



北京圣思园科技有限公司
<http://www.shengsiyuan.com>

主讲人：张龙

学习运用JavaScript脚本语言

- **JavaScript** 是网景（Netscape）公司开发的一种基于客户端浏览器、**基于面向对象**、事件驱动式的网页脚本语言。
JavaScript语言的前身叫作Livescript。



学习运用JavaScript脚本语言

- **JavaScript的特点:**

- 简单、易学、易用;
- 跨平台; IE、Navigator
- 符合ECMA (欧洲计算机制造协会) 标准, 可移植;
- 事件驱动式的脚本程序设计思想;
- 动态、交互式的操作方式。



学习运用JavaScript脚本语言

- **JavaScript的作用：**

- 交互式操作；
- 表单验证；
- 网页特效；
- Web游戏
- 服务器脚本开发等。



学习运用JavaScript脚本语言

- JavaScript与Java的区别体现在：
首先，它们是两个公司开发的不同的两个产品，Java是SUN公司推出的新一代面向对象的程序设计语言，特别适合于Internet应用程序开发；而JavaScript是Netscape公司的产品，其目的是为了扩展Netscape Navigator功能而开发的一种可以嵌入Web页面中的面向对象和事件驱动的解释性语言。



学习运用JavaScript脚本语言

- 其次，两种语言在其浏览器中所执行的方式不一样。**Java**的源代码在传递到客户端执行之前，必须经过编译，因而客户端上必须具有相应平台上的仿真器或解释器，它可以通过编译器或解释器实现独立于某个特定的平台编译代码的束缚。

JavaScript是一种解释性编程语言，其源代码在发往客户端执行之前不需经过编译，而是将文本格式的字符代码发送给客户，由浏览器解释执行



学习运用JavaScript脚本语言

- 第三，两种语言所采取的变量是不一样的。**Java**采用强类型变量检查，即所有变量在编译之前必须作声明。**JavaScript**中变量声明，采用其弱类型。即变量在使用前不需作声明，而是解释器在运行时检查其数据类型。



学习运用JavaScript脚本语言

- 第四，代码格式不一样。**Java**是一种与**HTML**无关的格式，必须通过像**HTML**中引用外媒体那么进行装载，其代码以字节代码的形式保存在独立的文档中。**JavaScript**的代码是一种文本字符格式，可以直接嵌入**HTML**文档中，并且可动态装载。编写**HTML**文档就像编辑文本文件一样方便。



学习运用JavaScript脚本语言

- 第五，静态绑定和动态绑定。Java采用静态编译，即Java的对象引用必须在编译时的进行，以使编译器能够实现强类型检查，如不经编译则就无法实现对象引用的检查。JavaScript采用动态联编，即JavaScript的对象引用在运行时进行检查。



学习运用JavaScript脚本语言

- **JavaScript的格式:**

- JavaScript区分大小写;
- JavaScript脚本程序须嵌入在HTML文件中;
- 每行写一条脚本语句;
- 语句末尾可以加分号;
- JavaScript脚本程序可以独立保存为一个外部文件，但其中不能包含<script></script>标签。



学习运用JavaScript脚本语言

- **JavaScript**脚本程序的几种基本格式:

- `<script>`

- `document.write("Hello World!!!");`

- `</script>`

- `<script language="JavaScript">`

- `document.write("Hello World!!!");`

- `</script>`



学习运用JavaScript脚本语言

- `<script language="JavaScript" type="text/JavaScript">
 document.write("Hello World!!!");
</script>`
- `<script language="JavaScript1.2">
 document.write("Hello World!!!");
</script>`
- `<script src="hello.js"> </script>`

注意: `document.write("Hello World!!!");` 必须保存为一个外部文件: `hello.js`



学习运用JavaScript脚本语言

- ```
<script language="JavaScript">
 <!--
 document.write("Hello World!!!");
 -->
</script>
```



# 学习运用JavaScript脚本语言

- 使用JavaScript协议:

```
请单击
```

```
请单击
```

```
请单击
```



# 学习运用JavaScript脚本语言

- 声明变量:
- `var stdId;`
- `var name,sex;`
- `var total=3721;`
- `var notNull=true;`
- `var i=j=0;`
- 也可以省略var
- `m = 9;`



# 学习运用JavaScript脚本语言

- 变量作用域：（方式、位置）
- 全局变量：在函数外声明
- 局部变量：在函数内声明
- 全局变量可在整个脚本中被使用，可在不同的窗口中相互引用（指定窗口名）





# 学习运用JavaScript脚本语言 (js1.html)

- `<Script>`  
`var globe = "JavaScript"; // globe是全局变量`  
`test();`
- `function test() {`  
    `var local = "VBScript"; // local是局部变量`  
    `document.write("<LI>" + globe );`  
    `document.write("<LI>" + local );`  
`}`
- `document.write("<LI>" + globe );`  
`document.write("<LI>" + local );`  
`</Script>`



