8 JS基础

浙 江 师 范 大 学

主讲老师：徐晓丹（663601）

主要内容

- JS基础
- 应用实例

- JavaScript是由Netscape公司开发并随Navigator导航者一起发布的、介于Java与HTML之间、基于对象事件驱动的编程语言。因它的开发环境简单，不需要Java编译器，而是直接运行在Web浏览器中，因此倍受Web设计者的喜爱。

JavaScript教程语言概况

- JavaScript的出现，它可以使得信息和用户之间不仅只是一种显示和浏览的关系，而是实现了一种实时的、动态的、可交式的表达能力．
- JavaScript脚本正是满足这种需求而产生的语言。它深受广泛用户的喜爱的欢迎。它是众多脚本语言中较为优秀的一种，它与WWW的结合有效地实现了网络计算和网络计算机的蓝图。无凝Java家族将占领Internet网络的主导地位。

JavaScript几个基本特点

- 脚本编写语言
- 基于对象的语言
- 简单性
- 安全性
- 动态性
- 跨平台性

JavaScript和Java的区别

- 虽然JavaScript与Java有紧密的联系，但却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；
- 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页面中的基于对象和事件驱动的解释性语言，它的前身是Live Script；
- Java的前身是Oak语言。

基于对象和面向对象

- Java是一种真正的面向对象的语言，即使是开发简单的程序，必须设计对象。
- JavaScript是脚本语言，它可以用来制作与网络无关的，与用户交互作用的复杂软件。它是一种基于对象（Object Based）和事件驱动（Event Driver）的编程语言。因而它本身提供了非常丰富的内部对象供设计人员使用。

解释和编译

- 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。
- 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端由浏览器解释执行。

强变量和弱变量

- 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。如:

```
Integer x;
String y;
x=1234;
x=4321;
```

其中X=1234说明是一个整数，Y=4321说明是一个字符串。

- 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，如：

```
x=1234;
y = "4321";
```

前者说明x为其数值型变量，而后者说明y为字符型变量。

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。

嵌入方式不一样

- 在HTML文档中，两种编程语言的标识不同，
- JavaScript使用<Script>...</Script>来标识
- Java使用<applet>...</applet>来标识。

静态联编和动态联编

- Java采用静态联编，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。
- JavaScript采用动态联编，即JavaScript的对象引用在运行时进行检查，如不经编译则就无法实现对象引用的检查。

HTML解析过程

- 页面渲染就是浏览器将html代码根据CSS定义的规则显示在浏览器窗口中的这个过程。
- 对于一个HTML文档，浏览器的解析顺序为：按照文档流，从上到下逐步解析页面的结构。

HTML解析过程

1.用户输入网址（假设是个html页面，并且是第一次访问），浏览器向服务器发出请求，服务器返回html文件；

2. 浏览器开始载入html代码，发现<head>标签内有一个<link>标签引用外部CSS文件；

3. 浏览器又发出CSS文件的请求，服务器返回这个CSS文件；

4. 浏览器继续载入html中<body>部分的代码，开始渲染页面了；

HTML解析过程

5. 浏览器在代码中发现一个<img>标签引用了一张图片，向服务器发出请求。此时浏览器不会等到图片下载完，而是继续渲染后面的代码；

6. 服务器返回图片文件，由于图片占用了一定面积，影响了后面段落的排布，因此浏览器需要回过头来重新渲染这部分代码；

7. 浏览器发现了一个包含一行Javascript代码的<script>标签， （style.display=”none”）

8. Javascript脚本执行了这条语句，重新渲染相应部分代码。

HTML解析过程

9. 终于等到了</html>的到来。

10. 用户交互：用户点了一下界面中的“换肤”按钮，Javascript让浏览器换了一下<link>标签的CSS路径；

11. 浏览器向服务器请 求了新的CSS文件，重新渲染页面。

编写第一个JavaScript程序

```
<html>
<head>
<Script Language ="JavaScript">
// JavaScript Appears here.
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>
</Head>
 <body>
  <Script Language ="JavaScript">
     alert("大家好");
    </Script>
 </body>
</Html>
```

编写第一个JavaScript程序

- JavaScript代码由 <Script Language ="JavaScript">...</Script>说明。
- alert()是JavaScript的窗口对象方法，其功能是弹出一个具有OK对话框并显示（ ）中的字符串。
- 通过<!-- ...//-->标识说明：若不认识JavaScript代码的浏览器，则所有在其中的标识均被忽略；若认识，则执行其结果。

JavaScript代码的加入

- 可以直接将JavaScript脚本加入文档<Script Language ="JavaScript">
  JavaScript语言代码；
  JavaScript 语言代码;
  ....
  </Script>
- 说明： 通过标识<Script>...</Script>指明JavaScript脚本源代码将放入其间。
  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。

exp2

```
<body>
<h1>Current Date and Time</h1>
<script language="javascript">
now = new Date();
localtime = now.toString();
utctime = now.toGMTString();
document.write("<strong>Local time:</strong> "+localtime + "<br/>");
document.write("<strong>UTC time:</strong> "+ utctime);
</script>
</body>
```

exp3

```
<body>
<h1>welcome!</h1>
<p>
<a href="http://www.zjnu.edu.cn/"
onclick="alert('Aha!');">Go to zjnu</a>
</body>
```

