

## Validform 使用入门

### 1、引入 css

请查看下载文件中的 `style.css`，把里面 Validform 必须部分复制到你的 `css` 中（文件里这个注释 `/*=====以下部分是 Validform 必须的=====*/` 之后的部分是必须的）。之前发现有部分网友把整个 `style.css` 都引用在了页面里，然后发现样式冲突了。

### 2、引入 js（jquery 1.4.3 以上版本都可以）

```
1 <script type="text/javascript" src="./js/jquery-  
2 1.6.2.min.js"></script>  
   <script type="text/javascript"  
   src="./Validform/v5.1/Validform_v5.1_min.js"></script>
```

### 3、给需要验证的表单元素绑定附加属性

```
1 <form class="demoform">  
2 <input type="text" value="" name="name" datatype="s5-16" errmsg=" 昵称至少 5 个字符,最多 16 个字符! " />  
3 </form>
```

### 4、初始化，就这么简单

```
1 $(".demoform").Validform();
```

注：

1、Validform 有非压缩、压缩和 NCR 三个版本提供下载，NCR 是通用版，当你页面因编码问题，提示文字出现乱码时可以使用这个版本；

2、Validform 没有限定必须使用 table 结构，它可以适用于任何结构，需要在 tiptype 中定义好位置关系。

## 绑定附加属性

凡要验证格式的元素均需绑定 `datatype` 属性，`datatype` 可选值内置有 10 类，用来指定不同的验证格式。

如果还不能满足您的验证需求，可以传入自定义 `datatype`，自定义 `datatype` 是一个非常强大的功能，通过它可以满足你的任何需求。

可以绑定的附加属性有：**`datatype`**、**`nullmsg`**、**`sucmsg`**、**`errmsg`**、**`ignore`**、**`recheck`**、**`tip`**、**`altercss`**、**`ajaxurl`** 和 **`plugin`**

绑定方法如下所示：

```
1 <!--ajax 实时验证用户名-->  
2 <input type="text" value="" name="name" datatype="e"  
3 ajaxurl="valid.php?myparam1=value1&myparam2=value2" sucmsg="用户名  
4 验证通过! " nullmsg="请输入用户名! " errmsg="请用邮箱或手机号码注册! "  
5 />  
6
```

```

7 <!--密码-->
8 <input type="password" value="" name="userpassword" datatype="*6-
9 15" errmsg="密码范围在 6~15 位之间!" />
10 <!--确认密码-->
11 <input type="password" value="" name="userpassword2" datatype="*"
12 recheck="userpassword" errmsg="您两次输入的账号密码不一致!" />
13
14 <!--默认提示文字-->
15 <textarea tip="请在这里输入您的意见。" errmsg="很感谢您花费宝贵时间给我
16 们提供反馈,请填写有效内容!" datatype="s" altercss="gray"
17 class="gray" name="msg" value="">请在这里输入您的意见。</textarea>
18
19 <!--使用 swfupload 插件-->
20 <input type="text" plugin="swfupload" class="inputxt"
21 disabled="disabled" value="">
  <input type="hidden" value="" pluginhidden="swfupload">

  <!--使用 passwordStrength 插件-->
  <input type="password" errmsg="密码至少 6 个字符,最多 18 个字符!"
  datatype="*6-18" plugin="passwordStrength" class="inputxt"
  name="password" value="">
  <div class="passwordStrength" style="display:none;"><b>密码强度:
  </b> <span>弱</span><span>中</span><span class="last">强
  </span></div>

  <!--使用 DatePicker 插件-->
  <input type="text" plugin="datepicker" class="inputxt"
  name="birthday" value="">

