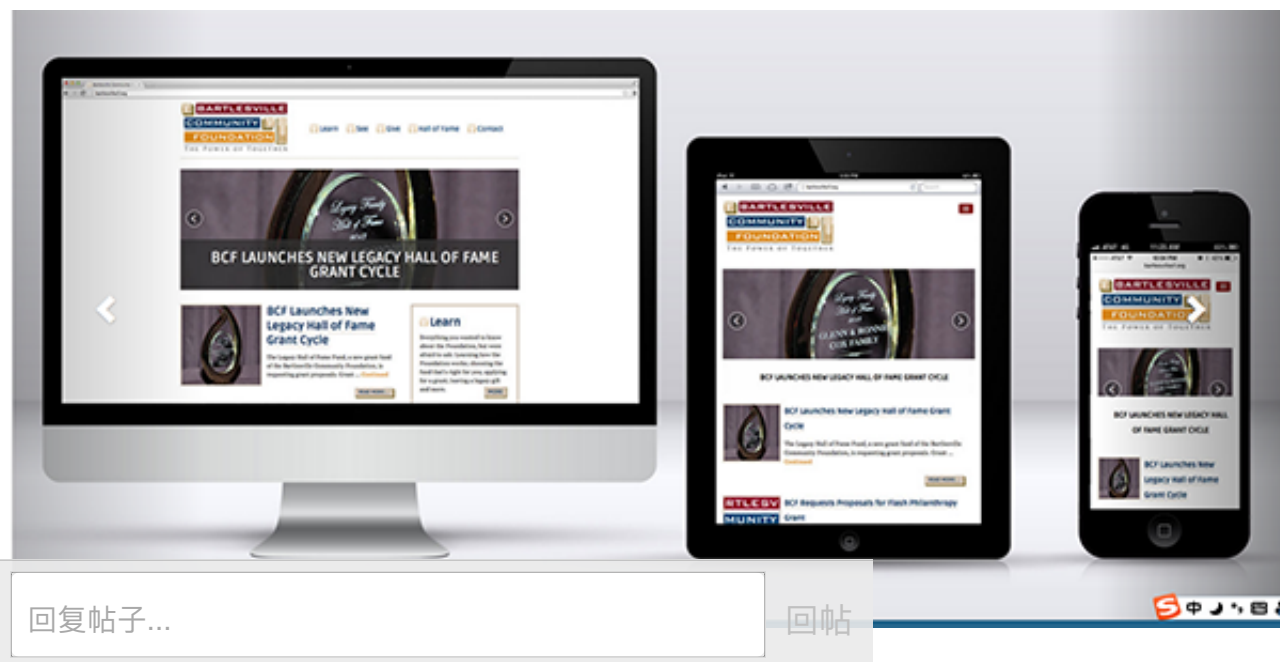


**【济南中心】JavaEE就业班同步笔记第一阶段：JavaWeb之JavaScr...**

小鲁哥哥 • 2016-11-24 21:37:45

**点击查看济南黑马校区最新开班计划****【济南中心】JavaEE就业班同步笔记第一阶段：  
JavaWeb之前端技术--JavaScript高级****1 案例一：使用JS完成图片的轮播的效果：****1.1 需求：****1.2 分析：****1.2.1 技术分析：****【JS中的Window对象的定时的操作】**

\* 设置定时的方法

SetInterval()指定周期内调用函数

setTimeout()指定毫秒后调用函数

\* 清除定时的方法

clearInterval()

clearTimeout()

1.2.2 步骤分析:

【步骤一】 创建一个HTML文件

【步骤二】 当页面加载的时候开始计时.使用onload事件.

【步骤三】 编写onload事件触发的函数.

【步骤四】 获得操作图片的控制权.

【步骤五】 修改图片的src的属性.

1.3 代码实现:

```
[mw_shl_code=javascript,true]<script>
    window.onload = function(){
        // 设置定时:
        window.setInterval("changeImg()",5000);
    }
    var i = 1;
    function changeImg(){
        i++;
        // 获得图片的控制权:
        if(i > 3){
            i=1;
        }
        var img1 = document.getElementById("img1");
        img1.src="../img/"+i+".jpg";
    }
</script>[/mw_shl_code]
```

## 2 案例二: 使用JS实现定时弹出广告定时隐藏.

### 2.1 需求:

在网站的首页上定时弹出一个广告, 并且定时隐藏掉.



