

# day34-jquery3基础

## 学习目标：

1. 能够引入jQuery
2. 能够使用jQuery的基本选择器
3. 能够使用jQuery的层级选择器
4. 能够使用jQuery的属性选择器
5. 能够使用jQuery的基本过滤选择器
6. 能够使用jQuery的完成案例-广告的显示与隐藏
7. 能够完成隔行换色案例
8. 能够完成全选全不选案例

## 案例一:使用jQuery完成页面定时弹出广告

### 一,需求分析



- 进入页面3s后弹出广告,3s后广告隐藏

### 二,技术分析

#### 1.JQ概述

##### 1.1 jQuery的概述

jQuery是一个优秀的javascript的轻量级框架之一，兼容css3和各大浏览器，提供了dom、events、animate、ajax 等简易的操作。并且jquery的插件非常丰富，大多数功能都有相应的插件解决方案。jquery的宗旨是write less do more。

说白了:JQ就是js库,封装了JS常见的操作,我们使用起来更加的简单

## 1.2 jQuery的作用

jQuery最主要的作用是简化js的Dom树的操作

## 1.3 jQuery框架的下载

jQuery的官方下载地址：<http://www.jquery.com>

## 1.4 jQuery的版本

- 1.x：兼容IE678，使用最为广泛的，官方只做BUG维护，功能不再新增。因此一般项目来说，使用1.x版本就可以了，最终版本：1.12.4 (2016年5月20日)
- 2.x：不兼容IE678，很少有人使用，官方只做BUG维护，功能不再新增。如果不考虑兼容低版本的浏览器可以使用2.x，最终版本：2.2.4 (2016年5月20日)
- 3.x：不兼容IE678，只支持最新的浏览器。除非特殊要求，一般不会使用3.x版本的，很多老的jQuery插件不支持这个版本。目前该版本是官方主要更新维护的版本
- 注:开发版本与生产版本，命名为jQuery-x.x.x.js为开发版本，命名为jQuery-x.x.x.min.js为生产版本，开发版本源码格式良好，有代码缩进和代码注释，方便开发人员查看源码，但体积稍大。而生产版本没有代码缩进和注释，且去掉了换行和空行，不方便发人员查看源码，但体积很小

## 2.jq入门

- 把jq库导入到我们项目下，在html页面使用 `<script>` 导入。

```
<script type="text/javascript" src="../js/jquery-3.3.1.min.js"></script>
```

- 代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script type="text/javascript" src="../js/jquery-3.3.1.min.js"></script>
  <script>
    $(document).ready(function (){
      alert("页面加载完毕");
    });
  </script>
</head>
<body>
HelloJquery
</body>
</html>
```

注: 没有导入jQuery库会出现如下bug:

```
Uncaught ReferenceError: $ is not defined
    at 01%E7%82%B9%E5%87%BB%E4%BA%8B%E4%BB%B6.html:20
```

### 3.jQuery对象和JS对象转换

jQuery本质上虽然也是js，但如果使用jQuery的属性和方法那么必须保证对象是jQuery对象而不是js方式获得的DOM对象。使用js方式获取的对象是js的DOM对象，使用jQuery方式获取的对象是jQuery对象。两者的转换关系如下：

- js的DOM对象转换成jQuery对象，语法：\$(js对象)
- jQuery对象转换成js对象，语法：jquery对象[索引] 或 jquery对象.get(索引); 一般索引写0

### 4.jq中事件的使用

事件

```
blur([[data],fn))
change([[data],fn))
click([[data],fn))
dblclick([[data],fn))
error([[data],fn))
focus([[data],fn))
focusin([[data],fn))
focusout([[data],fn))
keydown([[data],fn))
keypress([[data],fn))
keyup([[data],fn))
mousedown([[data],fn))
mouseenter([[data],fn))
mouseleave([[data],fn))
mousemove([[data],fn))
mouseout([[data],fn))
mouseover([[data],fn))
mouseup([[data],fn))
resize([[data],fn))
scroll([[data],fn))
select([[data],fn))
submit([[data],fn))
unload([[data],fn))
```

#### 4.1 基本事件的使用

- 事件在jq里面都封装成了方法. 去掉了JS里面on. 语法:

