

04-jQuery

讲师：王振国

今日任务

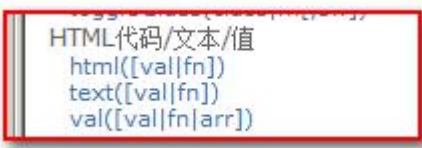
- 1.jQuery 的属性操作
- 2.jQuery 练习
- 3.jQuery CSS 样式操作
- 4.jQuery 动画操作
- 5.jQuery 事件操作

课堂笔记

今天课程内容：

1、jQuery 的属性操作

jQuery 属性操作



```
HTML代码/文本/值  
html([val|fn])  
text([val|fn])  
val([val|fn|arr])
```

html()	它可以设置和获取起始标签和结束标签中的内容。	跟 dom 属性 innerHTML 一样。
text()	它可以设置和获取起始标签和结束标签中的文本。	跟 dom 属性 innerText 一样。
val()	它可以设置和获取表单项的 value 属性值。	跟 dom 属性 value 一样

val 方法同时设置多个表单项的选中状态：

```
<!DOCTYPE html>  
<html lang="zh_CN">  
<head>  
  <meta charset="UTF-8">  
  <title>Title</title>
```

```
<script type="text/javascript" src="script/jquery-1.7.2.js"></script>
<script type="text/javascript">

    $(function () {

/*
        // 批量操作单选
        $(":radio").val(["radio2"]);
        // 批量操作筛选框的选中状态
        $(".checkbox").val(["checkbox3", "checkbox2"]);
        // 批量操作多选的下拉框选中状态
        $("#multiple").val(["mul2", "mul3", "mul4"]);
        // 操作单选的下拉框选中状态
        $("#single").val(["sin2"]);
*/

$("#multiple,#single,:radio,:checkbox").val(["radio2", "checkbox1", "checkbox3", "mul1", "mul4", "sin3"]);

    });

</script>
</head>
<body>
<body>
    单选:
    <input name="radio" type="radio" value="radio1" />radio1
    <input name="radio" type="radio" value="radio2" />radio2
    <br/>
    多选:
    <input name="checkbox" type="checkbox" value="checkbox1" />checkbox1
    <input name="checkbox" type="checkbox" value="checkbox2" />checkbox2
    <input name="checkbox" type="checkbox" value="checkbox3" />checkbox3
    <br/>

    下拉多选 :
    <select id="multiple" multiple="multiple" size="4">
        <option value="mul1">mul1</option>
        <option value="mul2">mul2</option>
        <option value="mul3">mul3</option>
        <option value="mul4">mul4</option>
    </select>
    <br/>

    下拉单选 :
    <select id="single">
        <option value="sin1">sin1</option>
        <option value="sin2">sin2</option>
        <option value="sin3">sin3</option>
    </select>
</body>
```

```
</body>
</html>
```



attr() 可以设置和获取属性的值，不推荐操作 checked、readOnly、selected、disabled 等等

attr 方法还可以操作非标准的属性。比如自定义属性：abc,bbj

prop() 可以设置和获取属性的值,只推荐操作 checked、readOnly、selected、disabled 等等

2、jQuery 练习

2.全选，全不选，反选

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src="../../script/jquery-1.7.2.js"></script>
<script type="text/javascript">

    $(function(){
        // 给全选绑定单击事件
        $("#checkedAllBtn").click(function () {
            $(".checkbox").prop("checked",true);
        });

        // 给全不选绑定单击事件
        $("#checkedNoBtn").click(function () {
            $(".checkbox").prop("checked",false);
        });

        // 反选单击事件
        $("#checkedRevBtn").click(function () {
```

```
// 查询全部的球类的复选框
$(":checkbox[name='items']").each(function () {
    // 在 each 遍历的 function 函数中, 有一个 this 对象。这个 this 对象是当前正在遍历到的 dom 对象
    this.checked = !this.checked;
});

// 要检查 是否满选
// 获取全部的球类个数
var allCount = $(":checkbox[name='items']").length;
// 再获取选中的球类个数
var checkedCount = $(":checkbox[name='items']:checked").length;

