### avaScript 概述：

javascript 基于对象和事件的脚本语言

特点：

1. 安全性 （不允许直接访问本地硬盘），他可以做到信息的动态交互
2. 跨平台型（ 只要是可以解释js的浏览器都可以执行，和平台无关）

##### Javascripty与Java 不同

1. JS 是Netscape公司的产品，Java是sun公司的产品
2. JS 是基于对象，Java是面向对象
3. JS只需要解释就可以执行，Java需要先编译字节码文件在执行
4. JS 是弱语言，Java是强语（ JavaScript是没有定义数据类型，而且没有一定的严谨性）

##### JavaScript 与Html 的结合方式

想要将js代码融入到Html中，都是以标签的形式

1. Js代码存放在标签对上<script> </script>
2. 使用script标签的src属性引入一个js文件中，方便后期维护和代码的扩展
3. 在html上出现的先后位置来决定javaScript代码编译的先后，可以放在<head >,<body>中的标签类，也可以放在之间，由于<head>比<body>编译快，所以如果是公用的方法，需要被页面其他地方代码调用最好放在<head>内

###### 对于不兼容的浏览器中隐藏javaScript

1. 使用html注释的方法来注释javaScript代码，这样不支持javaScript的浏览器就不会把代码内容当做文本显示在浏览器中

<script type=*"text/javascript"*>

<!-- html comment begin

alert("ok");

// html comment end-->

</script>

2．使用javaScript标签

<noscript>

The javaScript code cat not be execute

</noscript>

浏览器限制就会打印出The javaScript code ---

### JavaScript语法

每一种语言都有自己的语法规则，Js语法与Java很想，有Java基础学JavaScript比较容易，Js有变量，语句，函数，数组等常用的语言组成元素

1. 通过var 来定义，弱语言是不用指定具体的数据类型

例： <script type=*"text/javascript"*>

**var** i = 3;

alert(i);

</script>

注：Js中特殊的常量值：undefined、当变量没有初始化就被使用，改变量的值就是undefined（未定义）

对于字符串输出时需要分号表示字符串时：

alert(‘english is "WuLiang"');

alert("english is 'WuLiang'");

JavaScript 中有 // /\* \*/ 注释方法

有两种格式

1. 直接在Html JSP页面写入，当然越在html代码内前端导入，越运行快点
2. 也可以使用封装的方法写入Js文件中

<script type=” text/javascript” src=”demo.js” />

1. 通过与java语法像

Java有条件语句 do while switch while for等

javaScript中0就假 非0为真

null false 假

switch中区别，其中java只能用bety char short int四种类型，而在javaScript中全部都可以

##### 九九乘法口诀：

<html>

<head>

<style type=*"text/css"*>

**table**{

width:*600px*;

border-collapse:*collapse*;

}

</style>

//样式表必用css 其中border-collapse 默认为separate分离线 collapse是倒塌

</head>

<body>

<script type=*"text/javascript"*>

document.write("<table border='1'>");

**for**(**var** i=1; i<=9; i++){

document.write("<tr>");

**for**(**var** j=1; j<=i; j++){

document.write("<th>"+j+"\*"+i+"="+i\*j+"</th>");

}

document.write("</tr>");

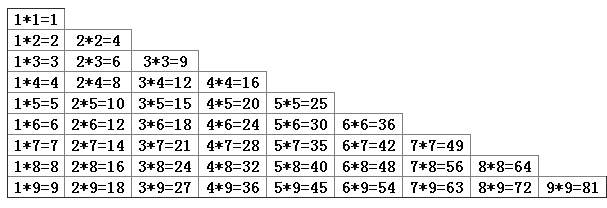
}

document.write("</table>");

</script>

</body>

</html>



##### javaScript数组:

var array = ['1',"吴亮",2,true];数组是一个集合，什么类型的数据都可以存储，而且不存在过界的原因，所以数组中的是需要可以临时加的

由于javaScript是弱语言，没有数据类型。所以数组也就不存在数据类型

特点：可以存储任意元素，长度是可变的

##### 函数：function

函数必须被调用才能运行，代码封装

function(i) 直接加变量，不需要var

1. **普通函数**

**格式： function 函数名(形式参数。。){**

**执行语句；**

**Return 返回值；**

**}**

**函数是多条执行语句的封装体，只有被调用才会执行**

**注意：调用有参数的函数，但没有传值，函数一样可以运行，或者没有参数的函数，给其传值，该函数也一样运行**

**函数得到参数，是保存在一个arguments数组对象中，该对象将其专递的都封装在一个数组中**

**function** s(x,y){

alert(arguments.length);//arguments参数，可以通过这个来看函数转入的参数

}

s(1,2,3,4);

