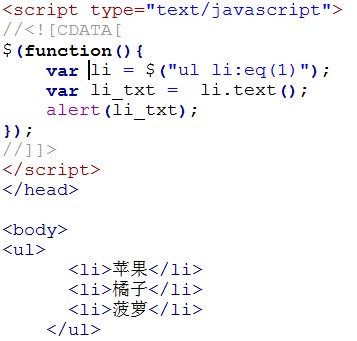
jQuery小记

1.$符号

在 jQuery 中，[美元](https://www.baidu.com/s?wd=%E7%BE%8E%E5%85%83&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Ydmym3rHD1mHRYmWfdmW--0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3ErjTkPWfLrj6)符号（$）仅仅是 jQuery 的别名，就代表了jQuery对象，例如，$("div") 和 jQuery("div")是等价的。  
注意两点：  
1、即使不使用 $ 也能保证jQuery的所有功能性。  
2、为了避免与其他javascript库的冲突，可以释放 jQuery 对 $ 变量的控制，同时为 jQuery 变量规定新的自定义名称。例如：  
执行 var jq=$.noConflict(); 后，$ 将不再控制当前的jQuery， 而是让渡给了jq变量，此时jq("div") 和 jQuery("div")是等价的。

1.1

# jquery中变量加$和不加$有什么区别

没有区别，只是习惯。  
一般在给jquery对象取名的时候在前面加$。 一看就知道是jquery对象。

**变量命名规则中起始字符可以是 字母,下划线(\_),美元符($)**,只是**很多的js库喜欢使用$作为全局变量标志**,jQuery也不例外.加上$,作为jQuery变量标志更容易进行区分.

- - || 结贴啦……  
我看到一种说法（见《Learning jQuery》第59页，中文版《jQuery基础教程》第4章）  
var $time = $('#time'); 用$开头命名变量是为了表示这是一个jQuery对象，仅此而已。  
就好像我们用小写s开头的变量名表示字符串，用i开头变量名表示整数，用o开头的变量名表示一个Object对象  
var sName = "Mike";   
var iNum = 10;  
var oPeople = GetPeople();  
如果我用j开头标识jQuery对象也可以， var jTime = $('#time'); 和 $time用没有什么区别的。

## 我觉得这可能是个人定义习惯，加上$代表这个是jquery的变量； 不加代表是js的变量。 方便区别，到时候转换的时候也不会出错

## 因为前端工资比较低，中医讲究缺啥补啥，所以就用 $ 了

## 2. jQuery 语法

jQuery 语法是为 HTML 元素的选取编制的，可以对元素执行某些操作。

基础语法是：*$(selector).action()*

* 美元符号定义 jQuery
* 选择符（selector）“查询”和“查找” HTML 元素
* jQuery 的 action() 执行对元素的操作

**示例**

$(this).hide() - 隐藏当前元素

$("p").hide() - 隐藏所有段落

$(".test").hide() - 隐藏所有 class="test" 的所有元素

$("#test").hide() - 隐藏所有 id="test" 的元素

提示：jQuery 使用的语法是 XPath 与 CSS 选择器语法的组合。

<html>

<head>

<script type="text/javascript" src="/jquery/jquery.js"></script>

<script type="text/javascript">

$(document).ready(function(){

$("button").click(function(){

$("#test").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>

<p id="test">This is another paragraph.</p>

<button type="button">Click me</button>

</body>

</html>

## 3. 文档就绪函数

您也许已经注意到在我们的实例中的所有 jQuery 函数位于一个 document ready 函数中：

$(document).ready(function(){

--- jQuery functions go here ----

});

这是为了防止文档在完全加载（就绪）之前运行 jQuery 代码。

如果在文档没有完全加载之前就运行函数，操作可能失败。下面是两个具体的例子：

* 试图隐藏一个不存在的元素
* 获得未完全加载的图像的大小

<!DOCTYPE html>

<html>

<head>

<script src="/jquery/jquery-1.11.1.min.js"></script>

<script type="text/javascript">

$(document).ready(function(){ //为了在DOM完全加载后才运行jQuery代码，都放这里面

$("#hide").click(function(){

$("p").hide();