JS放置位置

- 1.脚本在head标记中，定义成函数的形式，在body中调用：12-1-2
- 2.脚本位于body标记中：12-1-1
- 3.脚本位于外部JS文件中，在head中引入，在body中调用。12-1-3
- 4.直接写在事件处理的代码中。12-1-4

```
<Script Language ="JavaScript">
  function message()
  {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>
</head>
<body>
   <form action="">
      <input type="button" name="ok" onclick="message()" value="click me">
   </form>
```

JS外部文件示例1

```
<html>
<head>
 <script src="demo.js"></script>
</head>
<body>
   <form action="">
      <input type="button" name="ok" onclick="message()" value="click me">
      <input type="button" name="ck" value="welcome" onclick="alert('welcome')">
   </form>
<body>
</head>
```

JS demo.js > message

```
1   function message()
2   {
3       {alert("这是第一个JavaScript例子!");
4       alert("欢迎你进入JavaScript世界!");
5       alert("今后我们将共同学习JavaScript知识! ");}
6   }
```

外部JS文件示例2

```
<html>
<head>
<script type=” text/javascript”  src=” st.js” ></script>
</head>
<body>
<form action=” ” >
<input type="button" onclick="alert('You are clicking me!');" value=” click me” >
<input type="button" onclick= “showtime()" value=” showtime()” >
</form>
</html>
```

```
function showtime()
        {
        now= new Date();
        localTime= now.toString();
        utcTime= now.toGMTString();
        alert("local time is:" + localTime  +
          "UTC time is:" + utcTime );
        }
```

JS消息对话框

- 警告框 12-2-1

alert(message)

- 确认框 12-2-2

confirm(message)

- 提示框:12-2-3

prompt(text,defautValue)

Js注释

- 单行注释 //
- 多行注释 /*   */
- 例子：12-2-4

Js语法

- 区分大小写
- var 定义变量
- 语句用分号隔开
- 数据类型：String，number，boolean,null

Undefined,object,

例子：12-3-1

课堂练习

- 完成心理测试页面

JavaScript程序构成

- 控制语句
- 函数
- 对象
- 方法
- 属性

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。
- 12-5-2
- 12-5-3

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。
- 12-5-6
- Switch case 语句 12-5-4

For循环语句

- 基本格式
  for（初始化；条件；增量）
  语句集；
  　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。
- For　in　12-5-8

while循环

- 基本格式

  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。

  For与while语句

  两种语句都是循环语句，使用For语句在处理有关数字时更易看懂，也较紧凑；而while循环对复杂的语句效果更特别。

break和continue语句

- 与C++语言相同，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。

函数

- 函数为程序设计人员提供了一个丰常方便的能力。通常在进行一个复杂的程序设计时，总是根据所要完成的功能，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次用到的模块。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。这是与其它语言不样的地方。

常用系统函数

- Eval（ string ）:返回字符串string 表达式的值
  eval(“2+2”）　返回4
- parseFloat(string) :返回字符串对应的数值
- parseInt(string)
- isNaN(string):判断字符串是否为数值

JavaScript函数定义

- Function 函数名 （参数,变元）{
  函数体;.
  Return 表达式;
  }
  说明：
  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。
  必须使用Return将值返回。
  函数名对大小写是敏感的。
- 12-6-7：自定义求梯形面积
- 12-6-8 ：return返回计算结果

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.wrile(exp2);
  if (Number>2)
  document.write(exp3);
  if(Number>3)
  document.write(exp4);
  ...

事件驱动及事件处理

- JavaScript是基于对象(object-based)的语言。这与Java不同,Java是面向对象的语言。而基于对象的基本特征，就是采用事件驱动(event-driven)。它是在用形界面的环境下，使得一切输入变化简单化。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）。

事件处理程序

- 在JavaScript中对象事件的处理通常由函数(Function)担任。其基本格式与函数全部一样，可以将前面所介绍的所有函数作为事件处理程序。
  格式如下：
  Function 事件处理名（参数表）{
  事件处理语句集；
  ……
  }

主要有以下几个事件

- 单击事件onClick
- onChange改变事件
- 选中事件onSelect
- 获得焦点事件onFocus
- 失去焦点onBlur
- 载入文件onLoad
- 卸载文件onUnload

范例1

```
<HTML>
<HEAD>
<script Language="JavaScript">
<!--
function loadform(){
alert("这是一个自动装载例子!");
}
function unloadform(){
alert("这是一个卸载例子!");
}
//-->
</Script>
</HEAD>
<BODY OnLoad="loadform()" OnUnload="unloadform()">
<a href="test.htm">调用</a>
</BODY>
</HTML>
```

例子

- 13-1-4
- 获得焦点和失去焦点13-2-1
- 提交及重置事件 13-2-2
- 改变及选择事件 13-2-3
- 鼠标单击13-3-1
- 鼠标移动替换图片 13-3-2
- 键盘事件 13-4-1
- 综合实例 13-6-1

谢　谢　！

web开发

主讲老师：徐晓丹

浙 江 师 范 大 学