# 第四章 WEB04\_jQuery篇

**今日任务**

使用JQuery完成页面定时弹出广告(DOM转换和选择器)

使用JQuery完成表格的隔行换色

使用JQuery完成复选框的全选效果(jQuery的属性操作)

使用JQuery完成省市联动效果(jQuery的遍历和DOM操作)

使用JQuery完成下列列表左右选择(jQuery的事件)

使用JQuery完成表单的校验.

**教学导航**

|  |  |
| --- | --- |
| 教学目标 | 掌握JQuery的基本使用  掌握JQuery的基本选择器,层次选择器  会使用JQuery完成DOM的基本操作. |
| 教学方法 | 案例驱动法 |

## 一、使用JQ完成首页的定时广告弹出

### 1.需求分析

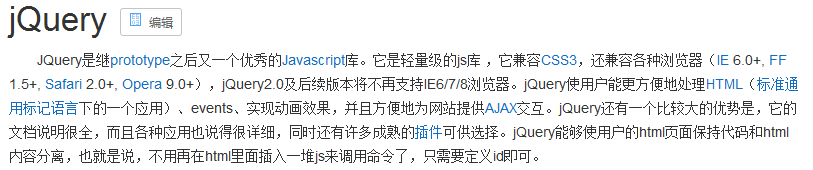
在首页中(logo的上方)显示一个广告图片(页面加载后，间隔3秒弹出广告图片，再间隔3秒后隐藏广告图片)。【使用jQuery实现】



### 2.技术分析

#### 2.1 jquery相关的知识

##### 什么是jquery



Jquery它是javascript的一个轻量级框架，对javascript进行封装，它提供了很多方便的选择器。供你快速定位到需要操作的元素上面去。还提供了很多便捷的方法。

##### 怎么使用jquery

Jquery它是一个库(框架)，要想使用它，必须先引入！

jquery-1.8.3.js：一般用于学习阶段。

jquery-1.8.3.min.js：用于项目使用阶段

##### Jquery的简单入门

所有的jquery代码写在页面加载函数

$(function(){

Jquery代码

});

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>jquery入门</title>  <script src="../../js/jquery-1.8.3.js"></script>  <script>  //整个文档加载完毕后执行  /\*function init(){  alert("张三");  }\*/    /\*window.onload= function(){  alert("张三");  }    //整个文档加载完毕后(包括图片)执行  window.onload= function(){  alert("老王");  }    //dom树绘制完毕后执行,可能DOM元素关联的东西并没有加载完  jQuery(document).ready(function(){  alert("李四");  });    //jquery的简写方法(页面加载)  $(function(){  alert("王五");  });  \*/    function init(){  document.getElementById("a1").onclick= function(){  location.href="";  }  }    $(function(){  document.getElementById("a1").onclick= function(){  location.href="";  }  });    $(function(){  document.getElementById("a2").onclick= function(){  location.href="";  }  });    </script>    </head>  <body onload="init()">  <a href="#" id="a1">ss</a>  <a href="#" id="a2"></a>  </body>  </html> |

##### 获取元素

JS:document.getElementById();

JQ:$(“#id”);

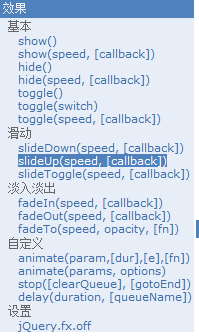
##### Jquery对象与DOM对象转换

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title></title>  <script type="text/javascript" src="../../js/jquery-1.8.3.js" ></script>  <script>  function JSWrite(){  //document.getElementById("span1").innerHTML="美美哒！";  var spanEle = document.getElementById("span1");  $(spanEle).html("美美哒！");  }    $(function(){  /\*document.getElementById("btn1").onclick = function(){  document.getElementById("span1").innerHTML="帅帅哒！";  }\*/  $("#btn1").click(function(){  //JQ对象转换成DOM对象的第一种方式  //$("#span1")[0].innerHTML="呵呵哒！";  //JQ对象转换成DOM对象的第二种方式  $("#span1").get(0).innerHTML="呵呵哒！";  });    });  </script>  </head>  <body>  <input type="button" value="JS写入" onclick="JSWrite()"/>  <input type="button" value="JQ写入" id="btn1"/><br /><br />  <span id="span1">sssss</span>  </body>  </html> |

注意：JQ对象只能操作JQ里面的属性和方法

JS对象只能操作JS里面的属性和方法。

##### Jquery的效果



### 3.实现步骤

第一步：引入jQuery相关的文件

第二步：书写页面加载函数

第三步：在页面加载函数中，获取显示广告图片的元素。

第四步：设置定时操作(显示广告图片的函数)