```
jq对象.事件方法名(function(){}); eg: 点击: jq对象.click(function(){});
```

- 点击事件

```

<body>
<span id="nid"></span><br/>
<input id="buttona" type="button" value="确认"/>
</body>
<script>
    //获取按钮对象的点击事件
    $("#buttona").click(function(){
        //获取span对象后转换成js对象
        $("#nid")[0].innerHTML="hello";
    });
</script>

```

- 获得焦点和失去焦点

```

<body>
<center>
<div>
    <input type="text" id="textid"/>
</div>
</center>
</body>
<script>
/*$("#textid").focus(function () {
    document.write("获取焦点");
})*
$("#textid").blur(function () {
    document.write("失去焦点");
})
</script>

```

- 内容改变事件

```

<body>
    <select id="starSelectId">
        <option value="Jordan">乔丹</option>
        <option value="James">詹姆斯</option>
        <option value="Kobe">科比</option>
        <option value="Iverson">艾弗森</option>
    </select>
</body>

<script>
    //jq对象.change(function(){} ) 内容改变事件
    $("#starSelectId").change(function () {
        console.log("内容改变了..." + this.value);
    });
</script>

```

- 鼠标相关的事件

```

<body>
  <div id="divId"></div>
</body>

<script>
  // jq对象.mouseenter() 鼠标进入
  $("#divId").mouseenter(function () {
    //this.style.backgroundColor = "red";
    $(this).css("backgroundColor", "red"); //把js对象转成了jq对象
  });

  // jq对象.mouseout() 鼠标离开
  $("#divId").mouseout(function () {
    //this.style.backgroundColor = "blue";
    $(this).css("backgroundColor", "blue");
  });

</script>

```

- 键盘相关事件

```

<body>
  <input id="inputId" type="text"/>
</body>

<script>

  // jq对象.keydown() 键盘按下
  $("#inputId").keydown(function () {
    console.log("键盘按下...");
  });

  // jq对象.keyup() 键盘抬起
  $("#inputId").keyup(function () {
    console.log("键盘抬起...");
  });

</script>

```

## 4.2jQuery的事件绑定与解绑

- 事件的绑定

jQuery元素对象.on(事件的类型, function(){} );  
 其中：事件名称是jQuery的事件方法的方法名称，例如：click、mouseover、mouseout、focus、blur等

- 事件的解绑

jQuery元素对象.off(事件名称);

其中：参数事件名称如果省略不写，可以解绑该jQuery对象上的所有事件

- 实例代码

```
<body>
<center>
    <input type="button" id="newid" value="点击"><br/><br/><br/>
    <input type="button" id="oldid" value="解绑">
</center>
</body>
<script>
$("#newid").click(function () {
    alert("点击成功");
});
$("#oldid").on("click",function () {
    $("#newid").off("click");
});
</script>
```

#### 4.3事件切换

- 链式写法

```
<style>
    #divId{
        border: 1px solid;
        margin-top: 100px;
        height: 100px;
        width: 300px;
        background-color: gray;
    }
</style>
</head>
<body>
<center>
<div id="divId"></div>
</center>
</body>
<script>
    // jq对象.mouseenter() 鼠标进入
    $("#divId").mouseenter(function () {
        //this.style.backgroundColor = "red";
        $(this).css("backgroundColor", "red");    //把js对象转成了jq对象
    }).mouseout(function () {
        //this.style.backgroundColor = "blue";
        $(this).css("backgroundColor", "gray");
    });
</script>
```

# 5.jq动画

## 5.1基本效果

- 方法

方法名称	解释
show([speed],[easing],[fn])	显示元素方法
hide([speed],[easing],[fn])	隐藏元素方法
toggle([speed],[easing],[fn])	切换元素方法，显示的使之隐藏，隐藏的使之显示

- 参数

参数名称	解释
speed	三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如：1000)
easing	用来指定切换效果，默认是"swing"，可用参数"linear"
fn	在动画完成时执行的函数，每个元素执行一次

- 实例