运行为4 ，说明函数没有重载形式，函数内部有一个数组为arguments，

1. 动态函数

**通过JS的内置对象Function实现**

**var** show =**new** Function("x","y","var sum=x+y; return sum");

alert(show(1,2));

//动态new一个对象，参数与函数体都可以作为参数变量传递，可以动态指定

1. 匿名函数

**var** show = **function**(){… }

show();

***通常用于定义事件属性的行为时较为常用***

***类似java处理对象***

**function** person(){

alert("run person");

}

**var** p = **new** person();

p.name = "java";

p.age = 19;

p.eat = **function**(){

alert(p.name+" do "+p.age+" ");

};

p.eat();

**function** person(name,age){

**this**.name = name;

**this**.age = age;

}

**var** p = **new** person("jack",22);

p.eat = **function**(){

alert(p.name+"...."+p.age);

};

p.eat();

可以将其放入js文件中去，代码可以复用

作业1 写一个ArrayTool 关于数组的，求其最大值 （js文件，代码复用）

**function** a(){

}

**var** a1 = a(); 如果是 var a1 = a;

alert(a1);

1. **其中a()是给a1一个值，就是a方法中返回值，当没有，就undefined**
2. **而a就是赋给a1整个对象，就完全打印出来**

**函数注意地方：**

1. **Var menthod = show();**
2. **Var menthod = show**

**两句话都成立**

**第一句 表示show方法运行后的结果赋值给menthod变量**

**第二句 表示的是将show指向的地址赋值给menthod，那么就可以通过menthod（）方法来调用show这个方法**

###### 殊语句：with for(s in stu)

**function** person(name,age){

**this**.name = name;

**this**.age = age;

}

**var** p = **new** person("jack",34);

**with**(p){

alert(name+"..."+age);

}

With语句是一个他可以简化我们书写

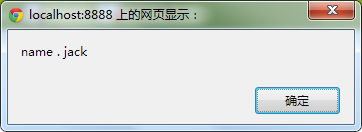
对象直接存放在with（参数）中，在该with语句中就可以直接对象的属性就可以引来调用

For in 语句可以遍历对象

**for**(s **in** p){

alert(s+" . "+p[s])

}



查看javaScript chm文件

#### 对象：

创建对象：

**var** a = **new** Object(); 创建空的对象

**var** date = **new** Date(); 创建时间对象

**var** student = {

name:"zhangSan",

age:25

}

操作对象的方法

**function** putInformation(name,age){

document.write("student.name:"+name);

document.write("<br>student.age:"+age);

}

**var** student = **new** Object();

student.information = putInformation;

student.information("wuliang", 22);

String

Math

**for**(**var** i=0; i<10; i++){

**var** m = parseInt(Math.random()\*10+1);

document.write(m+"<br/>");

}//random是和java与c等一样类型，其中parseInt中是Global对象，来转换数据类型

Global

**var** j = 9;

alert(j.toString(2));//toString是参数转换进制位数

**var** i = parseInt(110, 2);//2是进制，其中将数字转换到其进制

alert(i);

**function** getMax(){

**var** max;

**for**(**var** i=1; i<**this**.length; i++){

max = **this**[0];

**if**(max < **this**[i]){

max = **this**[i];

}

}

**return** max;

}

//var arr =getMax(array);

//alert(arr);

**Array.prototype.getM = getMax;**

**var** array = [1,23,5,232];

**var** s = array.getM();

alert(s);

**Prototype可以获取指定的对象引用，可以通过引用给已有的对象来赋予一些新的功能**

**其中自定义方法加入封装到数组中方法，prototype封装进去**

**自定义trim方法**

**function** trim(){

**var** start,end;

start = 0 ;

end = **this**.length-1;

**while**(start<end && **this**[start]==" "){

start++;

}

**while**(**this**.length-1 >end && **this**[end]==" ")

end--;

**this**.substr(start,end);

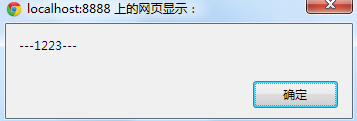
}

**var** str = " 1223 ";

Array.prototype.trim = trim;

**var** s = str.trim();

alert("---"+s+"---");



### 事件与事件处理：

DOM: document Object model ：文档对象模型

为了方便操作标记型文档，就将标记型文档中所有的标签封装成对象，标记型文档本身就是一个对象，所有的标签封装成一个对象，是可以在文档上增加属性与行为

户可以动态的增加标签，也可以删除对象

Dom 一共有三层模型

将HTML文档封装到对象

将XML文档封装成对象

将XML文档————

封装成对象，就产生了标签树，DOM树，节点

就产生了DHtml：动态html（html ，css css javascript

Html 将数据进行封装

Dom 将标签封装成对象

Css 负责标签数据的样式

Javascript 将三个进行融合，进行程序设计方式完成动态效果的操作

xmlHttpRequest –Ajax 与服务端交互的对象

例子：

<script type=*"text/javascript"*>

**var** doms = parseInt(Math.random()\*100+1);

**function** dome(){

**var** textObj = document.getElementById("textId");

**var** num = parseInt(textObj.value);

**if**(num > doms){

alert("大了，继续");

