

第三章 WEB03_JS 篇

今日任务

- 使用 JS 完成简单的数据校验(document 对象、事件、函数)
- 使用 JS 完成图片轮播效果(事件、定时操作)
- 使用 JS 完成页面定时弹出广告(总结 BOM 对象以及 JS 的引入方式)
- 使用 JS 完成表单的校验(总结常用事件)


教学导航

教学目标	掌握 JS 的基本语法,数据类型,能够使用 JS 完成简单的页面交互。 掌握 JS 中的 BOM 对象以及 JS 的引入方式 掌握 JS 中的常用的事件
教学方法	案例驱动法

一、使用 JS 完成注册表单数据校验

1. 需求分析

用户在进行注册的时候会输入一些内容,但是有些用户会输入一些不合法的内容,这样会导致服务器的压力过大,此时我们需要对用户输入的内容进行一个校验(前端校验和后台校验),前端校验防君子不防小人。

会员注册 USER REGISTER	
用户名	<input type="text"/>
密码	<input type="password"/>
确认密码	<input type="password"/>
Email	<input type="text"/>
姓名	<input type="text"/>
性别	<input type="radio"/> 男 <input type="radio"/> 女
出生日期	<input type="text"/>
验证码	<input type="text"/> 
<input type="button" value="注册"/>	

2. 技术分析

2.1 javascript 的介绍

什么是 JavaScript?

- JavaScript 被设计用来向 HTML 页面添加交互行为。
- JavaScript 是一种脚本语言（脚本语言是一种轻量级的编程语言）。
- JavaScript 由数行可执行计算机代码组成。
- JavaScript 通常被直接嵌入 HTML 页面。
- JavaScript 是一种解释性语言（就是说，代码执行不进行预编译）。
- 所有的人无需购买许可证均可使用 JavaScript。

Java 与 javascript 有什么区别？

它们没有关系，雷锋和雷峰塔的关系。

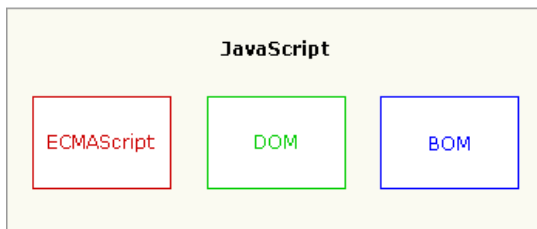
2.2 javascript 的作用？

HTML：它是整个网站的骨架。

CSS：它是对整个网站骨架的内容进行美化(修饰)

Javascript：它能够让整个页面具有动态效果。

2.3 javascript 的组成部分



ECMAScript:它是整个 javascript 的核心，包含(基本语法、变量、关键字、保留字、数据类型、语句、函数等等)

DOM: 文档对象模型，包含(整个 html 页面的内容)、

BOM: 浏览器对象模型,包含(整个浏览器相关内容)

2.4 javascript 语法

区分大小写

变量是弱类型的(`String str="aaa"` ,`var str="123";`)

每行结尾的分号可有可无(建议大家写上)

注释与 java、php 等语言相同。

2.5 javascript 的变量

变量可以不用声明，变量是弱类型。统一使用 **var** 来定义！定义变量的时候不要使用关键字和保留字。

2.6 javascript 数据类型

JavaScript 数据类型分为原始数据类型和引用数据类型
原始数据类型：

string、number、boolean、null、undefined

引用数据类型：

[Array](#)
[Boolean](#)
[Date](#)
[Math](#)
[Number](#)
[String](#)
[RegExp](#)

2.7 javascript 运算符

其它运算符与 java 大体一致，需要注意其等性运算符。

== 它在做比较的时候会进行自动转换。

=== 它在做比较的时候不会进行自动转换。

2.8 javascript 语句

所有语句与 java 大体一致。

2.9 获取元素内容

获取元素

```
document.getElementById("id 名称");
```

获取元素里面的值

```
document.getElementById("id 名称").value;
```

批注 [ThinkPad1]: 如果 id 是一个字符串，那么必须加上引号，如果是一个变量那么不需要。

2.10 javascript 事件

表单提交事件: `onsubmit`

2.11 javascripte 的输出

警告框: `alert()`;

向页面指定位置写入内容: `innerHTML`(属性)

向页面写入内容: `document.write()`;

3.步骤分析

第一步: 确定事件(`onsubmit`)并为其绑定一个函数

第二步: 书写这个函数(获取用户输入的数据<获取数据时需要在指定位置定义一个 `id`>)

第三步: 对用户输入的数据进行判断

第四步: 数据合法(让表单提交)

第五步: 数据非法(给出错误提示信息, 不让表单提交)

问题: 如何控制表单提交?