```

说明：

- **datatype**

内置基本的 **datatype** 类型有：\* | \*6-16 | n | n6-16 | s | s6-18 | p | m | e | url

\*：检测是否有输入，可以输入任何字符，不留空即可通过验证；

\*6-16：检测是否为 6 到 16 位任意字符；

n：数字类型；

n6-16：6 到 16 位数字；

s：字符串类型；

s6-18：6 到 18 位字符串；

p：验证是否为邮政编码；

m：手机号码格式；

e：email 格式；

url：验证字符串是否为网址。

自定义 `datatype` 的名称，可以由字母、数字、下划线、中划线和\*号组成。

形如"\*6-16"的 `datatype`，Validform 会自动扩展，可以指定任意的数值范围。如内置基本类型有"\*6-16"，那么你绑定 `datatype="*4-12"`就表示 4 到 12 位任意字符。如果你自定义了一个 `datatype="zh2-4"`，表示 2 到 4 位中文字符，那么 `datatype="zh2-6"`就表示 2 到 6 位中文字符。

5.2 版本之后，`datatype` 支持规则累加或单选。用","分隔表示规则累加；用"|"分隔表示规则多选一，即只要符合其中一个规则就可以通过验证，绑定的规则会依次验证，只要验证通过，后面的规则就会忽略不再比较。如绑定 `datatype="m|e"`，表示既可以填写手机号码，也能填写邮箱地址，如果知道填入的是手机号码，那么就不会再检测他是不是邮箱地址；`datatype="zh,s2-4"`，表示要符合自定义类型"zh"，也要符合规则"s2-4"。

注：

5.2.1 版本之后，`datatype` 支持：

**直接绑定正则：**如可用这样写 `datatype="/w{3,6}/i"`，要求是 3 到 6 位的字母，不区分大小写；

**支持简单的逻辑运算：**如 `datatype="m | e, *4-18 | /w{3,6}/i | /^validform\.rjboy\.cn$/"`，

这个表达式的意思是：可以是手机号码；或者是邮箱地址，但字符长度必须在 4 到 18 位；或者是 3 到 6 位的字母，不区分大小写；或者输入 `validform.rjboy.cn`，区分大小写。这里","分隔相当于逻辑运算里的"&&"； "|"分隔相当于逻辑运算里的"||"；不支持括号运算。

- **nullmsg**

当表单元素值为空时的提示信息，不绑定，默认提示"请填写信息！"。

如：`nullmsg="请填写用户名！"`

5.3 版开始，对于没有绑定 `nullmsg` 的对象，会自动查找 `class` 为 `Validform_label` 下的文字作为提示文字：

如这样的 html 结构：

```
<span class="Validform_label">*用户名：</span><input type="text" val="" datatype="s" />
```

当这个文本框里没有输入时的出错信息就会是："请输入用户名！"

这里 `Validform_label` 跟 `input` 之间的位置关系，不一定要同级关系，同级里没有找到的话，它还会在同级的子级、父级的同级、父级的同级的子级里查找。

- **sucmsg** <sup>5.3+</sup>

当表单元素通过验证时的提示信息，不绑定，默认提示"通过信息验证！"。

如：`sucmsg="用户名还未被使用，可以注册！"`

5.3 版开始，也可以在实时验证返回的 `json` 数据里返回成功的提示文字，请查看附加属性 `ajaxurl` 的介绍。

- **errmsg**

输入内容不能通过验证时的提示信息，默认提示"请输入正确信息！"。

如：`errmsg="用户名必须是 2 到 4 位中文字符！"`

5.3 版开始，Validform 可以根据你绑定的 `datatype` 智能的输出相应出错信息，具体介绍请查看 [tipmsg](#)

- **ignore**

绑定了 `ignore="ignore"`的表单元素，在有输入时，会验证所填数据是否符合 `datatype` 所指定数据类型，

没有填写内容时则会忽略对它的验证；

- **recheck**

表单里面经常需要检查两次密码输入是否一致，**recheck** 就是用来指定需要比较的另外一个表单元素。

如：**recheck="password1"**，那么它就会拿当前元素的值跟该表单下，**name** 为"password1"的元素比较。

- **tip**

表单里经常有些文本框需要默认就显示一个灰色的提示文字，当获得焦点时提示文字消失，失去焦点时提示文字显示。**tip** 属性就是用来实现这个效果。它通常和 **altercss** 搭配使用。

如<input type="text" value="默认提示文字" class="gray intxt" tip="默认提示文字"  
**altercss="gray" />**