# 学习运用JavaScript脚本语言

- **JavaScript语句**
- **with语句：**（对象操作语句）
- 功能：为一段程序建立默认对象。
- 格式：  

```
with (<对象>){
 <语句组>
}
```



# 学习运用JavaScript脚本语言 (js2.html)

- with (document) {  
    write ("欢迎你们: ");  
    write ("<Li>很高兴认识大家");  
    write ("<Li>希望我们成为朋友。");  
}



# 学习运用JavaScript脚本语言

- **for...in**语句：重复执行指定对象的所有属性
- 格式：

```
for (变量 in 对象){
 <语句组>
}
```





# 学习运用JavaScript脚本语言(js3)

- function member(name, sex) {  
 //构造函数member  
 • this.name = name;  
 • this.sex = sex;  
 • }  
  
• function showProperty(obj, objString)  
• {  
• var str = "";  
• for (var i in obj)  
• str += objString + "." + i + " = " + obj[i] + "<BR>";  
• return str;  
• }  
  
• obj = new member("刘德华", "男生");  
//建立对象实例obj  
• document.write(showProperty(obj, "person"))



# 学习运用JavaScript脚本语言

- 其他语句与我们学过的高级语言（例如java）很相似，比如if ....else, for() switch,break,continue等等，所以不再赘述



# 学习运用JavaScript脚本语言

- 对象
- 默认对象
  - 日期对象

- 格式：日期对象名称=new Date([日期参数])

日期参数：1.省略（最常用）；

2.英文-数值格式：月 日，公元年 [时：分：秒]

如：today=new Date("October 1,2008 12:00:00")

3.数值格式：公元年，月，日，[时，分，秒]

如：today=new Date(2008,10,1)



# 学习运用JavaScript脚本语言

- 日期对象的方法:

格式: 日期对象名称.方法 ([参数])





# 学习运用JavaScript脚本语言

- `getFullYear()`返回年份数
- `getFullYear()`返回年份数
- `getMonth()`返回月份数（0--11）
- `getDate()`返回日期数（1--31）
- `getDay()`返回星期数（0--6）
- `getHours()`返回时数（0--23）
- `getMinutes()`返回分数（0--59）



# 学习运用JavaScript脚本语言（js4）

- getSeconds()返回秒数（0--59）
- 等等



# 学习运用JavaScript脚本语言

- 数组对象;

## 建立数组对象:

格式 1 : 数组对象名称=new Array([元素个数])

格式 2 : 数组对象名称=new Array([[元素1][, 元素2, ...]])

格式 3 : 数组对象名称=[元素1[, 元素2, ...]]



# 学习运用JavaScript脚本语言

- `fruit=new Array(3);// fruit=new Array();`  
    `fruit[0]="苹果";`  
    `fruit[1]="梨子";`  
    `fruit[2]="橘子";`

`fruit=new Array("苹果", "梨子", "橘子");`

`fruit=["苹果", "梨子", "橘子"];`





# 学习运用JavaScript脚本语言

- 数组对象的属性:

格式: 数组对象名称.属性

属性: length      数组长度



# 学习运用JavaScript脚本语言(js5)

- <Script>
- var fruit = new Array("苹果", "梨子", "橘子");  
var i;

```
for (i=0; i < fruit.length; i++) {
 document.write("fruit [" + i + "] = ",
fruit[i], "
")
}
```

</Script>



# 学习运用JavaScript脚本语言

- 数组对象的方法:

格式: 数组对象名称.方法([参数])



# 学习运用JavaScript脚本语言

- `join([分隔符])`数组元素组合为字符串
- `toString()`以字符串表示数组
- `reverse()`数组反转
- `valueOf()`返回数组值





# 学习运用JavaScript脚本语言(js6)

- `<Script> var fruit = new Array("苹果", "梨子", "橘子");`

```
document.write("", fruit.join());
document.write("", fruit.join("、"));
document.write("", fruit.toString());
document.write("",
fruit.reverse().join());
document.write("", fruit.valueOf());
```

```
</Script>
```



# 学习运用JavaScript脚本语言(js7)

- 二维数组

- ```
<Script> var fruit = new Array(3);  
fruit[0] = new Array("苹果", 2);  
fruit[1] = new Array("梨子", 4);  
fruit[2] = new Array("橘子", 2);
```

```
for (i=0; i < fruit.length; i++) {  
    for (j=0; j < fruit[i].length; j++) {  
        document.write("fruit [" + i + "][" + j + "] = ",  
fruit[i][j], "<br>");  
    }  
    document.write("<br>");  
}
```

```
</Script>
```



学习运用JavaScript脚本语言

- 字符串对象;