// if (allCount == checkedCount) {
//     $("#checkedAllBox").prop("checked", true);
// } else {
//     $("#checkedAllBox").prop("checked", false);
// }

$("#checkedAllBox").prop("checked", allCount == checkedCount);

});

// 【提交】按钮单击事件
$("#sendBtn").click(function () {
    // 获取选中的球类的复选框
    $(":checkbox[name='items']:checked").each(function () {
        alert(this.value);
    });
});

// 给【全选/全不选】绑定单击事件
$("#checkedAllBox").click(function () {

    // 在事件的 function 函数中, 有一个 this 对象, 这个 this 对象是当前正在响应事件的 dom 对象
    // alert(this.checked);

    $(":checkbox[name='items']").prop("checked", this.checked);
});

// 给全部球类绑定单击事件
$(":checkbox[name='items']").click(function () {
    // 要检查 是否满选
    // 获取全部的球类个数
    var allCount = $(":checkbox[name='items']").length;
    // 再获取选中的球类个数
    var checkedCount = $(":checkbox[name='items']:checked").length;

    $("#checkedAllBox").prop("checked", allCount == checkedCount);
});
```

```
});

</script>
</head>
<body>

  <form method="post" action="">

    你爱好的运动是? <input type="checkbox" id="checkedAllBox" />全选/全不选

    <br />
    <input type="checkbox" name="items" value="足球" />足球
    <input type="checkbox" name="items" value="篮球" />篮球
    <input type="checkbox" name="items" value="羽毛球" />羽毛球
    <input type="checkbox" name="items" value="乒乓球" />乒乓球
    <br />
    <input type="button" id="checkedAllBtn" value="全 选" />
    <input type="button" id="checkedNoBtn" value="全不选" />
    <input type="button" id="checkedRevBtn" value="反 选" />
    <input type="button" id="sendBtn" value="提 交" />
  </form>

</body>
</html>
```

4、DOM 的增删改



内部插入:

appendTo()

a.appendTo(b)

把 a 插入到 b 子元素末尾，成为最后一个子元素

prependTo()	a.prependTo(b)	把 a 插到 b 所有子元素前面，成为第一个子元素
-------------	----------------	---------------------------

外部插入:

insertAfter()	a.insertAfter(b)	得到 ba
insertBefore()	a.insertBefore(b)	得到 ab

替换:

replaceWith()	a.replaceWith(b)	用 b 替换掉 a	所有的替换成1个
replaceAll()	a.replaceAll(b)	用 a 替换掉所有 b	

删除:

remove()	a.remove();	删除 a 标签
empty()	a.empty();	清空 a 标签里的内容

5、jQuery 练习二

2.从左到右，从右到左练习

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<style type="text/css">
  select {
    width: 100px;
    height: 140px;
  }

  div {
    width: 130px;
    float: left;
    text-align: center;
  }
</style>
<script type="text/javascript" src="script/jquery-1.7.2.js"></script>
<script type="text/javascript">
  // 页面加载完成
  $(function () {
```

```
// 第一个按钮 【选中添加到右边】
$("#button:eq(0)").click(function () {
    $("#select:eq(0) option:selected").appendTo($("#select:eq(1)"));
});
// 第二个按钮 【全部添加到右边】
$("#button:eq(1)").click(function () {
    $("#select:eq(0) option").appendTo($("#select:eq(1)"));
});

// 第三个按钮 【选中删除到左边】
$("#button:eq(2)").click(function () {
    $("#select:eq(1) option:selected").appendTo($("#select:eq(0)"));
});

// 第四个按钮 【全部删除到左边】
$("#button:eq(3)").click(function () {
    $("#select:eq(1) option").appendTo($("#select:eq(0)"));
});
});
</script>
</head>
<body>

<div id="left">
    <select multiple="multiple" name="sel01">
        <option value="opt01">选项 1</option>
        <option value="opt02">选项 2</option>
        <option value="opt03">选项 3</option>
        <option value="opt04">选项 4</option>
        <option value="opt05">选项 5</option>
        <option value="opt06">选项 6</option>
        <option value="opt07">选项 7</option>
        <option value="opt08">选项 8</option>
    </select>

    <button>选中添加到右边</button>
    <button>全部添加到右边</button>
</div>
<div id="righth">
    <select multiple="multiple" name="sel02">
    </select>
    <button>选中删除到左边</button>
    <button>全部删除到左边</button>
</div>

</body>
</html>
```

