

---

# JQuery

安装jQuery

jQuery选择器

jQuery事件

为了简化 JavaScript 的开发，一些 JavaScript 库诞生了。  
JavaScript 库封装了很多预定义的对象和实用函数。能帮助使用者建立有高难度交互的 Web2.0 特性的富客户端页面，并且兼容各大浏览器

当前流行的 JavaScript 库有：

jQuery, MooTools, Prototype, Dojo, YUI, EXT\_JS DWR

jQuery由美国人John Resig创建，至今已吸引了来自世界各地的众多javascript高手加入其team。

jQuery是继prototype之后又一个优秀的Javascript框架。其宗旨是——**写更少的代码，做更多的事情。**

它是**轻量级的js库**（压缩后只有21k），这是其它的js库所不及的，它兼容CSS3，还**兼容各种浏览器**（IE 6.0+，FF 1.5+，Safari 2.0+，Opera 9.0+）。

jQuery是一个快速的，简洁的javascript库，使用户能更方便地处理HTML documents、events、实现**动画效果**，并且方便地为网站提供**AJAX交互**。

jQuery还有一个比较大的优势是，它的**文档说明很全**，而且各种应用也说得很详细，同时还有许多**成熟的插件**可供选择。

jQuery能够使用户的html页保持**代码和html内容分离**，也就是说，**不用再在html里面插入一堆js来调用命令了，只需定义id即可。**

jQuery 对象就是通过jQuery包装DOM对象后产生的对象。

jQuery 对象是 jQuery 独有的。如果一个对象是 jQuery 对象，那么它就可以使用 jQuery 里的方法：`$("#test").html();`

比如：

`$("#test").html()` 意思是指：获取ID为test的元素内的html代码。其中html()是jQuery里的方法

这段代码等同于用DOM实现代码：

`document.getElementById(" test ").innerHTML;`

虽然jQuery对象是包装DOM对象后产生的，但是jQuery无法使用DOM对象的任何方法，同理DOM对象也不能使用jQuery里的方法。乱使用会报错

约定：如果获取的是 jQuery 对象，那么要在变量前面加上 `$.`

`var $variable = jQuery 对象`

`var variable = DOM 对象`

对于已经是一个DOM对象，只需要用\$()把DOM对象包装起来，就可以获得一个jQuery对象了。\$(DOM对象)

```
//<input type="text" name="username" id="username" value="张老师" />
//获取DOM对象
var username=document.getElementById("username");
alert(username.value);

//转化为jquery对象
var $username=$(username);
alert("^^^ "+$username.val());
```

转换后就可以使用 jQuery 中的方法了

两种转换方式将一个jQuery对象转换成DOM对象：[index]和.get(index);

(1) jQuery对象是一个数组对象，可以通过[index]的方法，来得到相应的DOM对象

```
//<input type="text" name="username" id="username" value="张老师" />
//获取jQuery对象
var $username=$("#username")
alert($username.val());

//转化为DOM对象
var username=$username[0];
alert(username.value);
```

(2) jQuery本身提供，通过.get(index)方法，得到相应的DOM对象

```
//转化为DOM对象
var username1=$username.get(0);
alert(username1.value);
```

选择器是 jQuery 的根基，在 jQuery 中，对事件处理，遍历 DOM 和 Ajax 操作都依赖于选择器

jQuery 选择器的优点：

简洁的写法

完善的事件处理机制

多样的选择方法



基本选择器是 jQuery 中最常用的选择器，也是最简单的选择器，它通过元素 id, class 和标签名来查找 DOM 元素(在网页中 id 只能使用一次，class 允许重复使用)。

1、#id            用法: `$("#myDiv");`

说明: 直接选择html中的id="myDiv"

2、Element        用法: `$("div")`

说明: html已经定义的标签元素,例如 div, input, a 等等.

3、class            用法: `$(".myClass")`

说明: 选择html代码中class="myClass"的元素或元素组.