建立字符串对象:

格式: 字符串对象名称=new String(字符串常量)
格式: 字符串变量名称="字符串常量"

字符串对象的属性:

格式: 字符串对象名称.属性

属性: length 字符串长度

字符串对象的方法:

格式: 字符串对象名称.方法



学习运用JavaScript脚本语言

- `charAt(索引)`返回索引位置的字符
- `indexOf("字符串" [,索引])`返回字符串在对象中的索引位置
- `lastIndexOf("字符串" [,索引])`返回字符串在对象中的索引位置（反向搜索）
- `replace("字符串1","字符串2")`字符串2替换字符串1
- `search("字符串")`返回字符串在对象中的索引位置
- `substring(索引i[,索引j])`返回索引i到索引j-1的子串
- 等等



学习运用JavaScript脚本语言(js8)

- <Script>

```
var str = "JavaScript";  
var num = 1234;
```

```
document.write(str.toUpperCase(),  
"<BR>");  
document.write(num.toString().charAt(2),  
"<BR>");  
document.write(str.substring(0,4),  
"<BR>"); </Script>
```



学习运用JavaScript脚本语言(js9)

- `<Script> function isEmail()`
- `{`
 - `var str = document.form1.add.value;`
 - `if (str.indexOf("@") == -1)`
 - `alert("请填写正确的EMail地址");`
 - `else`
 - `alert("OK!");`
- `}`
- `</Script>`
- `<FORM name="form1">`
 - 请输入你的 EMail 地址:
 - `<INPUT TYPE="text" name="add">`
 - `<INPUT TYPE="button" value="开始检查" onClick="isEmail()>`
- `</FORM>`



学习运用JavaScript脚本语言(js10)

- 自定义对象
 - 构造函数定义对象类型;
 - 建立对象实例。



学习运用JavaScript脚本语言

- 动态的定义对象属性
 - 构造函数定义对象方法名;
 - 建立方法的描述函数。



学习运用JavaScript脚本语言(js11)

- <Script>
- function member(name, sex) {
- this.name = name;
- this.sex = sex;
- this.display = display;
- }
- function display() {
- var str = this.name + "是" + this.sex;
- document.write("" + str);
- }
- var papa = new member("杨宏文", "男生");
- var mama = new member("黄雅玲", "女生");
- var doggy = new member("奇 奇", "宠物狗");
- papa.display();
- mama.display();
- doggy.display();
- </Script>



学习运用JavaScript脚本语言(js12)

- **事件处理程序：**
- 浏览器响应某个事件，实现用户的交互操作而进行的处理（过程）。
- **事件处理程序的调用：** 浏览器等待用户的交互操作，并在事件发生时，自动调用事件处理程序（函数），完成事件处理过程。



学习运用JavaScript脚本语言(js13)

- <FORM>

请输入基本资料:

姓名:

```
<INPUT TYPE="text" NAME="usr"  
SIZE="8">
```

```
<INPUT TYPE="button" VALUE=" 请  
单击" onClick="alert('谢谢你的填写  
...')">
```

```
</FORM>
```



学习运用JavaScript脚本语言(js14)

- 定时器:
- 用以指定在一段特定的时间后执行某段程序。
- **setTimeout():**
- 格式:

[定时器对象名=] setTimeout("<表达式>",
毫秒)

功能：执行<表达式>一次。



学习运用JavaScript脚本语言(js15)

- **setInterval():**

- 格式:

[定时器对象名=] setInterval("<表达式>", 毫秒)

功能: 重复执行<表达式>, 直至窗口、框架被关闭或执行clearInterval。

- **clearInterval():** 终止定时器

- 格式:

clearInterval(定时器对象名)



学习运用JavaScript脚本语言

- JavaScript内置对象

- 图像对象
- 导航对象
- 窗口对象
- 屏幕对象
- 事件对象
- 历史对象



学习运用JavaScript脚本语言

- JavaScript内置对象

- 文件对象（重要）

- 锚点对象

- 链接对象

- 框架对象

- 表单对象（重要）

- 位置对象



学习运用JavaScript脚本语言(js16, js17)

- 窗口对象
- 格式:
- [window.]属性
[window.]方法（参数）
- opener.属性
opener.方法（参数）

窗口名称.方法（参数）



学习运用JavaScript脚本语言

- self.属性
self.方法（参数）
parent.属性
parent.方法（参数）
top.属性
top.方法（参数）
- 窗口名称.属性



学习运用JavaScript脚本语言(js18)

- 屏幕对象：
 - 描述屏幕的显示及颜色属性。
- 屏幕对象的属性：
 - 格式： `screen.属性`



学习运用JavaScript脚本语言(js19)

- 事件对象:

当事件发生时，浏览器自动建立该对象，并包含该事件的类型、鼠标坐标等。

- 事件对象的属性:

