

jQuery的表单验证

jQuery的是我们的一个js框架，围绕这一个js框架，它有很多第三方的插件，如我们之前使用过的 **layer**。所有依赖于jQuery的插件都必须导入在jQuery的后面。今天我们就来介绍一个工作当中使用得非常频繁的一个插件叫**jQuery.validator.js**

这个插件主要是做表单验证的

注册新学员

* 学生姓名	<input type="text" value="请输入学生姓名"/>	* 性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
	学生姓名不能为空		
* 密码	<input type="password" value="请输入密码"/>	* 确认密码	<input type="password" value="请再次输入密码"/>
	密码不能为空		确认密码不能为空
* 学校	<input type="text" value="请选择学校"/>	* 专业	<input type="text" value="请输入专业"/>
	学校不能为空		专业不能为空
* 手机号	<input type="text" value="请输入手机号"/>	QQ号	<input type="text" value="请输入QQ号"/>
	手机号不能为空		
* 学历	<input type="text" value="请输入学历"/>	英语等级	<input type="text" value="请输入英语等级"/>
	学历不能为空		
* 毕业年限	<input type="text" value="请输入毕业年限"/>	家庭电话	<input type="text" value="请输入家庭电话"/>
	毕业年限不能为空		
* 所属班级	<input type="text" value="所属班级"/>	* 学生状态	<input checked="" type="radio"/> 正常 <input type="radio"/> 休学
	所属班级不能为空		

上图就是一个表单，表单验证就是当提交数据或进行某个操作的时候，把表单里面的元素按照我们所需要的要求进行验证，在jQuery里面有一个现成的插件叫**jquery.validator.js**来完成

安装

这个插件直接在百度里面下载就可以了

```
1 <script src="./js/jquery-3.6.1.js"></script>
2 <script src="./js/jquery.validate.js"></script>
3 <script src="./js/messages_zh.js"></script>
```

配置规则

这个插件主要就是用来对表单里面的元素进行规则校验

```
1 <!DOCTYPE html>
2 <html lang="zh">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>表单验证</title>
9 </head>
10
11 <body>
12     <form id="loginForm">
13         <div>
14             用户名: <input type="text" placeholder="请输入用户名"
15             name="userName">
```

```

15         </div>
16     </div>
17     密码: <input type="text" placeholder="请输入密码" name="pwd">
18 </div>
19 </div>
20     <button type="button" id="btn1">登录</button>
21 </div>
22 </form>
23 </body>
24 <script src="./js/jquery-3.6.1.js"></script>
25 <script src="./js/jquery.validate.js"></script>
26 <script src="./js/messages_zh.js"></script>
27 <script>
28     $(function () {
29         //第一步: 要知道你要校验哪一个表单 ,我们给了表单一个id
30         //第二步: 你要校验哪些表单元素, 就在这个元素上面添加一个
name属性
31         //第三步: 在validate的下面写一个rules验证规则
32         //第四步: 当所有的规则配置好了以后, validate方法就会返回当前表单的一对象,有了这个表单的对象,我们就可以开始验证了
33         var loginForm = $("#loginForm").validate({
34             // 规则
35             rules: {
36                 userName: {
37                     required: true
38                 },
39                 pwd: {
40                     required: true
41                 }
42             },
43             //自定义验证不通过的消息显示
44             messages: {
45                 userName: {
46                     required: "兄弟, 用户名不能为空"
47                 },
48                 pwd: {
49                     required: "小伙子, 密码呢"
50                 }
51             }
52         });
53
54         $("#btn1").click(function () {
55             // 调用form()方法开始验证,得到一个boolean的结果,true代表验证通过,false代表验证不通过
56             var result = loginForm.form();
57             if (result) {
58                 console.log("恭喜你表单验证通过了");
59             }
60             else {

```

```
61         console.log("表单验证不通过");
62     }
63
64     })
65 })
66 </script>
67
68 </html>
```

