕外隐藏以及透明度为0等不在其中。

## 3. Boolean型属性检测

包装器方法，名称为 `$(obj).hasProp(prop)`，返回Boolean值，`true/false`。

所谓Boolean型属性指的是HTML5中 `required`，`multiple`，`novalidate` 等属性有，值缺省的属性。具体可参见“HTML5 Boolean属性值的JS获取”一文。

## 4. 全局对象OBJREG及扩展

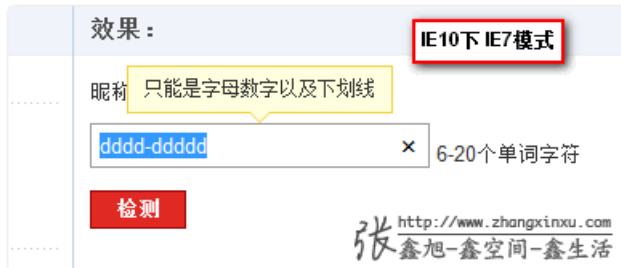


```

$(".button").bind("click", function() {
    var nickname = $(".nickname");
    if ($.html5Validate.isAllpass(nickname)) {
        alert("验证通过! ");
    }
});

```

于是乎，当含非单词字符的时候，就会有下图所示提示：



您可以狠狠地点击这里：[html5Validate全局对象OBJREG扩展demo](#)

### OBJREG对象与pattern属性之间关系

一句话，`pattern` 权重大于 `OBJREG`。

举个例子，有些情况下，数值输入框可以允许输入中文单位，您就可以这样处理：

```

<input type="number" pattern="\d+" />

```

这样子，插件会使用 `pattern` 属性对应的正则，而不是 `type` 对应的正则。于是，您输入“10人”就不会提示你“请输入数值”。

注意：`pattern` 属性对应的正则中的斜杠(`\`)等无需转义，但是，如果是 `OBJREG` 对象表示，则需要转义(`\\`)！

## 十、各类实际应用

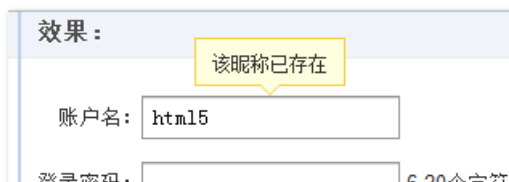
### 1. 注册及其验证

您可以狠狠地点击这里：[html5Validate下注册以及验证demo](#)

这个demo在demo里面算是蛮仿真的，验证码以及验证什么的都是有处理的。

值得说明的

① 昵称有个模拟Ajax验证的处理，当输入5个合法字符的时候，会提示“该昵称已存在”。



其中，使用了前面提到的 `$.testRemind.display` 以及 `$.html5Validate.isRegex(ele)`。

② 确认密码框禁用与不禁用的控制部分有这么一行代码：

```

if ($.html5Validate.isEmpty(this)) {

```

`$.html5Validate.isEmpty` 前面曾一笔带过，其用来检测输入框值是否为空，非 `password` 类型输入框会过滤前后空格。

③ `html5Validate` 插件并没有密码或手机号码前后一致验证的方法，如果在实际使用中遇到这类需求，您可以在 `html5Validate` 验证回调中进行额外验证。

```
// 前后密码一致的验证
if (pwd.val() !== pwdChk.val()) { 回调后的一致性验证
    pwdChk.testRemind("前后密码不一致").get(0).select();
} else {
```

## 2. 发表点评及验证

您可以狠狠地点击这里：[html5Validate发表点评及验证demo](#)



这个demo的重点就在星星的处理上。

星星的本质是单选框组，从差到好共5个单选，默认是显示的，当JS执行到此的时候，单选框组隐藏，默认隐藏的星星们显示；这个交互就是我上面提到的模拟表单！

单选框组的验证只有“是否必选”这一条，如果必选，则仅仅1个单选 `required` 即可，理论上是无论哪个都行。规范与统一期间，我们可以约定同组第一个 `radio` 设置 `required`。再由于这里的radio被模拟，因此，这个有 `required` 属性的单选框必须隐藏( `display:none` 或 `visibility:hidden` )，否则，提示文字的位置为指向这个看不见的单选框。

该单选框HTML代码如下：

```
<input type="radio" id="radioImp1" name="impress" value="1" required data-target="starScore"><label for="radioImp1">差</label>
```