- **altercss**

它需要和 **tip** 属性配合使用，**altercss** 指定的样式名，会在文本框获得焦点时被删除，没有输入内容而失去焦点时重新加上。

- **ajaxurl**

指定 **ajax** 实时验证的后台文件的地址。

后台页面如 **valid.php** 文件中可以用 **\$\_POST["param"]** 接收到值，Ajax 中会 POST 过来变量 **param** 和 **name**。**param** 是文本框的值，**name** 是文本框的 **name** 属性。

5.2 版本开始，可以在 **ajaxurl** 指定的 url 后绑定参数，如：

**ajaxurl="valid.php?myparam1=value1&myparam2=value2"**；5.3.1 开始，地址后面附带的参数内部不再做另外解析，仍附带在地址后面，所以需要用 **GET** 方式去获取地址后面带的参数。

5.3 之前的版本中，该文件输出的字符会作为错误信息显示在页面上，如果验证通过需输出小写字母"y"。

在 5.3 版中，实时验证的返回数据做了调整，须是含有 **status** 值的 json 数据！跟 **callback** 里的 **ajax** 返回数据格式统一，建议不再返回字符串"y"或"n"。目前这两种格式的数据都兼容。

注：

如果 **ajax** 校验通过，会在该元素上绑定 **validform\_valid** 值为 **true**。可以通过设置该值来控制验证能不能通过，如验证码的验证，第一次验证通过后，不小心右点击了下验证码图片，验证码换了，但是仍然指示这个对象已经通过了验证，这时可以手动调整该值：

**\$("#name")[0].validform\_valid="false"**。

怎样设置 **ajax** 的参数，具体可以查看 **Validform** 对象的 **config** 方法。

- **plugin**

指定需要使用的插件。

5.3 版开始，对于日期、**swfupload** 和密码强度检测这三个插件，绑定了 **plugin** 属性即可以初始化对应的插件，可以不用在 **validform** 初始化时传入空的 **usePlugin** 了。

如，你要使用日期插件，5.3 之前版本需要这样初始化：

- ```
$(".demoform").Validform({
```
- ```
    usePlugin:{
```
- ```
        datepicker:{}
```
- ```
    }
```
- ```
});
```
- 5.3 版开始，不需要传入这些空对象了，只需在表单元上绑定 **plugin="datepicker"** 就可以，初始化直接这样：  

```
$(".demoform").Validform();
```

## 初始化参数说明

所有可用参数如下：

```
1  $(".demoform").Validform({
2      btnSubmit:"#btn_sub",
3      btnReset:".btn_reset",
4      tiptype:1,
5      ignoreHidden:false,
6      dragonfly:false,
7      tipSweep:true,
8      label:".label",
9      showAllError:false,
10     postonce:true,
11     ajaxPost:true,
12     datatype:{
13         "*6-20": /^[^\s]{6,20}$/,
14         "z2-4" : /^[^\u4E00-\u9FA5\uF900-\uFA2D]{2,4}$/,
15         "username":function(gets,obj,curform,regxp){
16             //参数 gets 是获取到的表单元素值, obj 为当前表单
17             元素, curform 为当前验证的表单, regxp 为内置的一些正则表达式的引用;
18             var reg1=/^[^\w\.]{4,16}$/,
19                 reg2=/^[^\u4E00-\u9FA5\uF900-
20                 \uFA2D]{2,8}$/;
21
22             if(reg1.test(gets)){return true;}
23             if(reg2.test(gets)){return true;}
24             return false;
25
26             //注意 return 可以返回 true 或 false 或 字符串
27             文字, true 表示验证通过, 返回字符串表示验证失败, 字符串作为错误提示显示, 返回
28             false 则用 errmsg 或默认的错误提示;
29         },
30         "phone":function(){
31             // 5.0 版本之后, 要实现二选一的验证效果,
32             datatype 的名称 不 需要以 "option_" 开头;
33         }
34     },
35     usePlugin:{
36         swfupload:{},
37         datepicker:{},
38         passwordstrength:{},
39         jqtransform:{
40             selector:"select,input"
```

```

41     },
42     beforeCheck: function (curform) {
43         //在表单提交执行验证之前执行的函数, curform 参数是当前表
44 单对象。
45         //这里明确 return false 的话将不会继续执行验证操作;
46
47     },
48     beforeSubmit: function (curform) {
49         //在验证成功后, 表单提交前执行的函数, curform 参数是当前
50 表单对象。
51         //这里明确 return false 的话表单将不会提交;
52     },
53     callback: function (data) {
54         //返回数据 data 是 json 对象, {"info": "demo
55 info", "status": "y"}
56
57         //info: 输出提示信息;
58         //status: 返回提交数据的状态, 是否提交成功。如可以用"y"表
59 示提交成功, "n"表示提交失败, 在 ajax_post.php 文件返回数据里自定字符, 主要用
60 在 callback 函数里根据该值执行相应的回调操作;
61         //你也可以在 ajax_post.php 文件返回更多信息在这里获取, 进
62 行相应操作;
63
64         //ajax 遇到服务端错误时也会执行回调, 这时的 data 是
65 { status: **, statusText: **, readyState: **, responseText: ** };
66
67         //这里执行回调操作;
68         //注意: 如果不是 ajax 方式提交表单, 传入 callback, 这时
69 data 参数是当前表单对象, 回调函数会在表单验证全部通过后执行, 然后判断是否提交表
70 单, 如果 callback 里明确 return false, 则表单不会提交, 如果 return true 或
71 没有 return, 则会提交表单。
72     }
73 });