});

$("#show").click(function(){

$("p").show();

});

});

</script>

</head>

<body>

<p id="p1">如果点击“隐藏”按钮，我就会消失。</p>

<button id="hide" type="button">隐藏</button>

<button id="show" type="button">显示</button>

</body>

</html>

## 4. jQuery 选择器

在前面的章节中，我们展示了一些有关如何**选取** HTML 元素的实例。

关键点是学习 jQuery 选择器是如何准确地选取您希望应用效果的元素。

jQuery **元素选择器**和**属性选择器**允许您通过**标签名**、**属性名**或**内容**对 HTML 元素进行选择。

选择器允许您对 **HTML 元素组**或**单个元素**进行操作。

在 HTML DOM 术语中：

选择器允许您对 **DOM 元素组**（指定id或class）或单个 DOM 节点进行操作。

**jQuery 元素选择器**

jQuery 使用 **CSS 选择器**来**选取 HTML 元素**。

$("p") 选取 <p> 元素。

$("p.intro") 选取所有 class="intro" 的 <p> 元素。

$("p#demo") 选取所有 id="demo" 的 <p> 元素。

**jQuery 属性选择器**

jQuery 使用 **XPath** **表达式**来选择**带有给定属性**的元素。

$("[href]") 选取**所有带有 href 属性的元素**。

$("[href='#']") 选取所有带有 href **值等于 "#" 的元素**。

$("[href!='#']") 选取所有带有 href 值**不等于** "#" 的元素。

$("[href$='.jpg']") 选取所有 href **值以 ".jpg" 结尾**的元素。

**jQuery CSS 选择器**

jQuery CSS 选择器可用于**改变 HTML 元素的 CSS 属性**。

下面的例子把所有 p 元素的背景颜色更改为红色：

**实例**

$("p")**.css**("background-color","red");

**更多的选择器实例**

|  |  |
| --- | --- |
| **语法** | **描述** |
| $(this) | 当前 HTML 元素 |
| $("p") | 所有 <p> 元素 |
| $("p.intro") | 所有 class="intro" 的 <p> 元素 |
| $(".intro") | 所有 class="intro" 的元素 |
| $("#intro") | id="intro" 的元素 |
| $("ul li:first") | 每个 <ul> 的第一个 <li> 元素 |
| $("[href$='.jpg']") | 所有带有以 ".jpg" 结尾的属性值的 href 属性 |
| $("div#intro .head") | id="intro" 的 <div> 元素中的所有 class="head" 的元素 |

5. **jQuery 是为事件处理特别设计的。**

**jQuery 事件函数**

jQuery 事件处理方法是 jQuery 中的核心函数。

事件处理程序指的是当 HTML 中发生某些事件时所调用的方法。术语由事件“触发”（或“激发”）经常会被使用。

通常会把 jQuery 代码放到 **<head>部分**的事件处理方法中：

**实例**

<html>

<head>

<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript">

$(document).ready(function(){

$("button").**click**(function(){

$("p").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

<button>Click me</button>

</body>

</html>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=jquery_hide_p)

在上面的例子中，当按钮的点击事件被触发时会调用一个函数：

$("button").click(function() {..some code... } )

该方法隐藏所有 <p> 元素：

$("p").hide();

**单独文件中的函数**

如果您的**网站包含许多页面**，并且您**希望**您的 **jQuery 函数易于维护**，那么**请把您的 jQuery 函数放到独立的 .js 文件中**。

当我们在教程中演示 jQuery 时，会将函数直接添加到 <head> 部分中。不过，把它们**放到一个单独的文件中会更好**，就像这样（通过 src 属性来引用文件）：