4、\*                用法: `$("*")`

说明: 匹配所有元素,多用于结合上下文来搜索

5、selector1, selector2, selectorN            用法:  
`$("div,span,p.myClass")`

说明: 将每一个选择器匹配到的元素合并后一起返回.你可以指定任意多个选择器, 并将匹配到的元素合并到一个结果内.

如果想通过 DOM 元素之间的层次关系来获取特定元素，例如后代元素，子元素，相邻元素，兄弟元素等，则需要使用层次选择器。

1、ancestor descendant      用法: `$("form input")` ;

说明: 在给定的祖先元素下匹配所有后代元素。

2、parent > child      用法: `$("form > input")` ;

说明: 在给定的父元素下匹配所有子元素。

3、prev + next      用法: `$("label + input")` ;

说明: 匹配所有紧接在 prev 元素后的 next 元素

4、prev ~ siblings      用法: `$("form ~ input")` ;

说明: 匹配 prev 元素之后的所有 siblings 元素.注意:是匹配之后的元素,不包含该元素在内,并且siblings匹配的是和prev同辈的元素,其后辈元素不被匹配。

注意: (“prev ~ div”) 选择器只能选择 “# prev ” 元素后面的同辈元素; 而 jQuery 中的方法 `siblings()` 与前后位置无关, 只要是同辈节点就可以选取

**过滤选择器**主要是通过特定的过滤规则来筛选出所需的 DOM 元素，该选择器都以 “:” 开头

按照不同的过滤规则，过滤选择器可以分为

基本过滤，

内容过滤，

可见性过滤，

属性过滤，

子元素过滤

表单对象属性过滤选择器。

1、**:first**      用法: `$("tr:first")` ;

说明: 匹配找到的第一个元素

2、**:last**      用法: `$("tr:last")`

说明: 匹配找到的最后一个元素.与 **:first** 相对应.

3、**:not(selector)**      用法: `$("input:not(:checked)")`

说明: 去除所有与给定选择器匹配的元素.有点类似于“非”.

4、**:even**      用法: `$("tr:even")`

说明: 匹配所有索引值为偶数的元素.

5、**: odd**      用法: `$("tr:odd")`

说明: 匹配所有索引值为奇数的元素,和**:even**对应,从 0 开始计数.

6、**:eq(index)** 用法: `$("tr:eq(0)")`

说明: 匹配一个给定索引值的元素.

7、**:gt(index)** 用法: `$("tr:gt(0)")`

说明: 匹配所有大于给定索引值的元素.

8、**:lt(index)** 用法: `$("tr:lt(2)")`

说明: 匹配所有小于给定索引值的元素.

9、**:header(固定写法)**用法: `$(":header").css("background", "#EEE")`

说明: 匹配如 **h1, h2, h3**之类的标题元素.

内容过滤选择器的过滤规则主要体现在它所包含的子元素和文本内容上

## 1、:contains(text)

用法: `$("div:contains('John')")`      返回值    集合元素

说明: 匹配包含给定文本的元素. 这个选择器比较有用, 当我们要选择的不是dom标签元素时, 它就派上了用场了, 它的作用是查找被标签“围”起来的文本内容是否符合指定的内容的。

## 2、:empty

用法: `$("td:empty")`      返回值    集合元素

说明: 匹配所有不包含子元素或者文本的空元素

## 3、:has(selector)

用法: `$("div:has(p")).addClass("test")`      返回值    集合元素

说明: 匹配含有选择器所匹配的元素. 这个解释需要好好琢磨, 但是一旦看了使用的例子就完全清楚了: 给所有包含p元素的div标签加上class="test".

## 4、:parent

用法: `$("td:parent")`      返回值    集合元素

说明: **匹配含有子元素或者文本的元素**. 注意: 这里是“:parent”, 可不是“.parent”哦! 感觉与上面讲的“:empty”形成反义词。

可见度过滤选择器是**根据元素的可见和不可见状态**来选择相应的元素

## 1、:hidden

**用法:** `$("tr:hidden")`    **返回值**    **集合元素**

说明: 匹配所有的不可见元素, input 元素的 type 属性为 "hidden" 的话也会被匹配到.意思是css中display:none和input type="hidden"的都会被匹配到.