```

### 参数说明：【所有参数均为可选项】

- 必须是表单对象执行 Validform 方法, 示例中".demoform"就是绑定在 form 元素上的 class 名称;
- **btnSubmit**  
指定当前表单下的哪一个按钮触发表单提交事件, 如果表单下有 submit 按钮时可以省略该参数。示例中".btn\_sub"是该表单下要绑定点击提交表单事件的按钮;
- **btnReset**  
".btn\_reset"是该表单下要绑定点击重置表单事件的按钮;
- **tiptype**  
可用的值有: 1、2、3、4 和 function 函数, 默认 tiptype 为 1。3、4 是 5.2.1 版本新增  
1=> 自定义弹出框提示;

2=> 侧边提示(会在当前元素的父级的 `next` 对象的子级查找显示提示信息对象, 表单以 `ajax` 提交时会弹出自定义提示框显示表单提交状态);

3=> 侧边提示(会在当前元素的 `siblings` 对象中查找显示提示信息对象, 表单以 `ajax` 提交时会弹出自定义提示框显示表单提交状态);

4=> 侧边提示(会在当前元素的父级的 `next` 对象下查找显示提示信息对象, 表单以 `ajax` 提交时不显示表单的提交状态);

如果上面的 4 种提示方式不是你需要的, 你可以给 `tiptype` 传入自定义函数。通过自定义函数, 可以实现你想要的任何提示效果:

- `tiptype: function(msg, o, cssctl) {`
- `//msg: 提示信息;`
- `//o: {obj: *, type: *, curform: *},`
- `//obj 指向的是当前验证的表单元素 (或表单对象, 验证全部验证通过, 提交表单时 o.obj 为该表单对象),`
- `//type 指示提示的状态, 值为 1、2、3、4, 1: 正在检测/提交数据, 2: 通过验证, 3: 验证失败, 4: 提示 ignore 状态,`
- `//curform 为当前 form 对象;`
- `//cssctl: 内置的提示信息样式控制函数, 该函数需传入两个参数: 显示提示信息对象 和 当前提示的状态 (既形参 o 中的 type);`
- `}`

