



东方标准人才服务有限公司

Oriental Standard Human Resources CO.,LTD



JavaScript基础

主讲人:Yinzhi Geng

章节内容

www.oristand.com

- 1 添加题目
- 2 添加题目
- 3 添加题目
- 4 添加题目

JavaScript

- JavaScript是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言， JavaScript的脚本包括在HTML中，它成为HTML文档的一部分。与HTML标识相结合，构成了一个功能强大的Internet网上编程语言 JavaScript代码由 `<Script Language="JavaScript">...</Script>` 说明。
- 在标识`<Script Language="JavaScript">...</Script>` 之间就可加入JavaScript脚本。

示例

```
<Script Language = "JavaScript">  
alert("这是第一个JavaScript例子!");  
alert("欢迎你进入JavaScript世界!");  
alert("今后我们将共同学习JavaScript知识! ");  
</Script>
```

通过属性 `Language = "JavaScript"` 说明标识中是使用的何种语言，这里是JavaScript语言，表示在JavaScript中使用的语言 `alert()` 是JavaScript的窗口对象方法JavaScript 以 `</Script>` 标签结束

再如：`document.write(document.lastModified);`

window对象方法

- `alert()`:警告对话框
- `confirm()`:确认对话框
- `prompt()`:接收用户输入的询问对话框

JavaScript基本数据结构

- 数值（整数和实数）、字符串型（用“”号或‘’括起来的字符或数值）、布尔型（使**True**或**False**表示）和空值
- 在JavaScript中，变量可以用命令**Var**作声明 ,在JavaScript中，变量以可以不作声明，而在使用时再根据数据的类型来确其变量的类型

变量的声明及其作用域

- **JavaScript**变量可以在使用前先作声明，并可赋值。通过使用**var**关键字对变量作声明
- 对于变量还有一个重要性——那就是变量的作用域。在**JavaScript**中同样有全局变量和局部变量。全局变量是定义在所有函数体之外，其作用范围是整个函数；而局部变量是定义在函数体之内，只对该函数是可见的，而对其它函数则是不可见的

表达式和运算符

- 在定义完变量后，就可以对它们进行赋值、改变、计算等一系列操作，这一过程通常又叫称一个叫表达式来完成，可以说它是变量、常量、布尔及运算符的集合
- 算术运算符
+（加）、-（减）、*（乘）、/（除）、%（取模）
- 比较运算符
<（小于）、>（大于）、<=（小于等于）、>=（大于等于）、==(等于)、!=(不等于)
- 布尔逻辑运算符
?:（三目操作符）、||（或）、==(等于)、!=(不等于)。

JavaScript程序构成

- 基本构成是由控制语句、函数、对象、方法、属性等,来实现编程的

程序控制流

if条件语句

基本格式

if（表述式）

语句段 1 ；

else

语句段 2 ；

功能：若表达式为**true**，则执行语句段 1 ； 否则执行语句段 2 。

if -else 语句说明

- **if -else** 语句是JavaScript中最基本的控制语句，通过它可以改变语句的执行顺序
- 表达式中必须使用关系语句，来实现判断，它是作为一个布尔值来估算的。它将零和非零的数分别转化成**false**和**true**。若**if**后的语句有多行，则必须使用花括号将其括起来。
- **if**语句的嵌套
 if (布尔值) 语句 1 ;
 else (布尔值) 语句 2 ;
 else if (布尔值) 语句 3 ;
 else 语句 4 ;
- 在这种情况下，每一级的布尔表述式都会被计算，若为真，则执行其相应的语句，否则执行**else**后的语句。

For循环语句

- **for**（初始化；条件；增量）
- 功能：实现条件循环，当条件成立时，执行语句集，否则跳出循环体。
- 说明：初始化参数告诉循环的开始位置，必须赋予变量的初值
- 条件：是用于判别循环停止时的条件。若条件满足，则执行循环体，否则 跳出
- 增量：主要定义循环控制变量在每次循环时按什么方式变化
- 三个主要语句之间，必须使用逗号分隔。

while循环

while (条件)
语句集;

该语句与**For**语句一样，当条件为真时，重复循环，否则退出循环。

break和continue语句

- 使用**break**语句使得循环从**For**或**while**中跳出，**continue**使得跳过循环内剩余的语句而进入下一次循环

数组

两种创建方式:

```
msgArray = new Array()
```

```
new Array(arrayLength);
```

```
new Array(element0, element1, ..., elementN);
```