## 2、:visible

**用法:** `$("tr:visible")`    **返回值**    **集合元素**

说明: 匹配所有的可见元素.

属性过滤选择器的过滤规则是**通过元素的属性来获取相应的元素**

1、[attribute]            用法: \$("div[id]" );

说明: **匹配包含给定属性的元素**. 例子中是选取了所有带" id" 属性的div标签.

2、[attribute=value]

用法: \$("input[name='newsletter']").attr("checked", true);

说明: **匹配给定的属性是某个特定值的元素**.

3、[attribute!=value]

用法: \$("input[name!='newsletter']").attr("checked", true);

说明: **匹配所有不含有指定的属性, 或者属性不等于特定值的元素**.

4、[attribute^=value]            用法: \$("input[name^='news']")

说明: **匹配给定的属性是以某些值开始的元素**.

## 5、[attribute\$=value]

用法: `$( "input[name$='letter']" )` 返回值 集合元素

说明: **匹配给定的属性是以某些值结尾的元素.**

## 6、[attribute\*=value]

用法: `$( "input[name*='man']" )` 返回值 集合元素

说明: **匹配给定的属性是以包含某些值的元素.**

## 7、[attributeFilter1][attributeFilter2][attributeFilterN]

用法: `$( "input[id][name$='man']" )` 返回值 集合元素

说明: **复合属性选择器,需要同时满足多个条件时使用. 又是一个组合,** 这种情况我们实际使用的时候很常用. 这个例子中选择的是所有含有 id 属性, 并且它的 name 属性是以 man 结尾的元素.



`:nth-child(index/even/odd/equation)`

用法: `$("ul li:nth-child(2)")`

说明: **匹配其父元素下的第N个子或奇偶元素**. 这个选择器和之前说的[基础过滤 \(Basic Filters\)](#)中的 `eq()` 有些类似, 不同的地方就是**前者是从0开始, 后者是从1开始**.

`:first-child`      用法: `$("ul li:first-child")`

说明: **匹配第一个子元素**. `:first` 只匹配一个元素, 而此选择符将为每个父元素匹配一个子元素. 这里需要特别点的记忆下区别.

`:last-child`      用法: `$("ul li:last-child")`

说明: **匹配最后一个子元素**. `:last` 只匹配一个元素, 而此选择符将为每个父元素匹配一个子元素.

`:only-child`      用法: `$("ul li:only-child")`

说明: **如果某个元素是父元素中唯一的子元素, 那将会被匹配**. 如果父元素中含有其他元素, 那将不会被匹配. 意思就是: 只有一个子元素的才会被匹配!

此选择器主要对所选择的表单元素进行过滤

1、:enabled      用法: \$("input:enabled")

说明: **匹配所有可用元素**.意思是查找所有input中不带有disabled="disabled"的input.不为disabled,当然就为enabled啦.

2、:disabled      用法: \$("input:disabled")

说明: **匹配所有不可用元素**.与上面的那个是相对应的.

3、:checked      用法: \$("input:checked")

说明: **匹配所有选中的被选中元素(复选框、单选框等)**,不包括select中的option).这话说起来有些绕口.

4、:selected      用法: \$("select option:selected")

说明: **匹配所有选中的option元素**.

1、**:input**      用法: `$(":input")` ;

说明: 匹配所有 **input**, **textarea**, **select** 和 **button** 元素

2、**:text**      用法: `$(":text")` ;

说明: 匹配所有的单行文本框.

3、**:password**    用法: `$(":password")` ;

说明: 匹配所有密码框.

4、**:radio**      用法: `$(":radio")` ;

说明: 匹配所有单选按钮.

5、**:checkbox**      用法: `$(":checkbox")` ;

说明: 匹配所有复选框

6、**:submit**      用法: `$(":submit")` ;

说明: 匹配所有提交按钮.

7、**:image**      用法: `$(":image")`

说明: 匹配所有图像域.