3. 动态添加、删除表格记录

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Untitled Document</title>
<link rel="stylesheet" type="text/css" href="styleB/css.css" />
<script type="text/javascript" src="../../script/jquery-1.7.2.js"></script>
<script type="text/javascript">

    $(function () {

        // 创建一个用于复用的删除的 function 函数
        var deleteFun = function(){

            // alert("删除 操作 的 function : " + this);

            // 在事件响应的 function 函数中, 有一个 this 对象。这个 this 对象是当前正在响应事件的 dom 对象。
            var $trObj = $(this).parent().parent();

            var name = $trObj.find("td:first").text();

            /**
             * confirm 是 JavaScript 语言提供的一个确认提示框函数。你给它传什么, 它就提示什么<br/>
             * 当用户点击了确定, 就返回 true。当用户点击了取消, 就返回 false
             */
            if( confirm("你确定要删除[" + name + "]吗? ") ){
                $trObj.remove();
            }

            // return false; 可以阻止 元素的默认行为。
            return false;

        };

        // 给【Submit】按钮绑定单击事件
        $("#addEmpButton").click(function () {
            // 获取输入框, 姓名, 邮箱, 工资的内容
            var name = $("#empName").val();
            var email = $("#email").val();
            var salary = $("#salary").val();

            // 创建一个行标签对象, 添加到显示数据的表格中
            var $trObj = $("<tr>" +
```



```
"<td>" + name + "</td>" +  
"<td>" + email + "</td>" +  
"<td>" + salary + "</td>" +  
"<td><a href=\"deleteEmp?id=002\">Delete</a></td>" +  
"</tr>");
```

```
// 添加到显示数据的表格中
```

```
$trObj.appendTo( $("#employeeTable") );
```

```
// 给添加的行的 a 标签绑上事件
```

```
$trObj.find("a").click( deleteFun );
```

```
});
```

```
// 给删除的 a 标签绑定单击事件
```

```
$("a").click( deleteFun );
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<table id="employeeTable">
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Email</th>
```

```
<th>Salary</th>
```

```
<th>&nbsp;</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Tom</td>
```

```
<td>tom@tom.com</td>
```

```
<td>5000</td>
```

```
<td><a href="deleteEmp?id=001">Delete</a></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Jerry</td>
```

```
<td>jerry@sohu.com</td>
```

```
<td>8000</td>
```

```
<td><a href="deleteEmp?id=002">Delete</a></td>
```

```
</tr>
```

```
<tr>
```

```
<td>Bob</td>
```

```
<td>bob@tom.com</td>
<td>10000</td>
<td><a href="deleteEmp?id=003">Delete</a></td>
</tr>
</table>

<div id="formDiv">

<h4>添加新员工</h4>

<table>
  <tr>
    <td class="word">name: </td>
    <td class="inp">
      <input type="text" name="empName" id="empName" />
    </td>
  </tr>
  <tr>
    <td class="word">email: </td>
    <td class="inp">
      <input type="text" name="email" id="email" />
    </td>
  </tr>
  <tr>
    <td class="word">salary: </td>
    <td class="inp">
      <input type="text" name="salary" id="salary" />
    </td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <button id="addEmpButton" value="abc">
        Submit
      </button>
    </td>
  </tr>
</table>