```
<input type="button" value="show()" id="btn1" />
<input type="button" value="hide()" id="btn2" />
<input type="button" value="toggle()" id="btn3" />

<br />
<div id="div1" style="width: 100px; height: 100px; border: 1px solid red;"></div>

<script>
    $("#btn1").click(function() {
        $("#div1").show(1000);

    });

    $("#btn2").click(function() {
        $("#div1").hide(1000,function(){
            alert("Bye...");
        });
    });

    $("#btn3").click(function() {
        $("#div1").toggle();

    });
</script>
```

```
</script>
```

## 5.2滑动效果

- 方法

方法名称	解释
slideDown([speed],[easing],[fn]])	向下滑动方法
slideUp([speed],[easing],[fn]])	向上滑动方法
slideToggle([speed],[easing],[fn]])	切换元素方法，显示的使之隐藏，隐藏的使之显示

- 参数

参数名称	解释
speed	三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如：1000)
easing	用来指定切换效果，默认是"swing"，可用参数"linear"
fn	在动画完成时执行的函数，每个元素执行一次

## 5.3 淡入淡出效果

- 方法

方法名称	解释
fadeIn([speed],[easing],[fn]])	淡入显示方法
fadeOut([speed],[easing],[fn]])	淡出隐藏方法
fadeToggle([speed],[easing],[fn]])	切换元素方法，显示的使之隐藏，隐藏的使之显示

- 参数

参数名称	解释
speed	三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如：1000)
easing	用来指定切换效果，默认是"swing"，可用参数"linear"
fn	在动画完成时执行的函数，每个元素执行一次



### 三,思路分析

- 创建这个页面,在logo上面添加一个广告区域,并且隐藏
- 导入JQ库
- 创建定时任务,显示广告

```
setTimeout("showAd()",3000);
```

- 创建showAd()函数

```
function showAd(){
    //1.获得广告div对象
    //2.调用show()方法展示
    //3.再创建定时任务,3s之后隐藏 setTimeout("hideAd()",3000);
}
```

- 创建hideAd()函数

```
function hideAd(){
    //1.获得广告div对象
    //2.调用hide()方法隐藏
}
```

### 四,代码实现

```
<style>
    #guang{
        width: 98%;
        display: none;
    }
</style>

<tr>
    <td width="100%" height="100px">
        
    </td>
</tr>

<script>
    setTimeout("showg()",3000);
    function showg() {
        $("#guang").show();
        setTimeout("hidn()",3000);
    }
    function hidn() {
        $("#guang").hide();
    }
</script>
```

# 五,选择器总结

选择器的作用: 得到标签对象, 获得就是JQ对象

## 1.基本选择器【重点】

- 语法

选择器名称	语法	解释
标签选择器（元素选择器）	<code>\$("html标签名")</code>	获得所有匹配标签名称的于元素
id选择器	<code>\$("#id的属性值")</code>	获得与指定id属性值匹配的元素
类选择器	<code>\$(".class的属性值")</code>	获得与指定的class属性值匹配的元素

## 2 层级选择器

- 语法

选择器名称	语法	解释
后代选择器	<code>\$("A B")</code>	选择A元素内部的所有B元素
子选择器	<code>\$("A &gt; B")</code>	选择A元素内部的所有B子元素
兄弟选择器	<code>\$("A + B")</code>	获得A元素同级的下一个B元素
兄弟选择器	<code>\$("A ~ B")</code>	获得A元素同级的所有B元素

## 3 属性选择器

- 语法

选择器名称	语法	解释
属性选择器	<code>\$("A[属性名]")</code>	包含指定属性的选择器
属性选择器	<code>\$("A[属性名=值]")</code>	包含指定属性等于指定值的选择器



## 4.基本过滤选择器

- 语法

选择器名称	语法	解释
首元素选择器	:first	获得选择的元素中的第一个元素
尾元素选择器	:last	获得选择的元素中的最后一个元素
非元素选择器	:not(selector)	不包括指定内容的元素
偶数选择器	:even	偶数，从 0 开始计数
奇数选择器	:odd	奇数，从 0 开始计数
等于索引选择器	:eq(index)	指定索引元素
大于索引选择器	:gt(index)	大于指定索引元素
小于索引选择器	:lt(index)	小于指定索引元素

- 用法