具体参见 [demo](#) 页。

`tiptype` 不为 1 时, `Validform` 会查找 `class` 为 `"Validform_checktip"` 的标签显示提示信息。

`tiptype=1` 时, 会自动创建弹出框显示提示信息。

`Validform_checktip` 和表单元素之间的位置关系, 会根据 `tiptype` 的值有对应的结构, 上面已经做了说明。

5.3 版本开始, 如果页面里没有显示出错信息的标签, 会根据 `tiptype` 值自动创建 `Validform_checktip` 对象。

- **ignoreHidden**  
可用值: `true | false`。  
默认为 `false`, 当为 `true` 时对 `:hidden` 的表单元素将不做验证;
- **dragonfly**  
可用值: `true | false`。  
默认 `false`, 当为 `true` 时, 值为空时不做验证;
- **tipSweep**  
可用值: `true | false`。  
默认为 `false`, 5.3 版中做了修正, 在各种 `tiptype` 下, 为 `true` 时提示信息将只会在表单提交时触发显示, 各表单元素 `blur` 时不会触发信息提示;
- **label** <sup>5.3.1+</sup>  
选择符  
在没有绑定 `nullmsg` 时查找要显示的提示文字, 默认查找 `".Validform_label"` 下的文字;
- **showAllError**  
可用值: `true | false`。

默认为 **false**, **true**: 提交表单时所有错误提示信息都会显示; **false**: 一碰到验证不通过的对象就会停止检测后面的元素, 只显示该元素的错误信息;

- **postonce**

可用值: **true** | **false**。

默认为 **false**, 指定是否开启二次提交防御, **true** 开启, 不指定则默认关闭;  
为 **true** 时, 在数据成功提交后, 表单将不能再继续提交。

- **ajaxPost**

可用值: **true** | **false**。

默认为 **false**, 使用 **ajax** 方式提交表单数据, 将会把数据 **POST** 到 **config** 方法或表单 **action** 属性里设定的地址;

- **datatype**

传入自定义 **datatype** 类型, 可以是正则, 也可以是函数。

- `datatype: {`
- `"zh2-4": /^[u4E00-\u9FA5\uF900-\uFA2D]{2,4}$/,`
- `"phone": function(gets, obj, curform, regexp) {`
- `//参数 gets 是获取到的表单元素值,`
- `//obj 为当前表单元素,`
- `//curform 为当前验证的表单,`
- `//regexp 为内置的一些正则表达式的引用。`
- `//return false 表示验证出错, 没有 return 或者 return true`
- `//表示验证通过。`
- `}`
- `}`

具体示例请参考 [demo](#) 页;

- **usePlugin**

目前已整合 **swfupload**、**datepicker**、**passwordstrength** 和 **jqtransform** 四个插件, 在这里传入这些插件使用时需要传入的参数。**datepicker** 在 **Validform** 内调用时另外扩展了几个比较实用的参数, 具体请参考 [demo](#) 页;

- **beforeCheck(curform)**

在表单提交执行验证之前执行的函数, **curform** 参数获取到的是当前表单对象。

函数 **return false** 的话将不会继续执行验证操作;

- **beforeSubmit(curform)**

在表单验证通过, 提交表单数据之前执行的函数, **data** 参数是当前表单对象。

函数 **return false** 的话表单将不会提交;

- **callback**

在使用 **ajax** 提交表单数据时, 数据提交后的回调函数。返回数据 **data** 是 **Json** 对象:

```
{"info":"demo info","status":"y"}
```

**info**: 输出提示信息,

**status**: 返回提交数据的状态, 是否提交成功, **"y"** 表示提交成功, **"n"** 表示提交失败, 在

**ajax\_post.php** 文件返回数据里自定字符, 主要用在 **callback** 函数里根据该值执行相应的回调操



作。你也可以在 `ajax_post.php` 文件返回更多信息在这里获取，进行相应操作；  
如果不是 `ajax` 方式提交表单，传入 `callback`，这时 `data` 参数是当前表单对象，回调函数会在表单验证全部通过后执行，然后判断是否提交表单，如果 `callback` 里 `return false`，则表单不会提交，如果 `return true` 或没有 `return`，则会提交表单。

5.3 版开始，`ajax` 遇到服务端错误时也会执行回调，这时的 `data` 是 `{ status:**, statusText:**, readyState:**, responseText:** }`