}**else if**(num == doms){

alert("相等");

}**else**{

alert("小了，继续");

}

}

</script>

<input type=*"button"* value=*"演示效果"* onclick="dome()">

<input type=*"text"* id=*"textId"*>

###### 处理错误事件：onError

Javascript 提供了error事件，图像对象处理的onError事件处理器，可以处理与加载文档相关的错误

**<script type="text/javascript" >**

**function errorHandle(errorMessage,url,line){**

**var message = "错误信息："+errorMessage+"\n错误地址："+url+"\n错误行号："+line;**

**alert(message)**

**}**

**window.onerror = errorHandle;**

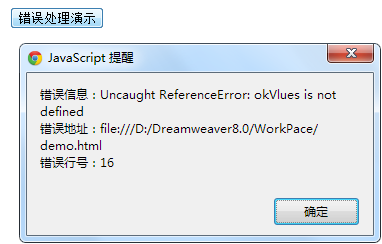
**</script>**

**</head>**

**<body>**

**<input type="button" value="错误处理演示" onclick="okVlues()" />**

**</body>**



### 窗口和框架



Window 浏览器窗口对象，显示HTML页面内容的区域 包含document

Document对象是整个Html代码 image link form layer

Form对象有许多表单数据对象组成 例如：text button radio reset submit 等等

##### Window对象

<http://www.w3school.com.cn/htmldom/dom_obj_window.asp> 看属性

##### frame对象

**创建框架，每个框架都可以看成是独立的窗口，包含自己的url，浏览器包含一个顶层window对象，而每个框架都是window对象，框架的window对象是有浏览器中顶层对象window对象继承过来的**

**例子：**

**Index.html：**

**<script type="text/javascript" >**

**function checkIsEmple(s){**

**for(var j=0; j<s.length; j++){**

**if(s.charAt(j) != ""){**

**return false;**

**}**

**}**

**return true;**

**}**

**function checkValue(){**

**if(checkIsEmple(document.forms[0].username.value)){**

**//return alert("用户名不能为空");**

**window.parent.demo.location.reload();**

**window.parent.frames[1].document.write("用户名不能为空！！！");**

**//window.parent.demo.document.write("12313");**

**}else if(checkIsEmple(document.forms[0].password.value)){**

**window.parent.frames[1].location.reload();**

**window.parent.frames[1].document.write("密码不能为空");**

**//window.parent.frames[1].document.write("sss");**

**//window.parent.demo.document.write("12313");**

**//return alert("密码不能为空");**

**}else{**

**//var w = window.open("","");**

**//w.document.write("你输入的通过验证");**

**//window.parent.frames[1].location.reload();**

**window.parent.frames[1].document.write("成功登入");**

**}**

**}**

**function clearHandle(){**

**document.test.username.value="";**

**document.test.password.value="";**

**}**

**function closeHandle(){**

**var w = window.confirm("你确定要关闭当前页面吗？");**

**if(w){**

**window.close();**

**}**

**}**

**</script>**

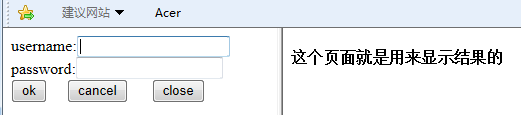
**主页面：**

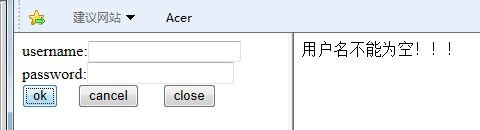
**<frameset rows="\*" cols="33%,\*">**

**<frame src="index.html" name="index" />**

**<frame src="demo.html" name="demo" />**

**</frameset><noframes></noframes>**





**注意点：**

**在浏览器Chrome和360极速5.0中，会出现错误：**

Unsafe JavaScript attempt to access frame with URL [file:///D:/Dreamweaver8.0/WorkPace/demo.html](file:///D:\Dreamweaver8.0\WorkPace\demo.html) from frame with URL [file:///D:/Dreamweaver8.0/WorkPace/index.html](file:///D:\Dreamweaver8.0\WorkPace\index.html). Domains, protocols and ports must match.