格式: `event.属性`



学习运用JavaScript脚本语言(js20,js21组合使用)

- 历史对象:

用以存储客户端最近访问的网址清单。 格式:

- history.属性
history.方法（参数）



学习运用JavaScript脚本语言(js22)

- 位置对象： 用来代表特定窗口的URL信息。
 -

格式：

- location.属性
location.方法（参数）



学习运用JavaScript脚本语言(js23)

- 文件对象： 代表当前HTML对象，是由<body>标签组构成的，对每个HTML文件会自动建立一个文件对象。
- 格式：
- document.属性
document.方法（参数）



学习运用JavaScript脚本语言(js24)

- 链接对象：
 - 网页中的链接均会被自动看作链接对象，并依顺序，分别表示为document.links[0]，document.links[1]...
- 定义链接对象的格式：
- 字符串.link(属性)



学习运用JavaScript脚本语言(js25)

- 表单对象(最常使用, 重要):
- 文件对象的子对象, Javascript的runtime engine自动为每一个表单建立一个表单对象。
- 格式:
- document.forms[索引].属性
- document.forms[索引].方法 (参数)
- document.表单名称.属性
- document.表单名称.方法 (参数)



学习运用JavaScript脚本语言

- **Cookie对象:**
- 是一种以文件（Cookie文件）的形式保存在客户端硬盘的Cookies文件夹中的用户数据信息（Cookie数据）。Cookie文件由所访问的Web站点建立，以长久的保存客户端与Web站点间的会话数据，并且该Cookie数据只允许被所访问的Web站点进行读取。
- **Cookie文件的格式:**
- NS: Cookie.txt
- IE: 用户名@域名.txt



学习运用JavaScript脚本语言

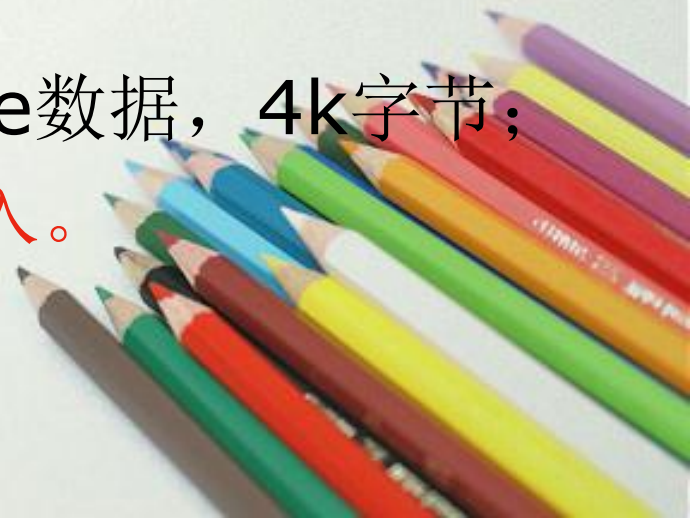
- 写入**Cookie**:
- 格式:
- `document.cookie = " 关键字 = 值 [; expires = 有效日期] [;...]"`



学习运用JavaScript脚本语言(js26)

- 备注:

- 1.有效日期格式: Wdy,DD-Mon-YY HH:MM:SS GMT
- 2.Wdy / Mon: 英文星期 / 月份;
- 3.还包含path、domain、secure属性;
- 4.每个Web站点(domain)可建立20个Cookie数据;
- 5.每个浏览器可存储300个Cookie数据, 4k字节;
- 6.客户有权禁止Cookie数据的写入。



学习运用JavaScript脚本语言

- 检查Cookie是否真正写入到客户端硬盘上
- 首先删除IE所有的Cookies
- 方法:
- 工具→Internet 选项→删除Cookies,删除文件
- 保存历史记录在:
 - C:\Documents and Settings\username\Local Settings\Temporary Internet Files(Win2000)
 - C:\ Documents and Settings\username\Cookies(XP,2003)



学习运用JavaScript脚本语言

- 为了清楚起见，我们到该目录下查看，保证其中已经没有任何文件，然后运行 **js26.html**,刷新该文件，会发现多出来一个文件



学习运用JavaScript脚本语言(js27)

- 读取**Cookie**:
 - 格式:
 - **document.cookie**



学习运用JavaScript脚本语言(js28)

- 删除**Cookie**:
- 格式:
- `document.cookie = " 关键字 = ; expires = 当前日期"`



综合实验

- 利用JavaScript与JSP进行动态页面编程
- 参见程序 begin.jsp与end.jsp



总结

- JavaScript是一个功能非常强大的脚本语言，在实际项目中得到了广泛的应用，现今很流行的Ajax(Asynchronous Javascript And Xml)，实际上就是javascript与xml及css 的综合运用