第五步：在显示广告图片的函数中，使用jQuery的方法让广告图片显示(show())

第六步：清除显示广告图片的定时操作

第七步：设置定时操作(隐藏广告图片的函数)

第八步：在隐藏广告图片的函数中，使用jQuery的方法让广告图片隐藏(hide())

第九步：清除隐藏广告图片的定时操作

### 4.代码实现

|  |
| --- |
| <script type="text/javascript">  var time;  $(function(){  time=setInterval("showAd()",3000);  });    function showAd(){  //$("#img1").show();  //$("#img1").slideDown(3000);  $("#img1").fadeIn(3000);  clearInterval(time);  time = setInterval("hideAd()",5000);  }    function hideAd(){  //$("#img1").hide();  //$("#img1").slideUp(3000);  $("#img1").slideUp(3000);  clearInterval(time);  }    </script> |

### 5.补充内容

Toggle的使用

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title></title>  <script type="text/javascript" src="../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  $("#btn1").click(function(){  $("#img1").toggle();  });  });    </script>  </head>  <body>  <input type="button" value="显示/隐藏" id="btn1"/><br />  <img src="../img/registImg.jpg" id="img1" />  </body>  </html> |

### 6.总结

#### 6.1 jquery的选择器

##### 基本选择器

id选择器：$(“#id名称”);

元素选择器：$(“元素名称”);

类选择器：$(“.类名”);

通配符：\*

多个选择器共用(并集)

**案例代码：**

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>基本选择器</title>  <link rel="stylesheet" href="../../css/style.css" type="text/css"/>  <script type="text/javascript" src="../../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  $("#btn1").click(function(){  $("#one").css("background-color","pink");  });  $("#btn2").click(function(){  $(".mini").css("background-color","pink");  });  $("#btn3").click(function(){  $("div").css("background-color","pink");  });  $("#btn4").click(function(){  $("\*").css("background-color","pink");  });  $("#btn5").click(function(){  $("#two .mini").css("background-color","pink");  });  });  </script>  </head>  <body>  <input type="button" id="btn1" value="选择为one的元素"/>  <input type="button" id="btn2" value="选择样式为mini的元素"/>  <input type="button" id="btn3" value="选择所有的div元素"/>  <input type="button" id="btn4" value="选择所有元素"/>  <input type="button" id="btn5" value="选择id为two并且样式为mini的元素"/>  <hr/>  <div id="one">  <div class="mini">  111  </div>  </div>    <div id="two">  <div class="mini">  222  </div>  <div class="mini">  333  </div>  </div>    <div id="three">  <div class="mini">  444  </div>  <div class="mini">  555  </div>  <div class="mini">  666  </div>  </div>    <span id="four">    </span>  </body>  </html> |

##### 层级选择器



ancestor descendant: 在给定的祖先元素下匹配所有的后代元素(儿子、孙子、重孙子)

parent > child : 在给定的父元素下匹配所有的子元素(儿子)

prev + next: 匹配所有紧接在 prev 元素后的 next 元素(紧挨着的，同桌)

prev ~ siblings: 匹配 prev 元素之后的所有 siblings 元素(兄弟)

代码演示：

|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>层级选择器</title>  <link rel="stylesheet" href="../../css/style.css" />  <script type="text/javascript" src="../../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  $("#btn1").click(function(){  $("body div").css("background-color","pink");  });  $("#btn2").click(function(){  $("body>div").css("background-color","pink");  });  $("#btn3").click(function(){  $("#two+div").css("background-color","pink");  });  $("#btn4").click(function(){  $("#one~div").css("background-color","pink");  });  });    </script>      </head>  <body>  <input type="button" id="btn1" value="选择body中的所有的div元素"/>  <input type="button" id="btn2" value="选择body中的第一级的孩子"/>  <input type="button" id="btn3" value="选择id为two的元素的下一个元素"/>  <input type="button" id="btn4" value="选择id为one的所有的兄弟元素"/>    <hr/>  <div id="one">  <div class="mini">  111  </div>  </div>    <div id="two">  <div class="mini">  222  </div>  <div class="mini">  333  </div>  </div>    <div id="three">  <div class="mini">  444  </div>  <div class="mini">  555  </div>  <div class="mini">  666  </div>  </div>    <span id="four">    </span>  </body>  </html> |

##### 基本过滤选择器

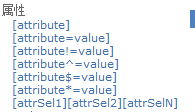
$('li').first() 等价于：$(“li:first”)