</div>

</body>
</html>
```

6、CSS 样式操作。

addClass()	添加样式
removeClass()	删除样式
toggleClass()	有就删除，没有就添加样式。
offset()	获取和设置元素的坐标。

7、jQuery 动画

基本动画

show()	将隐藏的元素显示
hide()	将可见的元素隐藏。
toggle()	可见就隐藏，不可见就显示。

以上动画方法都可以添加参数。

- 1、第一个参数是动画 执行的时长，以毫秒为单位
- 2、第二个参数是动画的回调函数 (动画完成后自动调用的函数)

淡入淡出动画

fadeIn()	淡入（慢慢可见）
fadeOut()	淡出（慢慢消失）
fadeTo()	在指定时长内慢慢的将透明度修改到指定的值。0 透明，1 完成可见，0.5 半透明
fadeToggle()	淡入/淡出 切换

练习 06、CSS_动画 品牌展示

需求：

- 1.点击按钮的时候，隐藏和显示卡西欧之后的品牌。
- 2.当显示全部内容的时候，按钮文本为“显示精简品牌”
然后，小三角形向上。所有品牌产品为默认颜色。
- 3.当只显示精简品牌的时候，要隐藏卡西欧之后的品牌，按钮文本为“显示全部品牌”
然后小三形向下。并且把 佳能，尼康的品牌颜色改为红色（给 li 标签添加 promoted 样式即可）

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>品牌展示练习</title>
<style type="text/css">
* {
    margin: 0;
    padding: 0;
}

body {
    font-size: 12px;
    text-align: center;
}

a {
    color: #04D;
    text-decoration: none;
}

a:hover {
    color: #F50;
    text-decoration: underline;
}

.SubCategoryBox {
    width: 600px;
    margin: 0 auto;
    text-align: center;
    margin-top: 40px;
}

.SubCategoryBox ul {
    list-style: none;
}

.SubCategoryBox ul li {
    display: block;
    float: left;
    width: 200px;
    line-height: 20px;
}

.showmore , .showless{
    clear: both;
    text-align: center;
    padding-top: 10px;
}

.showmore a , .showless a{
    display: block;
    width: 120px;
```

```

margin: 0 auto;
line-height: 24px;
border: 1px solid #AAA;
}

.showmore a span {
padding-left: 15px;
background: url(img/down.gif) no-repeat 0 0;
}

.showless a span {
padding-left: 15px;
background: url(img/up.gif) no-repeat 0 0;
}

.promoted a {
color: #F50;
}
</style>
<script type="text/javascript" src="script/jquery-1.7.2.js"></script>
<script type="text/javascript">
$(function() {
    // 基本初始状态
    $("li:gt(5):not(:last)").hide();

    // 给功能的按钮绑定单击事件
    $("div div a").click(function () {
        // 让某些品牌, 显示, 或隐藏
        $("li:gt(5):not(:last)").toggle();
        // 判断 品牌, 当前是否可见
        if( $("li:gt(5):not(:last)").is(":hidden") ){
            // 品牌隐藏的状态 : 1 显示全部品牌    == 角标向下 showmore
            $("div div a span").text("显示全部品牌");

            $("div div").removeClass();
            $("div div").addClass("showmore");

            // 去掉高亮
            $("li:contains('索尼')").removeClass("promoted");
        } else {
            // 品牌可见的状态: 2 显示精简品牌    == 角标向上 showless
            $("div div a span").text("显示精简品牌");

            $("div div").removeClass();
            $("div div").addClass("showless");

            // 加高亮
            $("li:contains('索尼')").addClass("promoted");
        }
    });
});

```

```

    }

    return false;
  });
});
</script>
</head>
<body>
  <div class="SubCategoryBox">
    <ul>
      <li><a href="#">佳能</a><i>(30440)</i></li>
      <li><a href="#">索尼</a><i>(27220)</i></li>
      <li><a href="#">三星</a><i>(20808)</i></li>
      <li><a href="#">尼康</a><i>(17821)</i></li>
      <li><a href="#">松下</a><i>(12289)</i></li>
      <li><a href="#">卡西欧</a><i>(8242)</i></li>
      <li><a href="#">富士</a><i>(14894)</i></li>
      <li><a href="#">柯达</a><i>(9520)</i></li>
      <li><a href="#">宾得</a><i>(2195)</i></li>
      <li><a href="#">理光</a><i>(4114)</i></li>
      <li><a href="#">奥林巴斯</a><i>(12205)</i></li>
      <li><a href="#">明基</a><i>(1466)</i></li>
      <li><a href="#">爱国者</a><i>(3091)</i></li>
      <li><a href="#">其它品牌相机</a><i>(7275)</i></li>
    </ul>
    <div class="showmore">
      <a href="more.html"><span>显示全部品牌</span></a>
    </div>
  </div>
</body>
</html>