函数

- 将程序划分为一些相对独立的部分，每部分编写一个函数。从而，使各部分充分独立，任务单一，程序清晰，易懂、易读、易维护。**JavaScript**函数可以封装那些在程序中可能要多用到的模块。并可作为事件驱动的结果而调用的程序。从而实现一个函数把它与事件驱动相关联

```
Function 函数名 (参数,变元) {  
    函数体;  
    Return 表达式;  
}
```

- 当调用函数时,所用变量或字面量均可作为变元传递
- 函数由关键字**Function**定义。函数名：定义自己函数的名字。参数表，是传递给函数使用或操作的值，其值可以是常量，变量或其它表达式。通过指定函数名（实参）来调用一个函数。函数名对大小写是敏感的。

函数中的形式参数

- 在函数的定义中，我们看到函数名后有参数表，这些参数变量可能是一个或几个。那么怎样才能确定参数变量的个数呢？在JavaScript中可通过**arguments .Length**来检查参数的个数。例：

```
Function function_Name(exp1,exp2,exp3,exp4)
Number =function _Name . arguments .length;
if (Number>1)
document.write(exp2);
if (Number>2)
document.write(exp3);
if(Number>3)
document.write(exp4);
```

事件驱动

- 通常鼠标或热键的动作我们称之为事件（**Event**），而由鼠标或热键引发的一连串程序的动作，称之为事件驱动（**Event Driver**）。而对事件进行处理程序或函数，我们称之为事件处理程序（**Event Handler**）。

事件

- 1.onClick 当对象被点击
- 2.onLoad 当网页打开,只能书写在body中
- 3.onUnload 当网页关闭或离开时,只能书写在body中
- 4.onmouseover 当鼠标悬于其上时
- 5.onmouseout 当鼠标离开对象时
- 6.onmouseup 当鼠标松开

- 7.onmousedown 当鼠标按下键
- 8.onFocus 当对象获取焦点时
- 9.onSelect 当对象的文本被选中时
- 10.onChange 当对象的内容被改变
- 11.onBlur 当对象失去焦点
- 12.onSubmit=return(ss()) 表单调用时返回的值

常用方法

- Math.random();
- getDate();
- substr(int,int);
- replace(str1,str2);

selectedIndex

- options[selectedIndex].value表示下拉列表的值

浏览器.窗口对象

浏览器对象(Navigator),如navigator.appName,区别与
navigate(转向)

提供有关浏览器的信息

窗口对象(Windows)(self表示当前引用的窗口)

Window对象处于对象层次的最顶端,它提供了处理Navigator窗口
的方法和属性。包括open,print,close,confirm等

窗口及输入输出

- 有关输入可通过窗口（**Window**）对象来完成，而输出可通过文档（**document**）对象的方法来实现
- `Var test=window.prompt("请输入数据:");`
- `document.write(test+"JavaScript输入输出的例子");`
- 其中**window.prompt()**就是一个窗口对象的方法，其基本作用是，当装入**Web**页面时在屏幕上显示一个具有“确定”和“取消”的对话框，让你输出数据。**document.write**是一个文档对象的方法，它的基本功能，是实现**Web**页面的输出显示

Javascript弹出窗口

- `window.open ("page.html", "newwindow", "height=100, width=400, top=0, left=0, toolbar=no, menubar=no, scrollbars=no, resizable=no,location=no, status=no")`

常用对象

- 位置对象(Location)
Location对象提供了与当前打开的URL一起工作的方法和属性，它是一个静态的对象。如href,replace()
- 历史对象(History)
History对象提供了与历史清单有关的信息。如back,go(-1)
- 文档对象(Document)
document对象包含了与文档元素(elements)一起工作的对象，如title,bgcolor,write.reload等属性和方法

窗体

- 窗体（**Form**）：它构成了**Web**页面的基本元素。通常一个**Web**页面有一个窗体或几个窗体，使用**Forms[]**数组来实现不同窗体的访问
- 在**JavaScript**中要对窗体引用的条件是：必须先页面中用标识创建窗体

窗体提交方式

- 在**Form**表单中定义**submit**提交按钮,点击**submit**按钮即可提交表单中的数据
- 窗体对象的方法**submit()**方法,该方法主要功用就是实现窗体信息的提交