## Validform 对象[方法支持链式调用]

如示例 `var demo=$("#formsub").Validform()`，那么 `demo` 对象会有以下属性和方法可以调用：

### tipmsg 【object】

如：`demo.tipmsg.s="error! no message inputed."`；

通过该对象可以修改除 `tit` 以外的其他提示文字，这样可以实现同一个页面的不同表单使用不同的提示文字。

具体可修改的提示文字

```
$.Tipmsg={ //默认提示文字；
tit:"提示信息",
w:{
    "*":"不能为空！ ",
    "*6-16":"请填写 6 到 16 位任意字符！ ",
    "n":"请填写数字！ ",
    "n6-16":"请填写 6 到 16 位数字！ ",
    "s":"不能输入特殊字符！ ",
    "s6-18":"请填写 6 到 18 位字符！ ",
    "p":"请填写邮政编码！ ",
    "m":"请填写手机号码！ ",
    "e":"邮箱地址格式不对！ ",
    "url":"请填写网址！ "
},
def:"请填写正确信息！ ",
undef:"datatype 未定义！ ",
reck:"两次输入的内容不一致！ ",
r:"通过信息验证！ ",
c:"正在检测信息...",
s:"请 {填写|选择} {0|信息}！ ",
v:"所填信息没有经过验证，请稍后...",
p:"正在提交数据..."
};
```

要修改 `tit`（弹出框的标题文字）的话，可以这样：`$.Tipmsg.tit="Message Box"`，则弹出框的标题文字会换成 `"Message Box"`

注：5.3.2+

`$.Tipmsg.w` 里，形如 `"*6-16"` 的提示文字，里面的数字是会被替换的。如绑定 `datatype="*2-18"`，那它默认的出错信息就会是 `"请填写 2 到 18 位任意字符！"`，可以通过 `$.Tipmsg.w` 或 `Validform` 对象的 `tipmsg` 属性修改和扩展默认错误信息，如果你已经设置了 `"zh2-4"` 的提示信息

是"2-4 位中文",那么"zh2-8"出错的信息就自动会是: "2-8 位中文"。对于自定义的 `datatype`, 在扩展默认信息时, 注意错误信息的名字要跟 `datatype` 名字一样, 如上面示例是:

`$.Tipmsg.w["zh2-4"]="2-4 位中文"`。对于多页面或一个页面多表单有相同 `datatype` 来说, 在 `$.Tipmsg.w` 或 `Validform` 对象的 `tipmsg` 属性中扩展默认提示信息是个很好的选择。

#### 5.3.1+

`$.Tipmsg.s` 是用来指定在没有绑定 `nullmsg` 时的默认提示信息。"{0}"是会被找到的 `label` 参数指定的对象或 `Validform_label` 里的文字替换掉的, "{填写|选择}"里的文字在绑定了"recheck"属性的表单元素上检测时是会不显示的, 当前验证对象是 `radio`、`checkbox` 或 `select` 时, 会输出"选择", 是其他类型的元素时会输出"填写"和后面的"信息"。

具体示例请参见 [demo](#) 页。

## dataType 【object】

获取内置的一些正则:

```
{
  "match": /^ (. +?) (\d+) - (\d+) $/,
  "*": /\w\W+/,
  "*6-16": /\w\W {6, 16} $/,
  "n": /\d+$/,
  "n6-16": /\d {6, 16} $/,
  "s": /^ [\u4E00-\u9FA5\uF900-\uFA2D\w\.\s]+$/,
  "s6-18": /^ [\u4E00-\u9FA5\uF900-\uFA2D\w\.\s] {6, 18} $/,
  "p": /^[0-9] {6} $/,
  "m": /^[13[0-9] {9} $|14[0-9] {9} $|15[0-9] {9} $|18[0-9] {9} $|,
  "e": /^ \w+ ([-.' ] \w+)* @ \w+ ([-.] \w+)* \. \w+ ([-.] \w+)* $/,
  "url": /^ (\w+: \w\/)? \w+ (\. \w+)+ . * $/
}
```

## addRule(rule) 【返回值 : Validform】

可以通过 `Validform` 对象的这个方法给表单元素绑定验证规则, [绑定验证类型](#) 中列出的附加属性都可以通过这个方法绑定。

```
demo.addRule([
  {
    ele: "#name",
    datatype: "s6-18",
    ajaxurl: "valid.php",
    nullmsg: "请输入昵称!",
    errormsg: "昵称至少 6 个字符, 最多 18 个字符!"
  },
  {
    ele: "#userpassword",
    datatype: "*6-16",
    nullmsg: "请设置密码!",
    errormsg: "密码范围在 6~16 位之间!"
  }
])
```

```

    },
    {
        ele: "#userpassword2",
        datatype: "*",
        recheck: "userpassword",
        nullmsg: "请再输入一次密码！",
        errormsg: "您两次输入的账号密码不一致！"
    }
]);