```
$("#bt1").click(function () {  
    $("div:first").css("backgroundColor", "yellow")  
});
```

## 5.表单属性选择器

- 语法

选择器名称	语法	解释
可用元素选择器	:enabled	获得可用元素
不可用元素选择器	:disabled	获得不可用元素
选中选择器	:checked	获得单选/复选框选中的元素
选中选择器	:selected	获得下拉框选中的元素

## 案例二使用jQuery完成表格的隔行换色

### 一,需求分析

<input type="checkbox"/>	商品名称	商品价格	商品数量	操作
<input type="checkbox"/>	MAC	18000	10	删除 修改
<input type="checkbox"/>	MAC	18000	10	删除 修改
<input type="checkbox"/>	MAC	18000	10	删除 修改
<input type="checkbox"/>	MAC	18000	10	删除 修改

### 二,技术分析

#### 1.jQuery操作CSS样式

API方法	解释
css(name)	获取CSS样式
css(name,value)	设置CSS样式

```
<body>
  <div id="divId" style="width: 100px;height: 100px;background-color: red"></div>
  <input type="button" value="获得背景色的样式值" id="btn01"/>
  <input type="button" value="设置背景色的样式值为green" id="btn02"/>

</body>
<script>
  $("#btn01").click(function () {
    //获得div里面的CSS样式的值
    var colorValue = $("#divId").css("background-color");
    alert("colorValue="+colorValue);
  });

  $("#btn02").click(function () {
```

```
//设置背景色的样式值为green
$("#divId").css("background-color", "green");
});

</script>
```

## 2.jQuery操作标签的类名

API方法	解释
addClass(value)	给指定的对象添加类名
removeClass(value)	给指定的对象删除类名
toggleClass(value)	如果没类名，则添加，如果有类名，则删除

```
<body>
<span id="span1"></span>
<input id="butt1" type="button" value="确定"/>
</body>
<script>
    $("#butt1").click(function () {
        $("#span1").addClass("newClass");
    });
</script>
```

## 三,思路分析

- odd为奇数 even为偶数
- 筛选出奇数的tr, 设置一个背景色
- 筛选出偶数的tr, 设置一个背景色

## 四,代码实现

```
<script>
$("#tr:odd").css("backgroundColor", "gray");
$("#tr:even").css("backgroundColor", "yellow");
</script>
```

## 案例三使用jQuery完成复选框的全选效果

### 一,需求分析

<input checked="" type="checkbox"/>	商品名称	商品价格	商品数量	操作
<input checked="" type="checkbox"/>	Mac	18000	10	<a href="#">删除</a> <a href="#">修改</a>
<input checked="" type="checkbox"/>	Mac	18000	10	<a href="#">删除</a> <a href="#">修改</a>
<input checked="" type="checkbox"/>	Mac	18000	10	<a href="#">删除</a> <a href="#">修改</a>
<input checked="" type="checkbox"/>	Mac	18000	10	<a href="#">删除</a> <a href="#">修改</a>

## 二,技术分析

### 1.jq操作标签属性【重点】

- 语法

API方法	解释
attr(name[,value])	获得/设置属性的值
prop(name[,value])	获得/设置属性的值(checked , selected)

- attr与prop的注意问题  
attr与prop是以1.6为界限  
checked 和 selected 使用prop获取  
其他使用attr获取 获取不到换成prop

## 三.思路分析

- 创建页面
- 给最上面的复选框设置点击事件,创建一个匿名函数响应这个事件
- 在函数体里,获得下面4个复选框对象集合(.类名),把它们的checked属性设置和最上面的一致

## 四,代码实现

- js代码

```
<script>
    //给最上面的复选框设置点击事件,创建一个匿名函数响应这个事件
    $("#all").click(function() {
        //根据最上面的复选框是否选中而选中,this是最上面复选框对象(并且是dom(js)对象)
        $(".one").attr("checked",this.checked);
    });
</script>
```