`document.myform.submit()`

访问窗体对象

- 通过访问窗体
在窗体对象的属性中首先必须指定其窗体名，而后就可以通过下列标识访问窗体如：**document.myform**
- 通过数组来访问窗体
除了使用窗体名来访问窗体外，还可以使用窗体对象数组来访问窗体对象。但需要注意一点，因窗体对象是由浏览器环境提供的，而浏览器环境所提供的数组下标是由**0**到**n**。所以可通过下列格式实现窗体对象的访问：
document.forms[0]
document.forms[1]

innerHTML

- innerHTML获取元素的源代码
- innerText

框架

框架可以将屏幕分割成不同的区域，每个区域有自己的URL，通过Frames[]数组对象来实现不同框架的访问

```
<Frameset Rows="20%,80%">  
  <frame src="test9_1.html">  
  <Frameset Cols="50%,50%">  
    <frame src="test9_2.html">  
    <frame src="test9_3.html">  
  </Frameset>  
</Frameset>
```

访问框架

- `parent.frames[Index1].document.forms[index2]`
- 通过`parent.frames.length`确定窗口中窗体的数目。

IFRAME实现网页的内嵌

- **iframe**元素的功能是在一个文档里内嵌一个文档，创建一个浮动的帧
- 其部分属性简介如下：
 - name**: 内嵌帧名称
 - width**: 内嵌帧宽度(可用像素值或百分比)
 - height**: 内嵌帧高度(可用像素值或百分比)
 - frameborder**: 内嵌帧边框
 - marginwidth**: 帧内文本的左右页边距
 - marginheight**: 帧内文本的上下页边距
 - scrolling**: 是否出现滚动条(“auto” 自动, “yes” 显示, “no” 不显示)
 - src**: 内嵌入文件的地址
 - style**: 内嵌文档的样式(如设置文档背景等)
 - allowtransparency**: 是否允许透明

在父窗体中访问并控制子窗体中的对象

- 在父窗体中，**Iframe**即子窗体是**document**对象的一个子对象，可以直接在脚本中访问子窗体中的对象

```
<body onclick="alert(tt.myH1.innerHTML)">  
  <Iframe name="tt" src="frame1.htm" width="250"  
    height="200" scrolling="no" frameborder="0">  
  </iframe>  
  
  <h1 id="myH2">hello,buddy</h1>  
</body>
```

在子窗体中访问并控制父窗体中对象

- 在子窗体中我们可以通过其**parent**即父（双亲）对象来访问父窗口中的对象
- `parent.控件name.value="hello"`

视频播放案例

window.open() 方法

- 语法: `window.open(pageurl,name,parameters);`
- `window`对象的`open()`方法用于创建一个新的窗口实例, 新创建的窗口的外观由参数: `parameters`指定。新窗口中打开的文档由参数
- `alwaysRaised yes/no` 指定窗口浮在所有窗口之上。
- `dependent yes/no` 指定打开的窗口为父窗口的一个子窗口。并随父窗口的关闭而关闭
- `height pixel value` 设定新窗口的像素高度。
- `hotkeys yes/no` 在没有菜单栏的新窗口设置安全退出热键。
- `location yes/ no` 指明位置栏在新窗口中是否可见。
- `menubar yes /no` 指明菜单栏在新窗口中是否可见。
- `resizable yes /no` 指明新窗口是否可以调整。
- `scrollbars yes /no` 指明滚动栏在新窗口中是否可见。
- `toolbar yes /no` 指明工具栏在新窗口中是否可见。
- `Width pixel value` 设定窗口的像素宽度。
- `fullscreen yes / no` 打开的窗体是否进行全屏显示

按键捕捉

- Ctrl+Enter按键:if (event.ctrlKey && event.keyCode == 13)
- Alt加快捷键: Alt+A
- <button accessKey=A title="Alt+A"
- if (event.keyCode < 48 || event.keyCode >57) event.returnValue = false;
- 此方法控制TextBox只收数字: 0~9, 也自
可以定义其它可输入字符,如改成: 65~123,
只允许输入: a~z和A~Z 等

屏蔽鼠标右键

- `oncontextmenu="window.event.returnValue=false"`

屏蔽键盘所有键

- `function document.onkeydown(){`
- `event.keyCode = 0;`
- `event.returnValue = false;`
- `}`

全屏打开页面

- `window.open("www.sina.com.cn","news","fullscreen=3,height=100,`
- `width=400, top=0, left=0, toolbar=no,`
`menubar=no, scrollbars=no,`
- `resizable=no,location=no, status=no");`