代码案例演示：

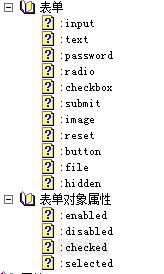
|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>基本过滤选择器</title>  <link rel="stylesheet" href="../../css/style.css" type="text/css"/>  <script type="text/javascript" src="../../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  $("#btn1").click(function(){  $("div:first").css("background-color","pink");  });  $("#btn2").click(function(){  $("div:last").css("background-color","pink");  });  $("#btn3").click(function(){  $("div:odd").css("background-color","pink");  });  $("#btn4").click(function(){  $("div:even").css("background-color","pink");  });  });  </script>    </head>  <body>  <input type="button" id="btn1" value="body中的第一个div元素"/>  <input type="button" id="btn2" value="body中的最后一个div元素"/>  <input type="button" id="btn3" value="选择body中的奇数的div"/>  <input type="button" id="btn4" value="选择body中的偶数的div"/>    <hr/>  <div id="one">  <div class="mini">  111  </div>  </div>    <div id="two">  <div class="mini">  222  </div>  <div class="mini">  333  </div>  </div>    <div id="three">  <div class="mini">  444  </div>  <div class="mini">  555  </div>  <div class="mini">  666  </div>  </div>    <span id="four">    </span>  </body>  </html> |

##### 属性选择器



|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>层级选择器</title>  <link rel="stylesheet" href="../../css/style.css" type="text/css"/>  <script type="text/javascript" src="../../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  $("#btn1").click(function(){  $("div[id]").css("background-color","pink");  });  $("#btn2").click(function(){  $("div[id='two']").css("background-color","pink");  });  });    </script>  </head>  <body>  <input type="button" id="btn1" value="选择有id属性的div"/>  <input type="button" id="btn2" value="选择有id属性的值为two的div"/>    <hr/>  <div id="one">  <div class="mini">  111  </div>  </div>    <div id="two">  <div class="mini">  222  </div>  <div class="mini">  333  </div>  </div>    <div id="three">  <div class="mini">  444  </div>  <div class="mini">  555  </div>  <div class="mini">  666  </div>  </div>    <span id="four">    </span>  </body>  </html> |

##### 表单选择器



|  |
| --- |
| <html>  <head>  <meta charset="UTF-8">  <title>表单选择器</title>  <link rel="stylesheet" type="text/css" href="../../css/style.css"/>  <script type="text/javascript" src="../../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  $("#btn1").click(function(){  $(":input").css("background-color","pink");  });  $("#btn2").click(function(){  $(":text").css("background-color","pink");  });  });  </script>  </head>  <body>  <input type="button" id="btn1" value="选择所有input元素" />  <input type="button" id="btn2" value="选择文本框" />  <br/>  <form>  <input type="text" /><br />  <input type="checkbox" /><br />  <input type="radio" /><br />  <input type="image" /><br />  <input type="file" /><br />  <input type="submit" />  <input type="reset" /><br />  <input type="password" /><br />  <input type="button" /><br />  <select><option/></select><br />  <textarea></textarea><br />  <button></button>  </form>  </body>  </html> |

## 二、使用JQ完成表格的隔行换色

### 1.需求分析

在页面加载的时候让显示用户的表格进行隔行换色显示，效果如下：使用JQ实现！



### 2.技术分析

需要使用jquery的选择器(基本选择器、基本过滤选择器)

还需要使用jquery的CSS的方法(css(name,value))

如果CSS样式已经由美工写好，此时可以使用jquery的CSS类操作



### 3.步骤分析

第一步：引入jquery的类库

第二步：直接写页面加载函数

第三步：直接使用jquery的选择器(组合选择)拿到需要操作的元素(奇数行和偶数行)

第四步：分别使用CSS的方法(css(name,value))对奇数行和偶数行设置背景颜色。

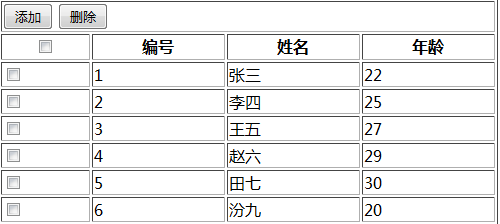
### 4.代码实现

|  |
| --- |
| <script type="text/javascript" src="../js/jquery-1.8.3.js" ></script>  <link rel="stylesheet" type="text/css" href="../css/style.css"/>  <script>  $(function(){  $("tbody tr:even").addClass("even");  $("tbody tr:odd").addClass("odd");  });  </script> |