## 2.2 分析：

### 2.2.1 技术分析：

#### 【JS的window对象的定时的方法】

##### \* 设置定时的方法

Setinterval()指定周期内调用函数

setTimeout()指定毫秒后调用函数

##### \* 清除定时的方法

clearInterval()

clearTimeout()

#### 【CSS的显示和隐藏的属性】

##### \* display

\* block：显示元素：

\* none：隐藏元素：

### 2.2.2 步骤分析：

【步骤一】创建一个HTML页面

【步骤二】确定事件：页面的加载事件

【步骤三】触发一个函数，编写该函数.

【步骤四】在函数中设置定时操作.定时执行一个显示的函数.

【步骤五】清空定时，重新设置定时，5秒钟隐藏.

## 2.3 代码实现：

```
[mw_shl_code=javascript,true]var time;
window.onload = function(){
    time = window.setInterval("show()",5000);
}
```

```

// 显示广告的方法
function show(){
    // 获得广告的div元素:
    var adDiv = document.getElementById("adDiv");
    adDiv.style.display = "block";
    window.clearInterval(time);
    time = window.setInterval("hide()",5000);
}

// 隐藏广告的方法:
function hide(){
    // 获得广告的div元素:
    var adDiv = document.getElementById("adDiv");
    adDiv.style.display="none";
    window.clearInterval(time);
}[/mw_shl_code]

```

## 2.4 总结:

### 2.4.1 JS中的BOM对象:

Window  
 Navigator  
 History  
 Location

#### Window对象的方法

- \* alert();            --弹出对话框
- \* setInterval(); --按照指定周期内调用函数
- \* setTimeout(); --在指定毫秒数后调用函数
- \* clearInterval(); --取消setInterval指定的周期
- \* clearTimeout(); --取消clearTimeout指定的毫秒数
- \* confirm();        --弹出一个确认窗口
- \* prompt();          --弹出一个可输入的对话框
- \* open();            --打开一个新窗
- o o o

Navigator：包含的是浏览器的信息。

- \* appName() --返回浏览器名称
- \* userAgent() --返回客户端发送给服务器的头部值
- o o o

Screen:用来获得屏幕信息:

- \* availHeight() --屏幕高度（不含win任务栏）
- \* availWidth() --屏幕宽度（不含win任务栏）
- \* height() --屏幕高度
- \* width() --屏幕宽度

History:浏览器的历史对象:

- \* back() --history的前一个url
- \* forward() --history的后一个url
- \* go() --history的某个url

Location:包含URL信息的对象

- \* host() --设置或返回主机名或当前url端口号
- \* href() --设置或返回完整url
- \* protocol() --设置或返回当前url协议

### 3 使用JS完成注册页面表单提示及校验

#### 3.1 需求:

用户名

密码

确认密码

性别 ☐男 ☐女

籍贯

爱好 ☐篮球 ☐足球 ☐排球 ☐羽毛球

邮箱

注册页面之前是弹出对话框的形式进行校验的这种方式不是特别友好! 可以将错误信息显示到文本框的后面.而且当光标落入到文本框的时候,提示的信息.

### 3.2 分析:

#### 3.2.1 技术分析:

##### 【JS的输出】

\* document.getElementById("").innerHTML="HTML的代码";

\* document.write("");

##### 【JS的事件】

\* onfocus :获得焦点.

\* onblur :失去焦点.

\* onsubmit :提交的时候.

#### 3.2.2 步骤分析:

【步骤一】创建一个html文档

【步骤二】在要去校验的文本框上添加事件.

【步骤三】触发函数

【步骤四】在函数中向文本框后的html的区域中写入一段提示的内容.

### 3.3 代码实现:

```
[mw_shl_code=javascript,true]function tips(id,content){
```

```

        document.getElementById(id+"Span").innerHTML = "<font color='red'>" + content + "</font>";
    }

    function checkForm(){
        // 判断用户名不能为空:
        var username = document.getElementById("username").value;
        if(username == ""){
            document.getElementById("usernameSpan").innerHTML = "<font color='red'>用户名不能为空! </font>";
            return false;
        }

        var password = document.getElementById("password").value;
        if(password == ""){
            document.getElementById("passwordSpan").innerHTML = "<font color='red'>密码不能为空! </font>";
            return false;
        }
    }
}[/mw_shl_code]

```

### 3.4 总结:

#### 3.4.1 JS的事件的总结:

- \* onload :加载
- \* onclick :单击
- \* onsubmit :提交
- \* onfocus :获取焦点
- \* onblur :失去焦点
- \* onchange :下拉列表改变事件.
- \* ondblclick:双击某个元素的事件.

