

## 03-jQuery

讲师：王振国

### 今日任务

- 1.jQuery Hello world
- 2.jQuery 选择器
- 3.jQuery 过滤器
- 4.jQuery 元素筛选

### 课堂笔记

## 1、jQuery 介绍

什么是 jQuery ?

jQuery，顾名思义，也就是 JavaScript 和查询（Query），它就是辅助 JavaScript 开发的 js 类库。

**jQuery 核心思想!!!**

它的核心思想是 write less, do more(写得更少,做得更多)，所以它实现了很多浏览器的兼容问题。

**jQuery 流行程度**

jQuery 现在已经成为最流行的 JavaScript 库，在世界前 10000 个访问最多的网站中，有超过 55%在使用 jQuery。

**jQuery 好处!!!**

jQuery 是免费、开源的，jQuery 的语法设计可以使开发更加便捷，例如操作[文档](#)对象、选择[DOM](#)元素、制作动画效果、事件处理、使用[Ajax](#)以及其他功能

## 2、jQuery 的初体验!!!

需求：使用 jQuery 给一个按钮绑定单击事件？

```
<!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">
    // window.onload = function () {
    //     var btnObj = document.getElementById("btnId");
    //     // alert(btnObj);//[object HTMLButtonElement] ===>>> dom 对象
    //     btnObj.onclick = function () {
    //         alert("js 原生的单击事件");
    //     }
    // }

    $(function () { // 表示页面加载完成 之后, 相当 window.onload = function () {}
        var $btnObj = $("#btnId"); // 表示按 id 查询标签对象
        $btnObj.click(function () { // 绑定单击事件
            alert("jQuery 的单击事件");
        });
    });

  </script>
</head>
<body>

  <button id="btnId">SayHello</button>

</body>
</html>
```

常见问题？

1、使用 jQuery 一定要引入 jQuery 库吗？ 答案： 是，必须

2、jQuery 中的 \$ 到底是什么？ 答案： 它是一个函数

3、怎么为按钮添加点击响应函数的？ 答案：

- 1、使用 jQuery 查询到标签对象
- 2、使用标签对象.click( function(){} );

## 3、jQuery 核心函数

\$ 是 jQuery 的核心函数，能完成 jQuery 的很多功能。\$()就是调用\$这个函数

- 1、传入参数为 [ 函数 ] 时：  
表示页面加载完成之后。相当于 `window.onload = function(){}`
- 2、传入参数为 [ HTML 字符串 ] 时：  
会对我们创建这个 html 标签对象
- 3、传入参数为 [ 选择器字符串 ] 时：  
`$("#id 属性值");` id 选择器，根据 id 查询标签对象  
`$("标签名");` 标签名选择器，根据指定的标签名查询标签对象  
`$(".class 属性值");` 类型选择器，可以根据 class 属性查询标签对象
- 4、传入参数为 [ DOM 对象 ] 时：  
会把这个 dom 对象转换为 jQuery 对象

## 4、jQuery 对象和 dom 对象区分

### 4.1、什么是 jQuery 对象，什么是 dom 对象

#### Dom 对象

- 1.通过 `getElementById()` 查询出来的标签对象是 Dom 对象
- 2.通过 `getElementsByName()` 查询出来的标签对象是 Dom 对象
- 3.通过 `getElementsByTagName()` 查询出来的标签对象是 Dom 对象
- 4.通过 `createElement()` 方法创建的对象，是 Dom 对象

DOM 对象 Alert 出来的效果是： `[object HTML 标签名 Element]`

#### jQuery 对象

- 5.通过 JQuery 提供的 API 创建的对象，是 JQuery 对象
- 6.通过 JQuery 包装的 Dom 对象，也是 JQuery 对象
- 7.通过 JQuery 提供的 API 查询到的对象，是 JQuery 对象