关闭父窗体

- `window.parent.close()`

关闭本弹出窗体,并刷新父页面

- `window.opener.location.reload();window.close();`

英文值检测

- function isEnglish(name)
- {
- if(name.length == 0)
- return false;
- for(i = 0; i < name.length; i++) {
- if(name.charCodeAt(i) > 128)
- return false;
- }
- return true;
- }

中文值检测

- function isChinese(name) {
- if(name.length == 0)
- return false;
- for(i = 0; i < name.length; i++) {
- if(name.charCodeAt(i) > 128)
- return true;
- }
- return false;
- }

数值检测

- function isNumber(name)
- {
- if(name.length == 0)
- return false;
- for(i = 0; i < name.length; i++) {
- if(name.charAt(i) < "0" || name.charAt(i) > "9")
- return false;
- }
- return true;
- }

查找对象

- `document.getElementsByName("r1");`
- `document.getElementById(id);`

定时

- `var count = 10;`
- `function test(){`
- `count = count-1;`
- `//document.write(new Date().toString());`
- `var v = new Date().toString();`
- `document.all("message").innerHTML="<font`
`size=9>"+count+"";`
- `}`
- `var timer = setInterval("test()",1000);`
- `clearInterval(timer);`

setTimeout

自动刷新

- `<meta HTTP-EQUIV="refresh" CONTENT="8;URL=http://c98.yeah.net">`

图标

- 地址栏图标
- `<link rel="Shortcut Icon" href="favicon.ico">`
- favicon.ico 名字最好不变16*16的16色,放虚拟目录根目录下
-
- 收藏栏图标
- `<link rel="Bookmark" href="favicon.ico">`

屏幕禁止

- 禁止右键
- `document.oncontextmenu = function() { return false;}`
- 禁止保存
- `<noscript><iframe src="*.htm"></iframe></noscript>`
- 禁止选取
`<body oncontextmenu="return false" ondragstart="return false"
onselectstart="return false" onselect="document.selection.empty()"
oncopy="document.selection.empty()" onbeforecopy="return false"
onmouseup="document.selection.empty()>`
- 禁止粘贴
- `<input type="text" onpaste="return false">`

checkbox扁平

- `<input type=checkbox style="position: absolute; clip:rect(5px 15px 15px 5px)">`

Checkbox实现全选

- //第一种方法
- function selectall1()
- {
- var a = document.getElementsByTagName("input");
- if(a[0].checked==true){
- for (var i=0; i<a.length; i++)
- if (a[i].type == "checkbox") a[i].checked = false;
- }
- else
- {
- for (var i=0; i<a.length; i++)
- if (a[i].type == "checkbox") a[i].checked = true;
- }
- }

- //第二种方法
- function selectall2() {
- var tform = document.forms['form1'];
- for (var i=0;i<tform.length;i++)
- {
- var e = tform.elements[i];
- if (e.type == "checkbox")
- e.checked = !e.checked;
- }
- }

- //第二种方法
- function selectall2() {
- var tform = document.forms['form1'];
- for (var i=0;i<tform.length;i++)
- {
- var e = tform.elements[i];
- if (e.type == "checkbox")
- e.checked = !e.checked;
- }
- }

- //第四种方法
- function selectall4(id){ //用id区分
- var tform=document.forms['form1'];
- for(var i=0;i<tform.length;i++){
- var e=tform.elements[i];
- if(e.type=="checkbox" && e.name==id)
- e.checked=!e.checked;
- }
- }

- //第五种方法
- function selectall(theform,thename){ //theform指定的form,thename是checkbox的name属性
- var tform=document.forms[theform];
- document.getElementById("thewen").value='反选';
- for(var i=0;i<tform.length;i++){
- var e=tform.elements[i];
- if(e.type=='checkbox' && e.name==thename)e.checked=!e.checked;
- }
- }

title换行

- `obj.title = "123sdfs "`

表格行指示

- `<tr onmouseover="this.bgColor='#f0f0f0'"
onmouseout="this.bgColor='#ffffff'">`

自带的打印预览

- WebBrowser.ExecWB(1,1) 打开
- Web.ExecWB(2,1) 关闭现在所有的IE窗口，并打开一个新窗口
- Web.ExecWB(4,1) 保存网页
- Web.ExecWB(6,1) 打印
- Web.ExecWB(7,1) 打印预览
- Web.ExecWB(8,1) 打印页面设置
- Web.ExecWB(10,1) 查看页面属性
- Web.ExecWB(15,1) 好像是撤销，有待确认
- Web.ExecWB(17,1) 全选
- Web.ExecWB(22,1) 刷新
- Web.ExecWB(45,1) 关闭窗口体无提示