键盘操作事件:

- \* onkeydown :键盘按下
- \* onkeyup :键盘抬起
- \* onkeypress:键盘按下并释放

鼠标操作事件:

- \* onmousemove:
- \* onmouseout:
- \* onmouseover:
- \* onmousedown
- \* onmouseup

#### 4 案例四：使用JS完成后台数据展示的隔行换色的表格：

##### 4.1 需求：

在后台展示数据的页面上，通常使用表格标签进行数据的展示.没有背景的表格比较难看的，可以使用JS控制表格的隔行换色.

分类的ID	分类的名称	分类的描述	操作
1	手机数码	手机数码	修改 删除
2	电脑办公	电脑办公	修改 删除
3	烟酒糖茶	烟酒糖茶	修改 删除
4	鞋靴箱包	鞋靴箱包	修改 删除
5	汽车用品	汽车用品	修改 删除

##### 4.2 分析：

##### 4.2.1 技术分析：

【使用JS获得表格的行数】

\* 获得到控制的表格元素：

```
* var tab1 = document.getElementById("tab1");
```

```
* var len = tab1.rows.length;
```

##### 4.2.2 步骤分析：

【步骤一】 创建一个HTML页面：可以使用后台设计页面.

【步骤二】 确定事件： onload事件.

【步骤三】 触发一个函数.在这个函数中获得操作的表格.

【步骤四】 获得表格中的所有的行数.

【步骤五】 遍历表格的行数.

【步骤六】 判断是否是奇数行还是偶数行.

##### 4.3 代码实现：

```
[mw_shl_code=javascript,true]function changeColor(){
    // 获得要操作的对象的控制权：
    var tab1 = document.getElementById("tab1");
```



```

// 获得表格的所有的行数:
var count = tab1.rows.length;
// 遍历每行:
for(var i = 0;i<count;i++){
    if(i % 2 == 0){
        // 偶数行
        tab1.rows.style.backgroundColor = "#00FF00";
    }else{
        // 奇数行
        tab1.rows.style.backgroundColor = "#00FFFF";
    }
}
}

```

[/mw\_shl\_code]4.4 总结:

#### 4.4.1 表格中的tbody和thead标签

```

[mw_shl_code=javascript,true]function changeColor(){
    // 获得操作的表格的控制权:
    var tab1 = document.getElementById("tab1");
    // 获得tbody中的所有的行.
    var len = tab1.tBodies[0].rows.length;
    for(var i = 0;i< len ;i++){
        if(i % 2 == 0){
            tab1.tBodies[0].rows.style.backgroundColor = "green";
        }else{
            tab1.tBodies[0].rows.style.backgroundColor = "gold";
        }
    }
}
[/mw_shl_code]

```

### 5 案例五：使用JS完成复选框的全选和全不选的效果

#### 5.1 需求:

在实际的开发中一条记录一条记录进行删除的话，效率很低，有的时候需要一起删除多条记录.需要通过在表格之前设置一个复选框的形式进行勾选复选框.点击一个删除的按钮.

<div> <input type="checkbox"/> 添加           <input type="checkbox"/> 删除         </div>				
<input checked="" type="checkbox"/>	分类的ID	分类的名称	分类的描述	操作
<input checked="" type="checkbox"/>	1	手机数码	手机数码	修改 删除
<input checked="" type="checkbox"/>	2	电脑办公	电脑办公	修改 删除
<input checked="" type="checkbox"/>	3	烟酒糖茶	烟酒糖茶	修改 删除
<input checked="" type="checkbox"/>	4	鞋靴箱包	鞋靴箱包	修改 删除
<input checked="" type="checkbox"/>	5	汽车用品	汽车用品	修改 删除

## 5.2 分析:

### 5.2.1 步骤分析:

【步骤一】 创建一个HTML页面.

【步骤二】 确定事件: 复选框的单击事件.