关于事件 `onsubmit`: 一般用于表单提交的位置, 那么需要在定义函数的时候给出一个返回值。 `onsubmit = return checkForm()`

4.代码实现

JS 代码:

```
<script>
function checkForm(){
    //alert("aa");
    /**校验用户名*/

    //1. 获取用户输入的数据
    var uValue = document.getElementById("user").value;
    //alert(uValue);
    if(uValue==""){
        //2. 给出错误提示信息

        alert("用户名不能为空!");

        return false;
    }
}
```

```
    }

    /*校验密码*/
    var pValue = document.getElementById("password").value;
    if(pValue==""){
        alert("密码不能为空!");
        return false;
    }

    /**校验确认密码*/
    var rpValue =
document.getElementById("repassword").value;
    if(rpValue!=pValue){
        alert("两次密码输入不一致!");
        return false;
    }

    /*校验邮箱*/
    var eValue = document.getElementById("eamil").value;
    if(!/^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+/.test(eValue)){
        alert("邮箱格式不正确!");
        return false;
    }
}
</script>
```

Html 部分:

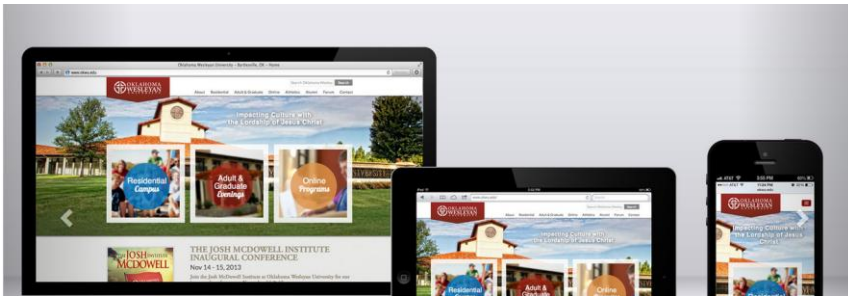
```
<form      action="#"      method="get"      name="regForm"
onsubmit="return checkForm()">
```

需要在指定位置添加 id

二、使用 JS 完成首页轮播图效果案例

1. 需求分析

我们希望在首页完成对轮播图的效果实现：



2. 技术分析

获取元素 `document.getElementById("id 名称")`

事件(`onload`)

定时操作: `setInterval("changeImg()", 3000);`

3. 步骤分析

第一步：确定事件(`onload`)并为其绑定一个函数

第二步：书写绑定的这个函数

第三步：书写定时任务(`setInterval`)

第四步：书写定时任务里面的函数

第五步：通过变量的方式，进行循环(获取轮播图的位置，并设置 `src` 属性)

批注 [ThinkPad2]：在循环的时候需要注意到了最后一张图片的时候要重置。

4. 代码实现

JS 代码：

```
<script>
function init(){
```

```
//书写轮图片显示的定时操作
setInterval("changeImg()",3000);
}

//书写函数
var i=0
function changeImg(){
    i++;
    //获取图片位置并设置src属性值

    document.getElementById("img1").src="../img/small0"+i+".jpg";
    if(i==3){
        i=0;
    }
}
</script>
```

Html:

```
<body onload="init()">
```

在指定位置定义 id。

三、使用 JS 完成页面定时弹出广告

1. 需求分析

我们希望在首页中的顶部做一个定时弹出广告图片。其实现效果如下:



2. 技术分析

获取图片的位置(document.getElementById(""))

传智播客致力打造专业的 IT 高级培训课程——务实、创新、质量、专注、分享、责任

隐藏图片: `display: none`

定时操作: `setInterval("显示图片的函数", 3000);`

3. 步骤分析

第一步: 在页面指定位置隐藏一个广告图片(使用 `display` 属性的 `none` 值)

第二步: 确定事件(`onload`)并为其绑定一个函数

第三步: 书写这个函数(设置一个显示图片的定时操作)

第四步: 书写定时器中的函数(获取广告图片的位置并设置属性 `style` 的 `display` 值 `block`)

第五步: 清除显示图片的定时操作()

第六步: 书写隐藏图片的定时操作

第七步: 书写定时器中的函数(获取广告图片的位置并设置属性 `style` 的 `display` 值 `none`)

第八步: 清除隐藏图片的定时操作()