## 三、使用JQ完成全选和全不选

### 1.需求分析

在系统后台进行人员管理时，进行批量删除，使用jq完成全选和全不选



### 2.技术分析

需要使用jquery的选择器(id选择器、类选择器)

需要使用jquery的属性操作方法 prop()

### 3.步骤分析

第一步：引入jquery文件

第二步：书写页面加载函数

第三步：为上面的复选框绑定单击事件

第四步：将下面所有的复选框的选中状态设置成跟上面的一致！

### 4.代码实现

|  |
| --- |
| <script type="text/javascript" src="../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  //找到下面所有的复选框并设置属性checked()  /\*if($("#select")[0].checked==true){  $(".selectOne").attr("checked",true)  }\*/  $("#select").click(function(){  $(".selectOne").attr("checked",this.checked);  });  });  </script> |

注意：attr在jquery1.11版本不适用，采用prop()来替代(在各个版本都适用)。

在jquery中如何调用方法？

元素.方法（）

## 四、使用JQ完成省市二级联动

### 1.需求分析



使用jquery完成省市二级联动

### 2.技术分析

#### 2.1数组的遍历操作

方式一：



|  |
| --- |
| $(function(){  // 全选/ 全不选  $("#checkallbox").click(function(){  var isChecked = this.checked;  //使用对象访问的方式进行遍历，语法：$().each(function(){})  $("input[name='hobby']").each(function(){  this.checked = isChecked;  });  });  }); |

方式二：



|  |
| --- |
| $.each( [0,1,2], function(i, n){  alert( "Item #" + i + ": " + n );  }); |

#### 2.2 文档处理操作

追加内容

apend: A.append(B) 将B追加到A的内容的末尾处

appendTo: A.appendTo(B) 将A加到B内容的末尾处

### 3.步骤分析

第一步：确定事件(change事件),在绑定的函数里面获取用户选择的省份

第二步：创建二维数组来存储省份和城市

第三步：遍历二维数组中的省份(与用户选择的省份进行比较)【使用JQ的遍历操作】

第四步：接着遍历数组中的城市

第五步：创建一个城市文本节点

第六步：创建option元素节点

第七步：将文本节点添加到元素节点中【使用JQ的文档操作方法】

第八步：获取第二个下拉列表并将option元素节点添加进去

第九步：清除第二个下拉列表的option内容

### 4.代码实现

|  |
| --- |
| <script type="text/javascript" src="../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  //2.创建二维数组用于存储省份和城市  var cities = new Array(3);  cities[0] = new Array("武汉市","黄冈市","襄阳市","荆州市");  cities[1] = new Array("长沙市","郴州市","株洲市","岳阳市");  cities[2] = new Array("石家庄市","邯郸市","廊坊市","保定市");  cities[3] = new Array("郑州市","洛阳市","开封市","安阳市");    $("#province").change(function(){  //10.清除第二个下拉列表的内容  $("#city").empty();    //1.获取用户选择的省份  var val = this.value;  //alert(val);  //3.遍历二维数组中的省份  $.each(cities,function(i,n){  //alert(i+":"+n);  //4.判断用户选择的省份和遍历的省份  if(val==i){  //5.遍历该省份下的所有城市  $.each(cities[i], function(j,m) {  //alert(m);  //6.创建城市文本节点  var textNode = document.createTextNode(m);  //7.创建option元素节点  var opEle = document.createElement("option");  //8.将城市文本节点添加到option元素节点中去  $(opEle).append(textNode);  //9.将option元素节点追加到第二个下拉列表中去  $(opEle).appendTo($("#city"));  });  }  });    });  });  </script> |

## 五、使用JQ完成下拉列表左右选择

### 1.需求分析

见图片文字部分内容。



### 2.分析

第一步：确定事件(鼠标单击事件click)

第二步：获取左侧下拉列表被选中的option($(“#left option:selected”)) [假设左侧select定义了一个id=left]

第三步：将获取到的option添加到右侧的下拉列表中去。(append方法的使用)

### 3.代码实现

|  |
| --- |
| <script type="text/javascript" src="../js/jquery-1.8.3.js" ></script>  <script>  $(function(){  /\*1.选中单击去右边\*/  $("#selectOneToRight").click(function(){  $("#left option:selected").appendTo($("#right"));  });    /\*2.单击全部去右边\*/  $("#selectAllToRight").click(function(){  $("#left option").appendTo($("#right"));  });    /\*3.选中双击去右边\*/  $("#left option").dblclick(function(){  $("#left option:selected").appendTo($("#right"));  });  });  </script> |

## 六、使用JQ完成表单校验

### 1.需求分析

### 2.技术分析

这里使用validation插件完成对表单数据的校验