```

8、jQuery 事件操作

`$(function(){})`;

和

`window.onload = function(){}`

的区别？

他们分别是在什么时候触发？

1、jQuery 的页面加载完成之后是浏览器的内核解析完页面的标签创建好 DOM 对象之后就会马上执行。

2、原生 js 的页面加载完成之后，除了要等浏览器内核解析完标签创建好 DOM 对象，还要等标签显示时需要的内容加载完成。

他们触发的顺序？

- 1、jQuery 页面加载完成之后先执行
- 2、原生 js 的页面加载完成之后

他们执行的次数？

- 1、原生 js 的页面加载完成之后，只会执行最后一次的赋值函数。
- 2、jQuery 的页面加载完成之后是全部把注册的 **function** 函数，依次顺序全部执行。

jQuery 中其他的事件处理方法：

click()	它可以绑定单击事件，以及触发单击事件
mouseover()	鼠标移入事件
mouseout()	鼠标移出事件
bind()	可以给元素一次性绑定一个或多个事件。
one()	使用上跟 bind 一样。但是 one 方法绑定的事件只会响应一次。
unbind()	跟 bind 方法相反的操作，解除事件的绑定
live()	也是用来绑定事件。它可以用来绑定选择器匹配的所有元素的事件。哪怕这个元素是后面动态创建出来的也有效

事件的冒泡

什么是事件的冒泡？

事件的冒泡是指，父子元素同时监听同一个事件。当触发子元素的事件的时候，同一个事件也被传递到了父元素的事件里去响应。

那么如何阻止事件冒泡呢？

在子元素事件函数体内，**return false;** 可以阻止事件的冒泡传递。

JavaScript 事件对象

事件对象，是封装有触发的事件信息的一个 javascript 对象。

我们重点关心的是怎么拿到这个 javascript 的事件对象。以及使用。

如何获取呢 javascript 事件对象呢？

在给元素绑定事件的时候，在事件的 **function(event)** 参数列表中添加一个参数，这个参数名，我们习惯取名为 **event**。这个 **event** 就是 javascript 传递参事件处理函数的事件对象。

比如：

//1.原生 javascript 获取 事件对象

```
window.onload = function () {  
    document.getElementById("areaDiv").onclick = function (event) {  
        console.log(event);  
    }  
}
```

//2.jQuery 代码获取 事件对象

```
$(function () {  
    $("#areaDiv").click(function (event) {  
        console.log(event);  
    });  
});
```

//3.使用 bind 同时对多个事件绑定同一个函数。怎么获取当前操作是什么事件。

```
$("#areaDiv").bind("mouseover mouseout",function (event) {  
    if (event.type == "mouseover") {  
        console.log("鼠标移入");  
    } else if (event.type == "mouseout") {  
        console.log("鼠标移出");  
    }  
});
```

☆练习 07 事件 图片跟随

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Insert title here</title>  
<style type="text/css">  
    body {  
        text-align: center;  
    }  
    #small {  
        margin-top: 150px;  
    }  
</style>
```



```
#showBig {
    position: absolute;
    display: none;
}
</style>
<script type="text/javascript" src="script/jquery-1.7.2.js"></script>
<script type="text/javascript">
    $(function(){
        $("#small").bind("mouseover mouseout mousemove",function (event) {
            if (event.type == "mouseover") {
                $("#showBig").show();
            } else if (event.type == "mousemove") {
                console.log(event);
                $("#showBig").offset({
                    left: event.pageX + 10,
                    top: event.pageY + 10
                });
            } else if (event.type == "mouseout") {
                $("#showBig").hide();
            }
        });
    });
</script>
</head>
<body>

    <div id="showBig">
        
    </div>

</body>
</html>
```