**实例**

**<head>**

<script type="text/javascript" src="jquery.js"></script>

<script type="text/javascript" src="my\_jquery\_functions.js"></script>

**</head>**

**jQuery 名称冲突**

jQuery 使用 $ 符号作为 jQuery 的简介方式。

某些**其他 JavaScript 库**（jQuery就是一个js库）中的函数（比如 Prototype）同样使用 $ 符号。

jQuery 使用名为 **noConflict()** 的方法来解决该问题。

*var jq=jQuery.noConflict()*，**帮助您使用自己的名称**（比如 jq）**来代替 $ 符号**。

<!DOCTYPE html>

<html>

<head>

<script src="/jquery/jquery-1.11.1.min.js"></script>

<script>

**var ix = $.noConflict()**;  **//使用ix代替$符号**

ix(document).ready(function(){

ix("button").click(function(){

ix("p").text("jQuery 仍在运行！");

});

});

</script>

</head>

<body>

<p>这是一个段落。</p>

<button>测试 jQuery</button>

</body>

</html>

**结论**

由于 jQuery 是为处理 HTML 事件而特别设计的，那么当您遵循以下原则时，您的代码会更恰当且更易维护：

* 把所有 jQuery 代码置于事件处理函数中
* 把所有事件处理函数置于文档就绪事件处理器中
* 把 jQuery 代码置于单独的 .js 文件中
* 如果存在名称冲突，则重命名 jQuery 库

**jQuery 事件**

下面是 jQuery 中事件方法的一些例子：

|  |  |
| --- | --- |
| **Event 函数** | **绑定函数至** |
| $(document).ready(function) | 将函数绑定到文档的就绪事件（当文档完成加载时） |
| $(selector).click(function) | 触发或将函数绑定到被选元素的点击事件 |
| $(selector).dblclick(function) | 触发或将函数绑定到被选元素的双击事件 |
| $(selector).focus(function) | 触发或将函数绑定到被选元素的获得焦点事件 |
| $(selector).mouseover(function) | 触发或将函数绑定到被选元素的鼠标悬停事件 |

如需完整的参考手册，请访问我们的 [jQuery 事件参考手册](http://www.w3school.com.cn/jquery/jquery_ref_events.asp)。

6. 平台使用jQuery inline form validation插件来处理校验。

参考文档：<http://www.position-relative.net/creation/formValidator/>

# 使用

插件提供的基本校验参考：

jquery.validationEngine-zh\_CN.js

位置：

\dssp-ui-bootstrap\3.0.5\dssp-ui-bootstrap-3.0.5\META-INF\resources\assets\js\plugin\formValidationEngine\js\languages

所有的form元素只需要添加class="validate[required]"即可以校验必填

详细说明：

|  |  |  |
| --- | --- | --- |
| **样式** | **说明** | **例子** |
| required | 必填 | class="validate[required]" |
| dateRange | 日期范围 |  |
| dateTimeRange | 时间范围 |  |
| minSize | 最少输入 | validate[required,minSize[6]] |
| maxSize | 最多输入 | validate[required,maxSize[6]] |
| groupRequired | 组选择 | validate[groupRequired[payments]] |
| min | 最小数字 | validate[required,min[6]] |
| max | 最大数字 | validate[required,max[6]] |
| largerThan | 必须大于 | validate[required,largerThan[6]] |
| past | 日期不能大于 | validate[custom[date],past[2010/01/01]] |
| future | 日期不能小于 | validate[custom[date],future[2010/01/01]] |
| digitNum | 小数点后不能超过 |  |
| maxCheckbox | 最多选择 |  |
| minCheckbox | 最少选择 |  |
| equals | 相等 |  |
| creditCard | 信用卡 |  |
| phone | 电话号码 | validate[required,custom[phone]] |
| email | 邮箱 | validate[required,custom[email]] |
| integer | 整数 |  |
| number | 数字 |  |
| date | 时间 |  |
| ipv4 | ipv4 |  |
| url | 链接 | validate[required,custom[url]] |
| onlyNumberSp | 仅数字 |  |
| onlyLetterSp | 仅字母 |  |
| onlyLetterNumber | 数字字母组合 |  |