4. 代码实现

```
<script>
    function init(){
        //书写轮图显示的定时操作
        setInterval("changeImg()", 3000);

        //1. 设置显示广告图片的定时操作
        time = setInterval("showAd()", 3000);
    }

    //书写函数
    var i=0
    function changeImg(){
        i++;
        //获取图片位置并设置src属性值
        document.getElementById("img1").src="../img/"+i+".jpg";
        if(i==3){
            i=0;
        }
    }

    //2. 书写显示广告图片的函数
```



```
function showAd(){
    //3. 获取广告图片的位置
    var adEle = document.getElementById("img2");
    //4. 修改广告图片元素里面的属性让其显示
    adEle.style.display = "block";
    //5. 清除显示图片的定时操作
    clearInterval(time);
    //6. 设置隐藏图片的定时操作
    time = setInterval("hiddenAd()", 3000);
}

//7. 书写隐藏广告图片的函数
function hiddenAd(){
    //8. 获取广告图片并设置其style属性的display值为none
    document.getElementById("img2").style.display = "none";
    //9. 清除隐藏广告图片的定时操作
    clearInterval(time);
}

</script>
```

Html 代码:

```
<body onload="init()">
```

```
<!-- 定时弹出广告图片位置 -->
```

```

```

5. 总结

5.1 javascript 的引入方式

➤ 内部引入方式

直接将 javascript 代码写到 `<script type="text/javascript"></script>`

➤ 外部引入方式

需要创建一个 .js 文件, 在里面书写 javascript 代码, 然后在 html 文件中通过 script 标签的 src 属性引入该外部的 js 文件

5.2 BOM 对象

BOM 对象: 浏览器对象模型 (操作与浏览器相关的内容)

➤ Window 对象

Window 对象表示浏览器中打开的窗口。

Window 对象方法

方法	描述
alert()	显示带有一段消息和一个确认按钮的警告框。
blur()	把键盘焦点从顶层窗口移开。
clearInterval()	取消由 setInterval() 设置的 timeout。
clearTimeout()	取消由 setTimeout() 方法设置的 timeout。
close()	关闭浏览器窗口。
confirm()	显示带有一段消息以及确认按钮和取消按钮的对话框。
createPopup()	创建一个 pop-up 窗口。
focus()	把键盘焦点给予一个窗口。
moveBy()	可相对窗口的当前坐标把它移动指定的像素。
moveTo()	把窗口的左上角移动到一个指定的坐标。
open()	打开一个新的浏览器窗口或查找一个已命名的窗口。
print()	打印当前窗口的内容。
prompt()	显示可提示用户输入的对话框。
resizeBy()	按照指定的像素调整窗口的大小。
resizeTo()	把窗口的大小调整到指定的宽度和高度。
scrollBy()	按照指定的像素值来滚动内容。
scrollTo()	把内容滚动到指定的坐标。
setInterval()	按照指定的周期（以毫秒计）来调用函数或计算表达式。
setTimeout()	在指定的毫秒数后调用函数或计算表达式。

[setInterval\(\)](#):它有一个返回值，主要是提供给 [clearInterval](#) 使用。

[setTimeout\(\)](#):它有一个返回值，主要是提供给 [clearTimeout](#) 使用。

[clearInterval\(\)](#):该方法只能清除由 [setInterval](#) 设置的定时操作

[clearTimeout\(\)](#):该方法只能清除由 [setTimeout](#) 设置的定时操作

弹出框的几个方法:

```
<script>
    //警告框
    //alert("aaa");

    //确认按钮
    //confirm("您确认删除吗? ");

    //提示输入框
    prompt("请输入价格: ");
</script>
```

</head>

> Location 对象

Location 对象包含有关当前 URL 的信息。

Location 对象属性

属性	描述
hash	设置或返回从井号 (#) 开始的 URL (锚)。
host	设置或返回主机名和当前 URL 的端口号。
hostname	设置或返回当前 URL 的主机名。
href	设置或返回完整的 URL。
pathname	设置或返回当前 URL 的路径部分。
port	设置或返回当前 URL 的端口号。
protocol	设置或返回当前 URL 的协议。
search	设置或返回从问号 (?) 开始的 URL (查询部分)。

href: 该属性可以完成通过 JS 代码控制页面的跳转。

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>location对象</title>
    <script>
      function tiao(){
        window.location.href="http://www.itcast.cn";
      }
    </script>
  </head>
  <body>
    <a href="#" onclick="tiao()">跳转到传智播客首页</a>
  </body>
</html>
```

> History 对象

History 对象包含用户（在浏览器窗口中）访问过的 URL。

History 对象方法

方法	描述
<code>back()</code>	加载 history 列表中的前一个 URL。
<code>forward()</code>	加载 history 列表中的下一个 URL。
<code>go()</code>	加载 history 列表中的某个具体页面。

历史页面：使用 location 页面(把 href 属性值改为当前的 history)

History 页面代码：