【步骤三】 触发一个函数

【步骤四】 在函数中,获得上面的复选框是否被选中.

【步骤五】 如果选中,下面的所有的复选框都被选中.

【步骤六】 如果不选中,下面的所有的复选框都不选中.

### 5.3 代码实现:

```
[mw_shl_code=javascript,true]function checkAll(){
    // 获得上面的复选框
    var selectAll = document.getElementById("selectAll");
    // 判断这个复选框是否被选中.
    var ids = document.getElementsByName("ids");
    if(selectAll.checked == true){
        // 上面复选框被选中: 获得下面所有的复选框,修改checked属性
        for(var i = 0 ;i<ids.length;i++){
            ids.checked = true;
        }
    }else{
        // 上面复选框没有被选中: 获得下面所有的复选框,修改checked属性
        for(var i = 0 ;i<ids.length;i++){
            ids.checked = false;
        }
    }
}[/mw_shl_code]
```

## 5.4 总结:

### 5.4.1 JS中的DOM对象:

## 【DOM的概述】

什么是DOM

DOM: Document Object Model: 文档对象模型.

将一个HTML的文档加载到内存形成一个树形结构,从而操作树形结构就可以改变HTML的样子

DOM的使用:

知道document, element, attribute中的属性和方法

## 【DOM的常用的操作】

获得元素:

```
* document.getElementById();           -- 通过ID获得元素.
* document.getElementsByName();        -- 通过name属性获得元素.
* document.getElementsByTagName();      -- 通过标签名获得元素.
```

创建元素:

```
* document.createElement();            -- 创建元素
* document.createTextNode();           -- 创建文本
```

添加节点:

```
* element.appendChild();              -- 在最后添加一个节点.
* element.insertBefore();             -- 在某个元素之前插入.
```

删除节点:

```
* element.removeChild();              -- 删除元素
```

## 【使用DOM完成对ul中添加一个li元素】

```
[mw_shl_code=javascript,true]function addElement(){
    var city = document.getElementById("city");
    // 创建一个元素:
    var liEl = document.createElement("li");
    // 创建一个文本节点:
    var text = document.createTextNode("深圳");
    // 添加子节点:
    liEl.appendChild(text);
```

```
city.appendChild(liEl);  
}[/mw_shl_code]
```

## 6 案例六：使用JS完成省市联动的效果：

### 6.1 需求：

在注册页面上有两个下拉列表,左侧省份的下拉列表一改变,右侧的市的下拉列表也要跟着发生变化.

用户

密码

确认密码

性别 ☐男 ☐女

籍贯

爱好 ☐篮球 ☐足球 ☐排球 ☐羽毛球

邮箱

### 6.2 分析：

#### 6.2.1 技术分析：

##### 【JS中创建数组】

New Array()

New Array(size)

New Array(element0,element1,.....)

##### 【JS的事件】

下拉的列表的改变的事件.onChange.

##### 【JS的DOM的操作】

创建元素：

添加元素：

### 6.2.2 步骤分析

【步骤一】 创建一个HTML文件.

【步骤二】 确定事件: onchange事件.

【步骤三】 触发函数,在函数中编写代码.

【步骤四】 获得到所选择的省份的信息.

【步骤五】 根据选择的省份的信息获得到对应的数组中的市的数据.

【步骤六】 遍历数组中的市的信息.

【步骤七】 创建元素, 创建文本,最后将元素添加到第二个列表中.

### 6.3 代码实现:

```
[mw_shl_code=javascript,true]// 定义数组: 二维数组:
var arrs = new Array(5);
arrs[0] = new Array("杭州市","绍兴市","温州市","义乌市","嘉兴市");
arrs[1] = new Array("南京市","苏州市","扬州市","无锡市");
arrs[2] = new Array("武汉市","襄阳市","荆州市","宜昌市","恩施");
arrs[3] = new Array("石家庄市","唐山市","保定市","邢台市","廊坊市");
arrs[4] = new Array("长春市","吉林市","四平市","延边市");

function changeCity(value){
    // 获得到选中的省份的信息.
    var city = document.getElementById("city");

    // 清除第二个列表中的内容:
    for(var i=city.options.length;i>0;i--){
        city.options = null;
    }
    // city.options.length = 0;

    // alert(value);
    for(var i= 0 ;i< arrs.length;i++){
        if(value == i){
            // 获得所有的市的信息.
            for(var j=0;j<arrs.length;j++){
                // alert(arrs[j]);
                // 创建元素:
                var opEl = document.createElement("option");// <option></option>
                // 创建文本节点:
```

```

        var textNode = document.createTextNode(arrs[j]);
        // 将文本的内容添加到option元素中.
        opEl.appendChild(textNode);
        // 将option的元素添加到第二个列表中.
        city.appendChild(opEl);
    }
}
}
}[/mw_shl_code]