# 自定义扩展

## 简单扩展

修改上面的js（jquery.validationEngine-zh\_CN.js）文件

添加一个自定义的校验

**自定义校验**

"percent100":{

   "regex":/^(([1-9]\d{0,1})|(0)|(100))(\.\d{1,2})?$/,

   "alertText": "\* 请输入0~100的数字，最多2位小数,示例:69.21"

},

调用

**自定义校验**

validate[custome[percent100]]

## 函数

自定义函数

**自定义校验函数**

function repass(field, rules, i, options) {

    var pass = $("#userPassword").val();

    return field.val() != pass ? "密码不一致！" : true;

}

调用

**调用**

validate[required, funcCall[repass]]

**7.** $(function(){}) 是 $(document).ready(function(){}) 的简写，用来在DOM加载完成之后执行一系列预先定义好的函数。

$(function(){ } 这是jquery里的,是当文档载入完毕就执行,的意思.  
  
$(function(){   
$("#treeDiv1").load("composer.jsp?type=list");  
$("#treeDiv2").load("composer.jsp?type=list2");  
});  
  
这个就是当文档加载完成,就执行  
$("#treeDiv1").load("composer.jsp?type=list");  
$("#treeDiv2").load("composer.jsp?type=list2");  
这两个语句,  
  
其实是跟较早的版本的$(document).ready(function(){ ...... }) 是一样的

(function(){})表示一个[匿名函数](https://www.baidu.com/s?wd=%E5%8C%BF%E5%90%8D%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3PhnsmhnkPHm4nvc1nhnL0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm3nj04njnL)。function(arg){...}定义了一个参数为arg的[匿名函数](https://www.baidu.com/s?wd=%E5%8C%BF%E5%90%8D%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3PhnsmhnkPHm4nvc1nhnL0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm3nj04njnL)，然后使用(function(arg){...})(param)来调用这个[匿名函数](https://www.baidu.com/s?wd=%E5%8C%BF%E5%90%8D%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3PhnsmhnkPHm4nvc1nhnL0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHm3nj04njnL)。其中param是传入这个匿名函数的参数。  
　　需要注意与$(function(){})的区别：$(function(){}) 是 $(document).ready(function(){}) 的简写，用来在DOM加载完成之后执行一系列预先定义好的函数。

# 8. jQuery 属性操作 - val() 方法

## 实例

设置输入域的值：

$("button").click(function(){

$(":text").val("Hello World");

});

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=jquery_attributes_val)

## 定义和用法

val() 方法返回或设置被选元素的值。

元素的值是通过 value 属性设置的。该方法大多用于 input 元素。

如果该方法未设置参数，则返回被选元素的当前值。

### 语法

$(*selector*).val(*value*)

|  |  |
| --- | --- |
| **参数** | **描述** |
| *value* | 可选。规定被选元素的新内容。 |

## 返回 Value 属性

返回第一个匹配元素的 value 属性的值。

### 语法

$(*selector*).val()

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=jquery_attributes_val_get)

## 设置 Value 属性的值

$(*selector*).val(*value*)

|  |  |
| --- | --- |
| **参数** | **描述** |
| *value* | 设置 Value 属性的值。 |

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=jquery_attributes_val_set)

## 使用函数设置 Value 属性的值

$(*selector*).val(*function(index,oldvalue)*)

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=jquery_attributes_val_func)

|  |  |
| --- | --- |
| **参数** | **描述** |
| *function(index,oldvalue)* | 规定返回要设置的值的函数。   * *index* - 可选。接受选择器的 index 位置。 * *oldvalue* - 可选。接受选择器的当前 Value 属性。 |

9.