```

其中 **ele** 是指定要绑定规则的对象，会在 **Validform** 对象下查找这些对象。

### eq(n) 【返回值：Validform】

获取 **Validform** 对象的第 **n** 个元素。

如你页面上有多个 **form** 的 **class** 都是 **formsub**，执行上面的验证绑定，得到的 **demo** 对象就可以操作所有这些表单，如果你要对其中某个表单执行某些操作，那么就可以使用这个方法。

如 **demo.eq(0).resetForm()**，重置第一个表单。

### ajaxPost(flag, sync, url) 【返回值：Validform】

以 **ajax** 方式提交表单。**flag** 为 **true** 时，跳过验证直接提交，**sync** 为 **true** 时将以同步的方式进行 **ajax** 提交。

参数 **url** 是 5.3 版新增，传入了 **url** 地址时，表单会提交到这个地址

如 **demo.ajaxPost(true)**，不做验证直接 **ajax** 提交表单。

### abort() 【返回值：Validform】

终止 **ajax** 的提交。

如执行上面的 **ajaxPost()** 之后，发现某些项填写不对，想取消表单提交，那么就可以执行这个操作：**demo.abort()**

### submitForm(flag, url) 【返回值：Validform】

以初始化时传入参数的设置方式提交表单，**flag** 为 **true** 时，跳过验证直接提交。

参数 **url** 是 5.3 版新增，传入了 **url** 地址时，表单会提交到这个地址

如 **demo.submitForm(true)**，不做验证直接提交表单。

### resetForm() 【返回值：Validform】

重置表单。

如 **demo.resetForm()**，重置表单到初始状态。

### resetStatus() 【返回值：Validform】

重置表单的提交状态。传入了 **postonce** 参数的话，表单成功提交后状态会设置为 **"posted"**，重置提交状态可以让表单继续可以提交。

如 **demo.resetStatus()**

## getStatus() 【返回值：String】

获取表单的提交状态，normal：未提交，posting：正在提交，posted：已成功提交过。

如 demo.getStatus()

## setStatus(status) 【返回值：Validform】

设置表单的提交状态，可以设置 normal，posting，posted 三种状态，不传参则设置状态为 posting，这个状态表单可以验证，但不能提交。

如 demo.setStatus("posted")

## ignore(selector) 【返回值：Validform】

忽略对所选择对象的验证，不传入 selector 则忽略所有表单元素。

如 demo.ignore("select,textarea,#name")，忽略 Validform 对象下所有 select，textarea 及一个 id 为"name"元素的验证。

## unignore(selector) 【返回值：Validform】

将 ignore 方法所忽略验证的对象重新获取验证效果，不传入 selector 则恢复验证所有表单元素。

如 demo.unignore("select,textarea,#name")，恢复 Validform 对象下所有 select，textarea 及一个 id 为"name"元素的验证。

## check(bool,selector) 【返回值：Boolean】

bool 为 true 时则只验证不显示提示信息

对指定对象进行验证(默认验证当前整个表单)，通过返回 true，否则返回 false（绑定实时验证的对象，格式符合要求时返回 true，而不会等 ajax 的返回结果）

如 demo.check()，验证当前整个表单，且只验证但不显示对错信息。

## config(setup) <sup>5.3+</sup> 【返回值：Validform】

setup 参数是一个对象。

如：

```
demo.config({
    showAllError:true,
    url:"这里指定提交地址",
    ajaxpost:{
        //可以传入$.ajax()能使用的，除 dataType 外的所有参数;
    },
    ajaxurl:{
        //可以传入$.ajax()能使用的，除 dataType 外的所有参数;
    }
})
```

### 可用参数:

tiptype 等 <sup>5.3.2+</sup>: 可以在这里设置初始化可用的所有参数

url: 指定表单的提交路径, 这里指定的路径会覆盖表单 action 属性所指定的路径

ajaxpost: 表单以 ajax 提交时, 可以在这里配置 ajax 的参数

ajaxurl: 配置实时验证 ajax 的参数

①、config 方法还能已经初始化过的表单重新配置参数了! <sup>5.3.2+</sup>

如 `var demo=$( ".formsub").Validform();`

`demo.config({tiptype:2});`

demo 下的各表单还能分别配置不同参数!