上面的代码是一个最简单的表单验证的过程步骤代码，我们也已经看到了结果

思考：上面的代码当中我们已经知道了jQuery.validator的使用方法和步骤，我们也知道required就代表这个字段必填，但是除了这个验证规则以后，还有没有其它的验证规则呢？

验证规则

jQuery.validator这个插件提供了很多内置的验证规则，如下表所示

默认校验规则

序号	规则	描述
1	required:true	必须输入的字段。
2	remote:"check.php"	使用 ajax 方法调用 check.php 验证输入值。
3	email:true	必须输入正确格式的电子邮件。
4	url:true	必须输入正确格式的网址。
5	date:true	必须输入正确格式的日期。日期校验 ie6 出错，慎用。
6	dateISO:true	必须输入正确格式的日期 (ISO)，例如：2009-06-23，1998/01/22。只验证格式，不验证有效性。
7	number:true	必须输入合法的数字（负数，小数）。
8	digits:true	必须输入整数。
9	creditcard:	必须输入合法的信用卡号。
10	equalTo:"#field"	输入值必须和 #field 相同。
11	accept:	输入拥有合法后缀名的字符串（上传文件的后缀）。
12	maxlength:5	输入长度最多是 5 的字符串（汉字算一个字符）。
13	minlength:10	输入长度最小是 10 的字符串（汉字算一个字符）。
14	rangelength:[5,10]	输入长度必须介于 5 和 10 之间的字符串（汉字算一个字符）。
15	range:[5,10]	输入值必须介于 5 和 10 之间。
16	max:5	输入值不能大于 5。
17	min:10	输入值不能小于 10。

验证方式

在jQuery.validator的验证方式里面有两种验证方式

1. 配置式的验证
2. 侵入式的验证

配置式的验证

使用配置来完成相应的验证规则

```
1  <!DOCTYPE html>
2  <html lang="zh">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-
      scale=1.0">
8      <title>表单验证</title>
9      <style>
10         .form-item {
11             margin: 10px 0;
12         }
13     </style>
14 </head>
15
16 <body>
17     <form id="registerForm">
18         <div class="form-item">
19             用户名: <input type="text" name="userName">
20         </div>
21         <div class="form-item">
22             密码: <input type="text" name="pwd" id="pwd">
23         </div>
24         <div class="form-item">
25             确认密码: <input type="text" name="pwd2">
26         </div>
27         <div class="form-item">
28             年龄: <input type="text" name="age">
29         </div>
30         <div class="form-item">
31             身份证号: <input type="text" name="IDCard">
32         </div>
33         <div class="form-item">
34             爱好: <input type="text" name="hobby">
35         </div>
36         <div class="form-item">
37             <button type="button" id="btn-register">注册信息</button>
38         </div>
39     </form>
```

```
40 </body>
41 <script src="./js/jquery-3.6.1.js"></script>
42 <script src="./js/jquery.validate.js"></script>
43 <script src="./js/messages_zh.js"></script>
44 <script>
45     $(function () {
46         var registerForm = $("#registerForm").validate({
47             // 验证规则
48             rules: {
49                 userName: {
50                     ranglength: [6, 10],
51                     required: true
52                 },
53                 pwd: {
54                     required: true
55                 },
56                 pwd2: {
57                     required: true,
58                     equalTo: "#pwd"
59                 },
60                 age: {
61                     min: 0,
62                     required: true
63                 },
64                 hobby: {
65                     required: true
66                 }
67             },
68             // 验证消息
69             messages: {
70                 userName: {
71                     ranglength: "用户名长度必须是6~10之间",
72                     required: "用户名必填"
73                 },
74                 pwd: {
75                     required: "密码不能为空"
76                 },
77                 pwd2: {
78                     required: "确认密码不能为空",
79                     equalTo: "两次密码必须相同"
80                 },
81                 age: {
82                     min: "年龄必须是大于0的数",
83                     required: "年龄不能为空"
84                 },
85                 hobby: {
86                     required: "爱好不能为空"
87                 }
88             }
89         });
```

```

90
91     //当点击以后就开始验证
92     $("#btn-register").click(function () {
93         var result = registerForm.form();
94         if (result) {
95             console.log("验证通过");
96         }
97         else {
98             console.log("验证不能过");
99         }
100     })
101 })
102 </script>
103
104 </html>
105 <!--
106     用户名：长度必须是6-10位,必填
107     密码：不能为空
108     确认密码：必须与密码相同，也不能为空
109     年龄：正数
110     身份证：必须填入身份证号
111     爱好：不能为空
112
113     要求：点击按钮以后去完成上面的验证，怎么办
114     -->

