

Validform v5.3.2 文档

- Validform使用入门
- 绑定附加属性
- 初始化参数说明
- Validform对象
 - tipmsg
 - dataType
 - addRule(rule)
 - eq(n)
 - ajaxPost(flag,sync,url)
 - abort()
 - submitForm(flag,url)
 - resetForm()
 - resetStatus()
 - getStatus()
 - setStatus(status)
 - ignore(selector)
 - unignore(selector)
 - check(bool,selector)
 - config(setup)
- 调用外部插件
- Validform的公用对象

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-1.6.2.min.js"></script>
2 <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" errormsg="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**、**errormsg**、**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" ajaxurl="valid.php?myparam1=value1&myparam2=\
3
4 <!--密码-->
5 <input type="password" value="" name="userpassword" datatype="*6-15" errormsg="密码范围在6~15位之间!
6 <!--确认密码-->
7 <input type="password" value="" name="userpassword2" datatype="*" recheck="userpassword" errormsg=
8
9 <!--默认提示文字-->
10 <textarea tip="请在这里输入您的意见。" errormsg="很感谢您花费宝贵时间给我们提供反馈，请填写有效内容!" dataty
11
12 <!--使用swfupload插件-->
13 <input type="text" plugin="swfupload" class="inputtxt" disabled="disabled" value="">
14 <input type="hidden" value="" pluginhidden="swfupload">
15
16 <!--使用passwordStrength插件-->
17 <input type="password" errormsg="密码至少6个字符,最多18个字符!" datatype="*6-18" plugin="passwordStre
18 <div class="passwordStrength" style="display:none;"><b>密码强度: </b> <span>弱</span><span>中</span>
19
20 <!--使用DatePicker插件-->
21 <input type="text" plugin="datepicker" class="inputtxt" 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"*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 5.3+

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。

errormsg

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: errormsg="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, 只需在表单元素上绑定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        "2-4" : /^[u4E00-\u9FA5\uF900-\uFA2D]{2,4}$/,
15        "username":function(gets,obj,curform,regxp){
16            //参数gets是获取到的表单元素值, obj为当前表单元素, curform为当前验证的表单, regxp为P
17            var reg1=/^[w\.\-]{4,16}$/,
18                reg2=/^[u4E00-\u9FA5\uF900-\uFA2D]{2,8}$/;
19
20            if(reg1.test(gets)){return true;}
21            if(reg2.test(gets)){return true;}
22            return false;
23        },
24        //注意return可以返回true 或 false 或 字符串文字, true表示验证通过, 返回字符串表示验
25    },
26    "phone":function(){
27        // 5.0 版本之后, 要实现二选一的验证效果, datatype 的名称 不 需要以 "option_" 开头;
28    },
29    usePlugin:{
30        swfupload:{},
31        datepicker:{},
32        passwordstrength:{},
33        jqtransform:{
34            selector:"select,input"
35        }
36    },
37    beforeCheck:function(curform){
38        //在表单提交执行验证之前执行的函数, curform参数是当前表单对象。
39        //这里明确return false的话将不会继续执行验证操作;
40    },
41    beforeSubmit:function(curform){
42        //在验证成功后, 表单提交前执行的函数, curform参数是当前表单对象。
43        //这里明确return false的话表单将不会提交;
44    },
45    callback:function(data){
46        //返回数据data是json对象, {"info":"demo info","status":"y"}
47        //info: 输出提示信息;
48        //status: 返回提交数据的状态, 是否提交成功。如可以用"y"表示提交成功, "n"表示提交失败, 在ajax_p
49        //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;
50        //ajax遇到服务端错误时也会执行回调, 这时的data是{ status:**, statusText:**, readyState:*
51
52        //这里执行回调操作;
53        //注意: 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
54    }
55 });
```

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: 显示提示信息的对象 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时不会触发信息显示;

label 5.3.1+

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

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([\n  {\n    ele: "#name",\n    datatype: "s6-18",\n    ajaxurl: "valid.php",\n    nullmsg: "请输入昵称! ",\n    errmsg: "昵称至少6个字符,最多18个字符! "\n  },\n  {\n    ele: "#userpassword",\n    datatype: "*6-16",\n    nullmsg: "请设置密码! ",\n    errmsg: "密码范围在6~16位之间! "\n  },\n  {\n    ele: "#userpassword2",\n    datatype: "*",\n    recheck: "userpassword",\n    nullmsg: "请再输入一次密码! ",\n    errmsg: "您两次输入的账号密码不一致! "\n  }\n]);
```

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) 5.3+ 【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 5.3.2+: 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! 5.3.2+

```
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-u9FA5uf900-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。

\$.Hidemsg()

关闭Validform自定义的弹出框。
如\$.Hidemsg()