```
<input type="button" value="返回上一个页面"
onclick="javascript:history.back()"/>
```

go(参数)

参数：-1 返回上一个历史记录页面；-2 返回上上一个历史记录页面，1 进入下一个历史记录页面。

让按钮点击失效：

```
onclick="javascript:void(0)"
```

> Navigator 对象

Navigator 对象包含有关浏览器的信息。(该对象开发中不怎么常用)

Navigator 对象属性

属性	描述
<code>appName</code>	返回浏览器的代码名。
<code>appMinorVersion</code>	返回浏览器的次级版本。
<code>appName</code>	返回浏览器的名称。
<code>appVersion</code>	返回浏览器的平台和版本信息。
<code>browserLanguage</code>	返回当前浏览器的语言。
<code>cookieEnabled</code>	返回指明浏览器中是否启用 cookie 的布尔值。
<code>cpuClass</code>	返回浏览器系统的 CPU 等级。
<code>onLine</code>	返回指明系统是否处于脱机模式的布尔值。
<code>platform</code>	返回运行浏览器的操作系统平台。
<code>systemLanguage</code>	返回 OS 使用的默认语言。
<code>userAgent</code>	返回由客户机发送服务器的 user-agent 头部的值。
<code>userLanguage</code>	返回 OS 的自然语言设置。

> Screen 对象

Screen 对象包含有关客户端显示屏幕的信息。(该对象开发中不怎么常用)

Screen 对象属性

属性	描述
availHeight	返回显示屏幕的高度 (除 Windows 任务栏之外)。
availWidth	返回显示屏幕的宽度 (除 Windows 任务栏之外)。
bufferDepth	设置或返回调色板的比特深度。
colorDepth	返回目标设备或缓冲器上的调色板的比特深度。
deviceXDPI	返回显示屏幕的每英寸水平点数。
deviceYDPI	返回显示屏幕的每英寸垂直点数。
fontSmoothingEnabled	返回用户是否在显示控制面板中启用了字体平滑。
height	返回显示屏幕的高度。
logicalXDPI	返回显示屏幕每英寸的水平方向的常规点数。
logicalYDPI	返回显示屏幕每英寸的垂直方向的常规点数。
pixelDepth	返回显示屏幕的颜色分辨率 (比特每像素)。
updateInterval	设置或返回屏幕的刷新率。
width	返回显示器屏幕的宽度。

四、使用 JS 完成注册页面表单校验

1. 需求分析

之前我们已经使用弹出框的方式实现了表单校验的功能，但是此种方式用户体验效果极差！我们希望做成如下这种效果：

会员注册 USER REGISTER	
用户名	<input type="text"/> 提示信息和校验结果显示在该位置
密码	<input type="password"/>
确认密码	<input type="password"/>
Email	<input type="text"/>
姓名	<input type="text"/>
性别	<input type="radio"/> 男 <input type="radio"/> 女
出生日期	<input type="text"/>
验证码	<input type="text"/> 
<input type="button" value="注册"/>	

2. 技术分析

3. 步骤分析

第一步：确定事件(onfocus 聚焦事件)并为其绑定一个函数

第二步：书写绑定函数(在输入框的后面给出提示信息)

第三步：确定事件(onblur 离焦事件)并为其绑定一个函数

第四步：书写函数(对数据进行校验，分别给出提示)

4. 代码实现

Html 代码：

```
<input type="text" name="user" size="34px" id="user"
onfocus="showTips('user',' 用 户 名 必 填 ！  ')"
onblur="check('user',' 用 户 名 不 能 为 空 ！  ')" /><span
id="userspan"></span>
```

JS 代码：

```
function showTips(id,info){
```

```
document.getElementById(id+"span").innerHTML("<font  
color='gray'>" + info + "</font>");  
}  
  
function check(id,info){  
    //1.获取用户输入的用户名数据  
    var uValue = document.getElementById(id).value;  
    //2.进行校验  
    if(uValue==""){  
        document.getElementById(id+"span").innerHTML("<font  
color='red'>" + info + "</font>");  
    }else{  
        document.getElementById(id+"span").innerHTML="";  
    }  
}
```