打印

- `<style media=print>`
- `.Noprint{display:none;}<!--用本样式在打印时隐藏非打印项目-->`
- `.PageNext{page-break-after: always;}<!--控制分页-->`
- `</style>`
- `<object id="WebBrowser" width=0 height=0 classid="CLSID:8856F961-340A-11D0-A96B-00C04FD705A2">`
- `</object>`
- `<center class="Noprint" >`
- `<input type=button value=打印 onclick=document.all.WebBrowser.ExecWB(6,1)>`
- `<input type=button value=直接打印 onclick=document.all.WebBrowser.ExecWB(6,6)>`
- `<input type=button value=页面设置 onclick=document.all.WebBrowser.ExecWB(8,1)>`
- `</p>`
- `<p> <input type=button value=打印预览
onclick=document.all.WebBrowser.ExecWB(7,1)>`
- `</center>`

无模式的提示框

- function modelessAlert(Msg)
- {
- window.showModelessDialog("javascript:alert(\""+escape(Msg)+"\");window.close();",
"", "status:no;resizable:no;help:no;dialogHeight:height:30px;dialogHeight:40px;");
- }

刷新后不变的文本框

- <HTML>
- <HEAD>
- <META NAME="save" CONTENT="history">
- <STYLE>
- .sHistory {behavior:url(#default#savehistory);}
- </STYLE>
- </HEAD>
- <BODY>
- <INPUT class=sHistory type=text id=oPersistInput>
- </BODY>
- </HTML>

下载文件

- function DownURL(strRemoteURL,strLocalURL)
- {
- var xmlHTTP=new ActiveXObject("Microsoft.XMLHTTP");
- xmlHTTP.open("Get",strRemoteURL,false);
- xmlHTTP.send();
- var adodbStream=new ActiveXObject("ADODB.Stream");
- adodbStream.Type=1;//1=adTypeBinary
- adodbStream.Open();
- adodbStream.write(xmlHTTP.responseBody);
- adodbStream.SaveToFile(strLocalURL,2);
- adodbStream.Close();
- adodbStream=null;
- xmlHTTP=null;
- }

检查网页是否存在

```
• function CheckURL(URL)
• {
•   var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
•   xmlhttp.Open("GET",URL, false);
•   try
•   {
•     xmlhttp.Send();
•     var result = xmlhttp.status;
•   }
•   catch(e) {return(false); }
•   if(result==200)
•   {
•     return true;
•   }
•   xmlhttp = null;
•   return false;
• }
```

使用数据岛

- <body>
- srno: <input type=text datasrc=#xmldata DataFLD=srno size="76">

- times: <input type=text datasrc=#xmldata DataFLD=times size="76">

- <input id="first" TYPE=button value="<< 第一条记录" onclick="xmldata.recordset.moveFirst()">
- <input id="prev" TYPE=button value="< 上一条记录" onclick="xmldata.recordset.movePrevious()">
- <input id="next" TYPE=button value="下一条记录>" onclick="xmldata.recordset.moveNext()">
- <input id="last" TYPE=button value="最后一条记录>>" onclick="xmldata.recordset.moveLast()">
- <input id="Add" TYPE=button value="添加新记录" onclick="xmldata.recordset.addNew()">
- <XML ID="xmldata">
- <infolist>
- <info ><srno>20041025-01</srno><times>null</times></info>
- <info ><srno>20041101-09</srno><times>2004年10月1日2点22分0秒</times></info>
- </infolist>
- </XML>
- </body>

可编辑SELECT

- `<input type=text name=re_name style="width:100px;height:21px;font-size:10pt;">`
- ``
- `<select name="r00" style="margin-left:-100px;width:118px;`
- `background-color:#FFEEEE;"`
- `onChange="document.all.re_name.value=this.value;">`
- `<option value="1">11111111<option>`
- `<option value="2">222222</option>`
- `<option value="3">333333</option>`
- `</select>`
- ``

脚本永不出错

- `<SCRIPT LANGUAGE="JavaScript">`
- `function killErrors() {`
- `return true;`
- `}`
- `window.onerror = killErrors;`
- `</SCRIPT>`

广告栏

对联

创建DIV层

谢 谢!