```

像上面这种通过 **rules** 和 **messages** 来配置的 我们叫配置式的验证规则

在上面有个问题要注意一下，我们的身份证号好像还没有验证，因为这个插件里面还没有身份证号的验证规则，怎么办呢？

侵入式的验证规则

侵入式的验证规则是把验证的规则不写在js里面，而是直接通过自定义属性写在html代码里面，jQuery 表单验证里面新版本里面的特性

```

1  <!DOCTYPE html>
2  <html lang="zh">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>表单验证</title>
9  </head>
10
11 <body>
12     <form id="loginForm">
13         <div>

```

```

14         用户名: <input type="text" placeholder="请输入用户名"
      name="userName" data-rule-required="true"
15         data-msg-required="用户名不能为空">
16     </div>
17 </div>
18         密码: <input type="text" placeholder="请输入密码" name="pwd"
      data-rule-required="true"
19         data-msg-required="密码不能为空">
20     </div>
21 </div>
22     <button type="button" id="btn1">登录</button>
23 </div>
24 </form>
25 </body>
26 <script src="./js/jquery-3.6.1.js"></script>
27 <script src="./js/jquery.validate.js"></script>
28 <script src="./js/messages_zh.js"></script>
29 <script>
30     $(function () {
31         var loginForm = $("#loginForm").validate();
32         $("#btn1").click(function(){
33             var result = loginForm.form();
34             if(result){
35                 console.log("验证通过了");
36             }
37             else{
38                 console.log("很遗憾，验证不通过");
39             }
40         })
41     })
42 </script>
43
44 </html>

```

在上面的代码当中，我们没有把验证规则写在js的代码里面，而是写在了自定义属性 **data-** 里面

- **data-rule-xxx** 代表的是添加一个验证规则，如 **data-rule-required="true"** 或 **data-rule-min="0"**
- **data-msg-xxx** 代表的是当某一个规则校验不通过的时候的提示信息，如 **data-msg-required="用户名不能为空"**

无论是通过哪种方式来进行验证，表单元素上面的name的属性值都不能丢

自定义验证规则

虽然这个表单验证的插件帮助我们解决了很多验证的问题，但是仍然有些问题不能够实现，很多场景下面我们需要自定义校验规则，怎么办呢？

jQuery.validator 这个插件可以添加自定义的校验规则，最常见的就是把正则表达式的验证规则添加进去，所以这里我只展示正则表达式的扩展（其它的原理是一样的）

```
6. <meta http-equiv="X-UA-Compatible" content="IE=edge">
7. <meta name="viewport" content="width=device-width, initial-scale=1.0">
8. <title>自定义校验</title>
9. </head>
10. <body>
11. <form id="testForm">
12. <div>
13. 用户名: <input type="text" name="userName">
14. </div>
15. <div>
16. 身份证号: <input type="text" value="" name="IDCard"
17. data-rule-regexp="\d{18}$" >
18. </div>
19. <div>
20. 在上面的代码当中, 我们没有把验证规则写在 的代码里面, 而是写在了
21. <button type="button" id="btn">我要添加</button>
22. </div>
23. </form>
24. </body>
25. <script src="../js/jquery-3.6.1.js"></script>
26. <script src="../js/jquery.validate.js"></script>
27. <script src="../js/messages_zh.js"></script>
28. <script>
29. $(function() {
30. // 添加自定义的验证规则
31. $.validator.addMethod("regexp", function(value, element, params) {
32. // 这里的function就是你自定义的验证函数
33. // params代表的就是data-rule-regexp=" " 等号后面的内容
34. // value代表当前表单元素的内容
35. // element代表当前验证的元素
36. });
37. }
```

method
Type: [Function\(\)](#)
The actual method implementation, returning true if an element is valid. First argument: Current value. Second argument: Validated element. Third argument: Parameters.