如 `demo.eq(0).config({tiptype:1,ajaxPost:true});`

`demo.eq(1).config({tiptype:1,callback:function(curform){my_method_to_submit_form(); return false;}});`

②、执行 config 可以动态设置、添加参数, 如:

`demo.config({`

`url:"http://validform.rjboy.cn"`

`});`

`$( ".save").click(function() {`

`demo.config({`

`ajaxpost:{`

`timeout:1000`

`}`

`});`

`});`

那么在点击 save 按钮后, demo 所对应的表单的 config={

`url:"http://validform.rjboy.cn",`

`ajaxpost:{`

`timeout:1000`

`}`

`}`

③、参数 url 的优先级: form 表单的 action 所指定的提交地址会被 config.url 覆盖,

config.url 会被 config.ajaxpost.url 覆盖, config.ajaxpost.url 会被 Validform 对象的方法 submitForm(flag,url)

和 ajaxPost(flag, sync, url) 里的 url 覆盖。

如果表单里没有指定 action 提交地址, 那么就会提交到 config.url 设定的地址。

考虑到整个验证框架的逻辑, 传入 dataType 参数不会起作用, 不会被覆盖, ajax 必须返回含有 status 值的 json 数据。

④、另外注意的是: 传入的 success 和 error 方法里, 能多获取到一个参数, 如:

```

demo.config={
  ajaxpost:{
    url:"",
    timeout:1000,
    ...,
    success:function(data,obj){
      //data 是返回的 json 数据;
      //obj 是当前表单的 jquery 对象;
    },
    error:function(data,obj){
      //data 是{ status:**, statusText:**, readyState:**,
      responseText:** };
      //obj 是当前表单的 jquery 对象;
    }
  },
  ajaxurl:{
    success:function(data,obj){
      //data 是返回的 json 数据;
      //obj 是当前正做实时验证表单元素的 jquery 对象;
      //注意: 5.3 版中, 实时验证的返回数据须是含有 status
      值的 json 数据!
      //跟 callback 里的 ajax 返回数据格式统一, 建议不再返回
      字符串"y"或"n"。目前这两种格式的数据都兼容。
    }
  }
}

```

## 调用外部插件

### 文件上传 - swfupload

[选择文件后立即上传](#)

[表单验证通过后上传文件](#)

### 密码强度检测 - passwordStrength

[密码强度提示与验证提示同时显示](#)

[在符合验证要求时才有密码强度提示](#)

### 日期控件 - datePicker

[默认效果](#)

[选择日期后执行回调函数](#)

### 表单美化 - jqtransform

[默认对所有元素美化](#)  
[美化指定的表单元素](#)

## Validform 的公用对象

### \$.Datatype

可以通过\$.Datatype 对象来扩展 datatype 类型。  
如\$.Datatype.zh=/^[u4E00-u9FA5\uF900-\uFA2D]{1,}\$/

### \$.Tipmsg

可以通过\$.Tipmsg 对象来修改默认提示文字。具体可修改的提示文字请查看 Validform 对象的 tipmsg 属性。  
如果 Validform 对象的 tipmsg 属性没有找到相关的提示信息，那么就会到\$.Tipmsg 中查找对应提示文字。  
如\$.Tipmsg.tit="msg box"; //设置默认弹出框的标题文字。

### \$.Showmsg(msg)

调用 Validform 自定义的弹出框。  
参数 msg 是要显示的提示文字。  
如\$.Showmsg("这是提示文字"); //如果不传入信息则不会有弹出框出现，像\$.Showmsg()这样是不会弹出提示框的。

### \$.Hidmsg()

关闭 Validform 自定义的弹出框。  
如\$.Hidmsg()