然后通过：

```
$(O).css("visibility", "hidden");
```

将其隐藏不可见，打通任督二脉，以便 `data-target` 的乾坤大挪移招数可以发起来！于是，我们就看到了上图中“提示框指向星星”的画面。

点击星星的操作实际是让对应单选框选中，从这点来看，我们还可以把星星使用 `label` 标签表示，这样，连选中的JS代码都可以省了。呵呵~~

其他都是些常规验证，没有什么好说的。

## 十一、结语以及其他杂碎

IE10向下兼容模式的IE6-IE9，其HTML5表单属性又有严重的冲突bug（外表看不出来）。因此，本插件在这些浏览器下有部分的不支持，主要是IE10支持的HTML5 `type` 类型，具体可参见前面一篇文章：“[IE10 IE7-IE9 type=email的完全抛弃](#)”。



我个人的建议是忽略之，理由略.....如果您实在想支持，`email/tel/url/number` 这几种 `type` 后面加一个空格即可！在参数 `novalidate` 为 `false` 的时候，本插件会自动把后面的空格给 `remove` 掉的，并不影响正常的HTML5表单功能。

本插件验证机制多遵循Chrome浏览器，因此，其表单验证都是一个一个提示的，大范围标红这种需求本插件满足不了。

补充：关于即时验证

我去年这个时候写了篇讨论JS前端验证的文章“[JS前端验证与用户自由思想](#)”。我不喜欢跟风，我对即时响应验证是相当不感冒的，推崇结果为导向的验证。这些想法不是凭空YY出来的，而是借鉴与参考了Google以及Chrome浏览器。因此，像这种可以很好推广自己想法的机会，自己是不会让插件支持即时响应验证的。同时即时验证意味着要对各个表单元素绑定事情，这样，岂不反而陷入了“过多干预”这个缺点，限制了用户的自由。“用户自由”是未来的趋势，**this plugin designs for future!**

虽然插件支持即时验证无望，但是，插件之外是可以简单几行代码支持的，如下：

```
$.fn.immediValidate = function(passCallback) {
    return $(this).find(":input").each(function() {
        if (/^submit|image|reset|file|radio|checkbox|button$/i.test(this.type)) return;
        var self = this;
        $(this).bind("blur", function() {
            setTimeout(function() {
                if ($.testRemind.display == false && $.html5Validate.isAllpass(self))
            ) {
                $.isFunction(passCallback) && passCallback.call(self);
            }
        }, 100);
    });
};
```

于是，我们只要这样绑定JS就可以了：

```
$("#form").html5Validate().immediValidate();
```

本地已验证。

虽然html5Validate插件的完成细嚼慢咽，多番积累，但是，不可能面面俱到，因此，如果承蒙您使用了，并且发现了相关问题，欢迎提出来。我会尽快修改并发布的。因为github自己还玩得不熟，版本啊更新什么的都是比较传统的手动方式，所以可能会有JS是最新的，但是文中的更新日期还是落后的情况，希望见谅。

也欢迎提出其他宝贵意见，不甚感谢！

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



### 猜你喜欢

- HTML5 number类型文本框step属性的验证机制
- 残忍: IE10 IE7-IE9 type=email的完全抛弃
- 表单序列化、规则分离下的复杂表单计算
- 翻译: 关于表单每个设计师都必须知道的10件事
- 伪元素表单控件默认样式重置与自定义大全
- 小tip: CSS动态实现文本框清除按钮的隐藏与显示
- HTML可访问性fieldset,legend元素及CSS布局应用
- HTML5按钮元素新属性formaction,formenctype等简介
- CSS :focus-within伪类选择器及纯CSS下拉等应用举例
- reflection.js-实现图片投影倒影效果js插件
- style标签下的CSS代码的显示与实时编辑

分享到: 1

标签: HTML5, html5Validate, jQuery, 插件, 表单, 验证

### 发表评论 (目前110条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

### « 先前评论

1.

李祥说道:

2018年12月26日 19:12

鑫哥你好, 对你膜拜已久。最近发现这个验证插件的一个问题:

我在表单中输入的是中文括号 (由于某些原因不得使用中文括号), 提交表单是调用了isAllpass方法, 发现输入框的中文括号自动变成了英文括号。请问有什么配置可以避免该情况吗?

[回复](#)



2.

蚂蚁哥说道:

2018年05月9日 05:29

对jquery版本有什么要求么 换了jquery就不能用了。非要用你的那个jq才能运行

[回复](#)



张鑫旭说道:

2018年05月9日 10:31

请问你使用的是哪个版本? 然后使用的是老的那个验证, 还是lulu ui中的验证?