jQuery 对象 Alert 出来的效果是： `[object Object]`

## 4.2、问题：jQuery 对象的本质是什么？

jQuery 对象是 dom 对象的数组 + jQuery 提供的一系列功能函数。

## 4.3、jQuery 对象和 Dom 对象使用区别

jQuery 对象不能使用 DOM 对象的属性和方法

DOM 对象也不能使用 jQuery 对象的属性和方法

## 4.4、Dom 对象和 jQuery 对象互转

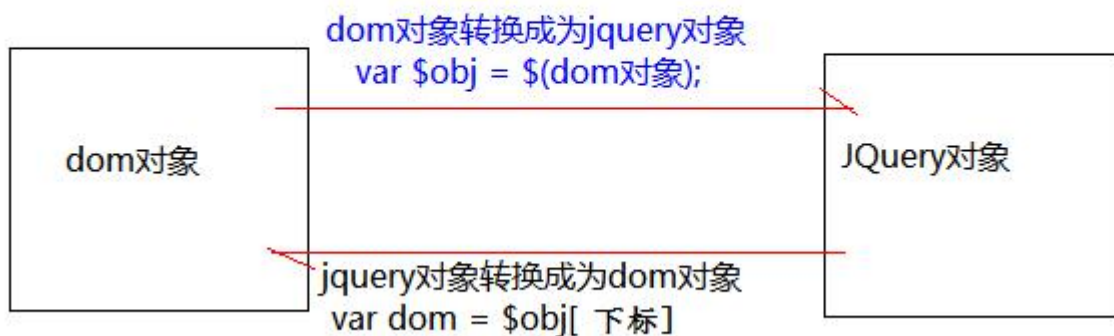
### 1、dom 对象转化为 jQuery 对象（\*重点）

- 1、先有 DOM 对象
- 2、\$( DOM 对象 ) 就可以转换成为 jQuery 对象

### 2、jQuery 对象转为 dom 对象（\*重点）

- 1、先有 jQuery 对象
- 2、jQuery 对象[下标]取出相应的 DOM 对象

### Jquery对象和Dom对象相互转换



## 5、jQuery 选择器（\*\*\*\*\*重点）

### 5.1、基本选择器（\*\*\*\*重点）



**#ID** 选择器：根据 id 查找标签对象

**.class** 选择器：根据 class 查找标签对象

**element** 选择器：根据标签名查找标签对象

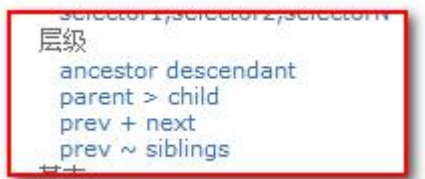
**\*** 选择器：表示任意的，所有的元素

**selector1, selector2** 组合选择器：合并选择器 1，选择器 2 的结果并返回

```
p.myClass
```

表示标签名必须是 p 标签，而且 class 类型还要是 myClass

### 5.2、层级选择器（\*\*\*\*重点）



**ancestor descendant** 后代选择器：在给定的祖先元素下匹配所有的后代元素

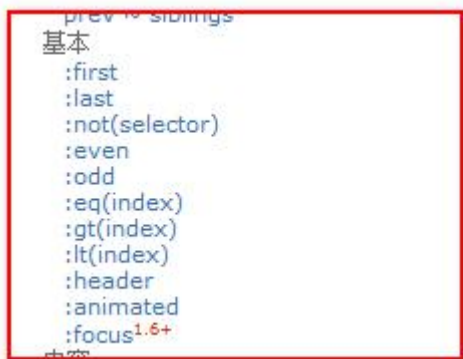
**parent > child** 子元素选择器：在给定的父元素下匹配所有的子元素

**prev + next** 相邻元素选择器：匹配所有紧接在 prev 元素后的 next 元素

**prev ~ siblings** 之后的兄弟元素选择器：匹配 prev 元素之后的所有 siblings 元素

## 5.3、过滤选择器

### 基本过滤器：



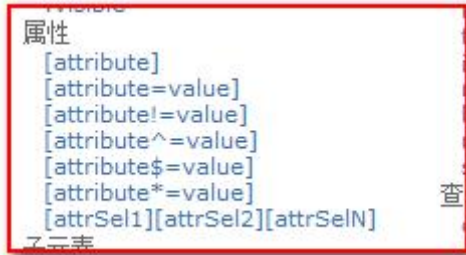
<b>:first</b>	获取第一个元素
<b>:last</b>	获取最后个元素
<b>:not(selector)</b>	去除所有与给定选择器匹配的元素
<b>:even</b>	匹配所有索引值为偶数的元素，从 0 开始计数
<b>:odd</b>	匹配所有索引值为奇数的元素，从 0 开始计数
<b>:eq(index)</b>	匹配一个给定索引值的元素
<b>:gt(index)</b>	匹配所有大于给定索引值的元素
<b>:lt(index)</b>	匹配所有小于给定索引值的元素
<b>:header</b>	匹配如 h1, h2, h3 之类的标题元素
<b>:animated</b>	匹配所有正在执行动画效果的元素