8、**:reset**      用法: `$(":reset")` ;

说明: 匹配所有重置按钮.

9、**:button**      用法: `$(":button")` ;

说明: 匹配所有按钮. 这个包括直接写的元素 **button**.

10、**:file**      用法: `$(":file")` ;

说明: 匹配所有文件域.

11、**:hidden**      用法: `$("input:hidden")` ;

说明: 匹配所有不可见元素, 或者 **type** 为 **hidden** 的元素. 这个选择器就不仅限于表单了, 除了匹配 **input** 中的 **hidden** 外, 那些 **style** 为 **hidden** 的也会被匹配.

## 使用jQuery事件处理函数的好处是：

### 1. 添加的是多播事件委托

我们可以为dom的click事件添加一个函数后,再次添加一个函数.

### 2. 统一了事件名称

添加多播委托时,ie在事件名称前加了"on",而firefox则直接使用事件名称

### 3. 可以将对象行为全部用脚本控制

使用脚本控制元素行为,使用HTML标签控制元素内容,用CSS控制元素样式,从而达到了元素的行为、内容、样式分离的状态.

# jQuery 中的事件 -- 加载 DOM

在页面加载完毕后，浏览器会通过 JavaScript 为 DOM 元素添加事件。在常规的 JavaScript 代码中，通常使用 `window.onload` 方法，在 jQuery 中使用 `$(document).ready()` 方法。

方法	<code>window.onload</code>	<code>\$(document).ready()</code>
执行时机	必须等待网页中的所有内容加载完毕后(包括图片)才能执行	网页中的所有 DOM 结构绘制完毕后就执行, 可能 DOM 元素关联的东西并没有加载完
编写个数	不能同时编写多个	能同时编写多个
简化写法	无	<code>\$()</code>

我们可以使用事件处理函数完成对象事件的几乎所有操作,但jQuery提供了更加简单的办法来封装事件.

在jQuery中,单击事件对应着的两个方法是`click()`和`click(fn)`,分别用来触发单击事件和设置单击事件

- blur
- change
- click
- dblclick
- error
- focus
- keydown
- keypress

- keyup
- load
- mousedown
- mousemove
- mouseover
- mouseup
- resize
- scroll
- select
- submit
- unload

```
$( "elem" ).on( events, [selector], [data], handler );
```

## 参数解释：

- **events:**

事件名称，可以是自定义事件名称

- **selector:**

选择器

- **data:**

事件触发时传递给事件处理函数

- **handler:**

事件处理函数



`off([evts[,sel][,fn]])` 解除事件绑定

`one(type,[data],fn)` 为匹配元素的指定事件添加一次性事件处理函数

`trigger(event,[data])` 在匹配的元素上触发某类事件此函数。会导致浏览器同名的默认行为被执行

`triggerHandler(event,[data])` 触发指定的事件类型上所绑定的处理函数。不会执行浏览器默认行为

## 1. 登录：按钮的事件、用户名和密码的获取、样式根据焦点的改变而改变



网站后台管理系统

账号 :

密码 :

验证码 :  

☐ 记住我, 一周自动登录

## 2.主界面：实现装备的界面

用户管理

商品管理

商品分类

商品列表

添加商品

订单管理

查找产品：  S级赛车 查找 添加

<input type="checkbox"/>	序号	商品图片	商品名称	所属分类	商品价格	商品库存	商品ID	操作
<input type="checkbox"/>	1		CNM	宝马系摩托	0.00	1000	212	 
<input type="checkbox"/>	2		狗屎	B级赛车手	0.00	1000	777	 
<input type="checkbox"/>	3		歪歪鱼1	A级赛车	0.00	1000	7777	 
<input type="checkbox"/>	4		歪歪鱼	A级赛车	0.00	1000	7777	 
<input type="checkbox"/>	5		歪歪鱼	A级赛车	0.00	1000	7777	 

批量删除

共 5 页 当前第 1/5 [首页] [上一页] [下一页] [末页] 跳转到： -请选择 Go