[回复](#)



3. **html5Validate**什么时候更新说道:

2018年03月16日 09:37

html5Validate什么时候更新开源。。。谢谢!

[回复](#)



4. 不指定submit绑定, 说道:

2018年01月7日 02:00

是否可以绑定其他的事件,  
按钮只能

不能是:

[回复](#)



5. 开源说道:

2017年06月13日 15:40

快开源了吗? 好期待! 要不https的网站是不是不能用啊

[回复](#)



6. 威武说道:

2017年01月24日 11:08

很好用, 但是目前数字验证有bug, 就是step那边, js的数字计算很容易出现精度问题, 造成验证不正确。  
96.7692 step=0.0001 理论上应该通过, 但实际不通过。同样的试验了一下chrome没有问题, IE10 有一样的问题

[回复](#)



威武说道:

2017年01月24日 11:09

主要原因就是出现了XXXX.9999999999999999的结果

[回复](#)



7. **meteor**说道:

2017年01月15日 16:18

身份证验证代码:

1、在if(\$(control).isVisible()) {里面加上

else if(type=='cardno'){ //add by meteor at 2017-01-14 15:00

//验证失败后的提示文字, 如果验证通过, 就不会执行到这里

finalText = "请输入正确的身份证号码";

\$(control).testRemind(finalText);

control.focus();

}

2、在if (type == "radio" && isRequired) 前加上下面的代码

if (type=="cardno" && isRequired){// Add By Meteor At 2017-01-14 PM

var UUserCard=\$(el).val();

var birthday = UUserCard.substring(6, 10) + "-" + UUserCard.substring(10, 12) + "-" + UUserCard.substring(12, 14);

//不足位数补0

var paddNum = function(num){

num += "";



```
return num.replace(/^(d)$/, "0$1");
}
var d=new Date(birthday.replace(/-/g, "/"));
if(birthday!=(d.getFullYear()+"-"+paddNum(d.getMonth()+1)+"-"+paddNum(d.getDate()))){
//console.log(birthday);
remind(el, type, tag);
allpass=false;
}
} else if (type == "radio" && isRequired) {
```

验证程度：是否15或18位，身份证日期是否正确，其它更高级别验证自己加上即可。

[回复](#)

8. **meteor**说道：

2017年01月15日 14:22

加个身份证验证就好了

[回复](#)



9. 看完了这些文章很强大。说道：

2016年12月24日 12:10

及时验证很好，但是点击取消其他地方的时提示框应该是要隐藏的吧

[回复](#)



10. **Mi**说道：

2016年12月6日 17:31

ie11 还是大红框 且验证无效

[回复](#)



11. **wj**说道：

2016年11月23日 11:24

验证失败4次，最后一次成功，最后会自动提交5次，，为嘛啊啊，，？？？

[回复](#)



张 鑫旭说道：

2016年11月23日 21:42

检查下事件绑定，可能用法上有些问题

[回复](#)



12. **傅鹏程**说道：

2016年10月13日 14:51

似乎不支持HTML5 的 input的form属性，只能对包裹在form标签中的表单进行校验呢

[回复](#)



13. **ljp404**说道：

2016年09月27日 13:19

关于基于HTML5的表单验证的问题，如果验证通过后，调用\$('#form').submit()这个表单会限于死循环，重复提交，js控制台报错：Maximum call stack size exceeded；如果用原生document.getElementById("form").submit()提交方式，就可以通过

[回复](#)



14.

木子田心七说道：

2016年09月18日 11:24

如果下拉框不使用Select.js，应该如何验证

回复


15.

龙大侠说道：

2016年09月1日 10:55

问个问题,以上校验针对的是页面已经加载完成了的元素.  
如果页面加载后,  
我试了试,通过JS或者JQ动态产生的元素,如一个新的 input 虽然加上了以上校验,但是仍然不会被拿去做校验.  
那么这种动态产生的元素,只能通过自定义校验来搞了吗?

回复


16.

chutianshu说道：

2016年07月29日 11:10

使用这个引用库时，出现“Uncaught ReferenceError: define is not defined”错误，没有引用成功。  
第一种方式倒是没什么问题，期待博主开源，学习学习

回复


17.

heyman说道：

2016年07月25日 22:45

我要验证的表单元素是在一个fixed定位的模态框中，在滚动鼠标的时候，提示消息会跟输入框错位

回复


18.

chinazhao说道：

2016年07月1日 17:18

谢谢，

回复


19.