### 内容过滤器：



<b>:contains(text)</b>	匹配包含给定文本的元素
<b>:empty</b>	匹配所有不包含子元素或者文本的空元素
<b>:parent</b>	匹配含有子元素或者文本的元素
<b>:has(selector)</b>	匹配含有选择器所匹配的元素

## 属性过滤器:



**[attribute]**

匹配包含给定属性的元素。

**[attribute=value]**

匹配给定的属性是某个特定值的元素

**[attribute!=value]**

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。

**[attribute^=value]**

匹配给定的属性是以某些值开始的元素

**[attribute\$=value]**

匹配给定的属性是以某些值结尾的元素

**[attribute\*=value]**

匹配给定的属性是以包含某些值的元素

**[attrSel1][attrSel2][attrSelN]**

复合属性选择器，需要同时满足多个条件时使用。

## 表单过滤器:



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

**:text** 匹配所有 文本输入框

**:password** 匹配所有的密码输入框

**:radio** 匹配所有的单选框

**:checkbox** 匹配所有的复选框

**:submit** 匹配所有提交按钮

**:image** 匹配所有 img 标签

**:reset** 匹配所有重置按钮

**:button** 匹配所有 input type=button <button>按钮

**:file** 匹配所有 input type=file 文件上传

**:hidden** 匹配所有不可见元素 display:none 或 input type=hidden

## 表单对象属性过滤器：

表单对象属性  
:enabled  
:disabled  
:checked  
:selected

:enabled 匹配所有可用元素

:disabled 匹配所有不可用元素

:checked 匹配所有选中的单选，复选，和下拉列表中选中的 option 标签对象

:selected 匹配所有选中的 option

## 6、jQuery 元素筛选

### 筛选

#### 过滤

eq(index|-index)  
first()  
last()  
hasClass(class)  
filter(expr|obj|ele|fn)  
is(expr|obj|ele|fn)<sup>1.6\*</sup>  
map(callback)  
has(expr|ele)  
not(expr|ele|fn)  
slice(start,[end])

#### 查找

children([expr])  
closest(expr,[con]|obj|ele)<sup>1.6\*</sup>  
find(expr|obj|ele)  
next([expr])  
nextall([expr])  
nextUntil([exp|ele],[fil])<sup>1.6\*</sup>  
offsetParent()  
parent([expr])  
parents([expr])  
parentsUntil([exp|ele],[fil])<sup>1.6\*</sup>  
prev([expr])  
prevall([expr])  
prevUntil([exp|ele],[fil])<sup>1.6\*</sup>  
siblings([expr])

eq() 获取给定索引的元素

功能跟 :eq() 一样

first() 获取第一个元素

功能跟 :first 一样

last() 获取最后一个元素

功能跟 :last 一样

filter(exp) 留下匹配的元素

is(exp) 判断是否匹配给定的选择器，只要有一个匹配就返回，true

has(exp) 返回包含有匹配选择器的元素的元素

功能跟 :has 一样

not(exp) 删除匹配选择器的元素

功能跟 :not 一样

children(exp) 返回匹配给定选择器的子元素

功能跟 parent>child 一样

find(exp) 返回匹配给定选择器的后代元素

功能跟 ancestor descendant 一样



---

next()	返回当前元素的下一个兄弟元素	功能跟 <code>prev + next</code> 功能一样
nextAll()	返回当前元素后面所有的兄弟元素	功能跟 <code>prev ~ siblings</code> 功能一样
nextUntil()	返回当前元素到指定匹配的元素为止的后面元素	
parent()	返回父元素	
prev(exp)	返回当前元素的上一个兄弟元素	
prevAll()	返回当前元素前面所有的兄弟元素	
prevUnit(exp)	返回当前元素到指定匹配的元素为止的前面元素	
siblings(exp)	返回所有兄弟元素	
add()	把 <code>add</code> 匹配的选择器的元素添加到当前 <code>jquery</code> 对象中	



## 04-jQuery

讲师：王振国

### 今日任务

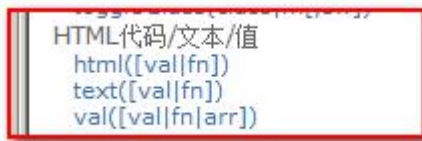
- 1.jQuery 的属性操作
- 2.jQuery 练习
- 3.jQuery CSS 样式操作
- 4.jQuery 动画操作
- 5.jQuery 事件操作

### 课堂笔记

### 今天课程内容：

## 1、jQuery 的属性操作

jQuery 属性操作



html()

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
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，只会执行最后一轮的赋值函数。
- 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>
```

通过event事件对象可以获取信息，包括鼠标 alt 等