value
Type: [String](#)
the current value of the validated element

element
Type: [Element](#)
the element to be validated

params
Type: [Object](#)
parameters specified for the method, e.g. for min: 5, the parameter is 5, for its [1, 5]

message

```
1 <!DOCTYPE html>
2 <html lang="zh">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>自定义校验</title>
9 </head>
10
11 <body>
12     <form id="testForm">
13         <div>
14             用户名: <input type="text" name="userName">
15         </div>
16         <div>
17             身份证号: <input type="text" name="IDCard"
18             data-rule-regexp="\d{18}$" data-msg-regexp="身份证号格式不
19 对">
20         </div>
21         <div>
22             <button type="button" id="btn1">我要添加</button>
23         </div>
24     </form>
25 </body>
26 <script src="../js/jquery-3.6.1.js"></script>
27 <script src="../js/jquery.validate.js"></script>
28 <script src="../js/messages_zh.js"></script>
29 <script>
30     $(function() {
31         // 添加自定义的验证规则
32         $.validator.addMethod("regexp", function(value, element, params) {
33             //这里的function就是你自定义的验证函数
```



```

33         //params代表的就是data-rule-regexp=""等号后面的内容
34         //value代表当前表单元素的内容(值)
35         //element代表当前验证元素
36         var reg = new RegExp(params);
37         return reg.test(value);
38     });
39
40     var testForm = $("#testForm").validate();
41     $("#btn1").click(function(event){
42         var result = testForm.form();
43         if(result){
44             console.log("验证通过");
45         }
46         else{
47             console.log("验证不通过");
48         }
49     })
50 })
51 </script>
52
53 </html>

```

自定义消息提示的样式

当我们的验证不通过以后会有一个消息提示，其实这个消息提示的样式，我们是可以自己定义的

```

1 var testForm = $("#testForm").validate({
2     errorClass:"error-text"
3 });

```

其中的 `error-text` 代表就是自己写的CSS样式，最终效果如下

用户名: 用户名不能为空

身份证号: 身份证号格式不对

自定义消息提示的位置

jQuery的表单验证默认情况下会把所有验证不通过的消息内容追加到当前验证元素的后面

```

1 var testForm = $("#testForm").validate({
2     errorClass:"error-text",
3     errorPlacement:function(error,element){
4         // error代表的就是这个错误提示的消息
5         // element代表你当前正在验证的这个元素
6         $(".box").append(error);
7     }
8 });

```

用户名：

身份证号：

我要添加

用户名不能为空
身份证号格式不对

上面的代码当中，它把我们的错误的消息放在了一个指定的盒子里面

作业与练习

请完成下面界面的布局，并完成表单验证的内容

新增学生信息

学生姓名

请输入学生姓名

学生姓名不能为空

学生性别

男

学生年龄不能为空

学生年龄

请输入学生年龄

手机号不能为空

手机号码

请输入手机号码

班级名称不能为空

所属班级

请输入班级名称

地址不能为空

家庭地址

请输入家庭地址

请至少选择一个爱好

爱好

☐ 看书

☐ 睡觉

☐ 玩游戏

☐ 逛街

☐ 写代码

保存数据

返回列表

有上面一个表单，当点击“保存数据”以后，要对表单做一次数据校验，校验的要求如下

参名数名	参数说明
sname	学生姓名
ssex	性别，只能填男与女
sage	必须是整数，并且必须是18~45之间
stel	手机号，不能为空
cname	班级名称，必须是H2101~H2109或J2101~J2109
hobby	通过多选得到，每个爱好之前使用英文的逗号隔开，如"看书,睡觉"
saddress	地址不能为空，最大只能输入100个字符