Terrance说道：

2016年06月25日 15:52

首先赞一个，在使用过程中发现jquery3.0.0不支持诶

回复


20.

allen说道：

2016年06月21日 11:23

插件在移动版safari上不可以自动聚焦，在桌面safari上，自定义的验证无效

回复


21.

greenbowa说道：

2016年05月6日 16:29

用了博主的插件，冒个泡顶一下，谢谢小张

回复


22.

good说道：

2016年03月23日 11:37

新版的开源时间点能透漏下吗



回复

23. xiangli说道:

2016年03月15日 14:11

这个插件不支持手机端吗? 我用微信浏览器跟手机端的ucweb都没有input气泡提示框跟红色边框提示  
demo: <http://www.zhangxinxu.com/study/201511/form-use-mpqq.html>

回复



24. zy说道:

2016年02月26日 17:43

您好, 请问现在对type=number的类型的小数没有校验是吗?

我这样子它只会对填写的是不是数字进行校验, 但是对小数没有校验, 我填写1.123还是能通过, step没有起作用

回复



zy说道:

2016年02月26日 18:01

我知道了, 要加上范围。。。

回复



张 鑫旭说道:

2016年02月26日 18:02

@zy 你好, 应该不至于吧, 我记得处理过的~

回复



25. php570说道:

2016年02月1日 03:31

点击提交时, 不能一次性验证所有表单规则, 希望可以加入“当点击提交时, 一次性验证完并提示”这个功能。

回复



张 鑫旭说道:

2016年02月2日 21:52

@php570 已有新版验证, 等有时间加上。

回复



php570说道:

2016年02月26日 11:11

请问什么时候可以更新? 能通知一下吗?

回复



张 鑫旭说道:

2016年02月26日 11:49

@php570 今年一定会的, 实际上目前已经可以访问了, [http://mp.gting.cn/old\\_mp/assets/js/common/ui/Validate.js](http://mp.gting.cn/old_mp/assets/js/common/ui/Validate.js) demo: <http://www.zhangxinxu.com/study/201511/form-use-mpqq.html>

php570说道:

2016年06月22日 21:07

[http://mp.gting.cn/old\\_mp/assets/js/common/ui/Validate.js](http://mp.gting.cn/old_mp/assets/js/common/ui/Validate.js) 我用 novalidate:true参数无效了? 是取消了吗? 最新版本什么时候发布, 方便告知吗? 谢谢



26.

a\_guest说道:

2016年01月29日 15:47

你好，插件和JQuery的1.11.1版本冲突。。。

[回复](#)



## « 先前评论

### 最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

### 今日热门

- » [常见的CSS图形绘制合集](#) (193)
- » [未来必热: SVG Sprite技术介绍](#) (120)
- » [粉丝群第1期CSS小测点评与答疑](#) (115)
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) (93)
- » [让所有浏览器支持HTML5 video视频标签](#) (86)
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) (82)
- » [CSS3下的147个颜色名称及对应颜色值](#) (80)
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) (77)
- » [写给自己看的display: flex布局教程](#) (76)
- » [小tips: 纯CSS实现打字动画效果](#) (76)

### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) (76)
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) (64)
- » [看, for..in和for..of在那里吵架!](#) (60)
- » [是时候好好安利下LuLu UI框架了!](#) (47)
- » [原来浏览器原生支持JS Base64编码解码](#) (35)
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) (33)
- » [炫酷H5中序列图片视频化播放的高性能实现](#) (31)
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) (30)
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) (26)
- » [写给自己看的display: flex布局教程](#) (26)

## 猜你喜欢

- [HTML5 number类型文本框step属性的验证机制](#)
- [残忍: IE10 IE7-IE9 type=email的完全抛弃](#)
- [表单序列化、规则分离下的复杂表单计算](#)
- [翻译: 关于表单每个设计师都必须知道的10件事](#)
- [伪元素表单控件默认样式重置与自定义大全](#)
- [小tip: CSS动态实现文本框清除按钮的隐藏与显示](#)
- [HTML可访问性fieldset, legend元素及CSS布局应用](#)
- [HTML5按钮元素新属性formaction, formenctype等简介](#)
- [CSS :focus-within伪类选择器及纯CSS下拉等应用举例](#)
- [reflection.js-实现图片投影倒影效果js插件](#)
- [style标签下的CSS代码的显示与实时编辑](#)