```

## 6.4 总结:

### 6.4.1 JS的内置对象:

Array

Boolean

Date

Math

Number

String

Regexp

Function

Events

#### Array:

- \* concat()连接数组
- \* join()所有元素放入一个字符串
- \* reverse()元素顺序反转
- \* sort()元素排序
- \* toString()转为字符串
- \* value()返回原始值

#### Boolean:

- \* toString()转为字符串
- \* valueOf()返回原始值

#### Date:

- \* Date()当前日期和时间
- \* getDate()date对象返回一月中的某一天

- \* getTime()返回从1970至今的毫秒数
- \* parse()返回从1970至指定日期的毫秒值

Math对象:

- \* abs(x)绝对值
- \* radom()返回0~1的随机值
- \* valueOf()返回Math对象原始值

String对象:

- \* charAt();返回在指定位置的字符
- \* indexOf();检索字符串
- \* lastIndexOf();从后向前搜索字符串
- \* split();把字符串分割为字符串数组
- \* replace();替换与正则表达式匹配的子串
- \* substring();提取字符串中两个指定的索引号之间的字符
- \* substr();从起始索引号提取字符串中指定数目的字符

6.4.2 JS的全局函数:

- \* parseInt();解析字符串返回一个整数
  - \* parseInt("11");
- \* parseFloat();解析字符串返回一个浮点数
  - \* parseFloat("32.09");
- \* 编码和解码的方法:

// 解码

- \* decodeURI();
- \* decodeURIComponent();

// 编码

- \* encodeURI();
- \* encodeURIComponent();

eval()解析js字符串并作为脚本运行:

- \* 将一段内容当成是JS的代码执行.

```
//var sss = "alert('aaaa');"
```

```
//eval(sss);
```



回帖

— 47条回帖 —



橘子哥

很赞!

沙发 • 2016-11-28 09:39:18

回帖



来到精彩世界

挺详细，谢谢分享

藤椅 • 2016-11-28 23:39:06

回帖



javantiger93000

赞一个，，，，

板凳 • 2016-11-29 12:12:01

回帖



你好，师姐

得劲,是个大神的风格

报纸 • 2016-11-30 01:02:56

回帖



ztlm

刚学着快

地板 • 2016-12-2 18:25:28

回帖

 来自宇宙超级黑马专属安卓客户端



黑马嗨嗨嗨

厉害了！收藏

7# • 2016-12-4 20:05:57

回帖



aA772807986



学习了.....

8# • 2016-12-7 22:41:03

回帖



xike1024

谢谢分享

9# • 2016-12-8 13:12:23

回帖



孤独于我

济南厉害，真不错

10# • 2016-12-22 21:26:59

回帖



小陵不懂

收藏一下,以后备用学习

11# • 2016-12-26 09:35:39

回帖



@211

好详细呀！赞赞

12# • 2017-1-1 16:33:58

回帖



事难懂

好东西，收藏了

13# • 2017-1-3 13:21:02

回帖



hittor

感谢分享！

14# • 2017-1-3 19:16:37

回帖



vip825779

不错正好需要

15# • 2017-1-4 08:50:18

回帖

Yin,,,Yan



好帖子 收藏下

16# • 2017-1-10 00:05:15

回帖



zmanx

还没有学到这里呢,不过后期应该用的上的所以先mark一下

17# • 2017-2-12 22:14:47

回帖



爱吃橘子的小泽

顶一下,祝黑马越办越好

18# • 2017-2-12 22:15:48

回帖



a690223483

顶顶顶 已关注

19# • 2017-2-22 21:16:09

回帖



zhangkaitong

谢谢分享 赞赞赞!!!!

20# • 2017-2-24 13:25:23

回帖

上一页

1/3

下一页