原因：**window.parent.demo.location.reload();**

**window.parent.frames[1].document.write("用户名不能为空！！！");**

　　今天测试一个脚本的时候发现，在谷歌chrome浏览器下面，iframe中获取父页的document时竟然为undefined，google chrome神奇了，其他浏览器如ie，firefox没出现这种问题。

**Iframe frames要获取到父页的document对象，需要通过服务器，就是http协议访问时才能获取到，要不直接双击运行【chrome为默认浏览器】或者直接拖进chrome浏览器查看时，iframe使用parent.document得到的是undefined。**

分帧：

要注意的地方，分帧放在html标签中head后不能再head标签内和body内

##### Location对象

Location 对象用来表示浏览器窗口中加载的文档URL ，该对象的属性说明了URL中各个部分

href 完整的url的方法

方法：

Reload（） 该方法从缓存或者从服务器中再次把当前文档加载进来，参数为boolean类型，true 无条件从所处的服务器上重新加载 ，false和默认是从上次加载完毕，文档才会重新加载

**Replace（url）用一个新的文档替换当前的文档**

与属性href不同，调用replace方法是，指定url会替换浏览器历史列表中的当前url，而不是创建一个新的条目，所以用replace方法加载一个新的页面，浏览器中“后退”将无法后退，而通过href可以

例子：

function changNewHref(){

**var url = window.parent.frames[0].document.forms[0].newUrl.value;**

**//window.parent.frames[1].location.href = url;**

**window.parent.frames[1].location.replace(url);**

}

##### History 对象

History对象是一个数组，其中元素存储了浏览历史中url，用来维护web浏览器的当前会话内所有打开的历史文件列表

History 对象三个方

Back（） forward（） go（）

function changHref(){

var url = window.parent.frames[0].document.test.oldUrl.value;

window.parent.frames[1].location.href =url;

}

function goHistory(h){

if(h == 1){

***window.parent.frames[1].history.back();***

}else if(h==2){

**window.parent.frames[1].history.forward();**

}else{

//**window.parent.history.go("http://www.baidu.com");**

var urlValue =window.parent.frames[0].document.forms[0].oldUrl.value;

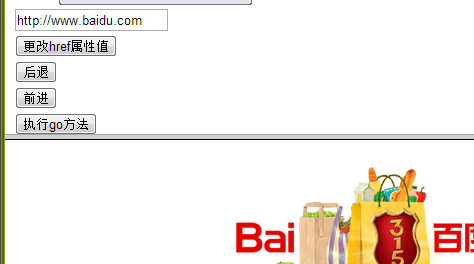
alert(urlValue)

window.parent.history.go(urlValue);

//不能使用window.parent.frames[1].history.go(urlValue)

}

}



##### Navigator 对象

Navigator航海员 该对象是用于获取用户浏览器的相关信息

属性：

appName：改属性用于表示web浏览器名称

appVersion：浏览器的版本号

userAgent：存储http用户代理请求头中的字符串，包含了appName appversion所有信息

platform 运行浏览器的平台

language 语言版本

javaEnabled（） 是否支持并激活java

function runGavigator(){

document.write("浏览器名称："+navigator.appName+"<br>");

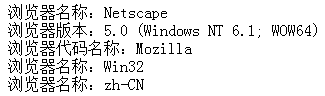
document.write("浏览器版本："+navigator.appVersion+"<br>");

document.write("浏览器代码名称："+navigator.appCodeName+"<br>");

document.write("浏览器名称："+window.navigator.platform+"<br>");

document.write("浏览器名称："+window.navigator.language+"<br>");

}



##### Form 表单数据：

<form name="formName" //表单名

[target="windowName"] //用于响应窗口名称

[action="url"] //表单提交目的地地址

[method="encoding"] //编码格式

[onReset="handler"] //表单重置时间处理器

[onSubmit="handler"] > //提交

</form>

表单对象是Web设计中最常用的也是最重要之一

Document.forms[index]

Document.forms[“fornName”]

Document.forms.fomName

Document.fornName

form对象的属性和方法

action 提交表单的地址

***element[]：该属性为一个数组：它们：button text password hidden checkbox reset submit radio select（option） textarea***

elements.length

例子：

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title></title>

<script type="text/javascript" >

新开一个窗口，其dom赋值与d

function openWin(){

**var w = window.open("","");**

**var d = w.document;**

d.write('<html><head>');

d.write('<title>form表单的数据</title>');

d.write('</head><body>');

d.write("原页面含有"+document.forms.length+"form对象<br>&nbsp;&nbsp;对象的名称和值为");

var forms = document.forms[0].elements.length;

**for**(var i=0; i<document.isForm.elements.length; i++){

d.write("名称："+document.isForm.elements[i].name+"&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;值："+document.isForm.elements[i].value+"<br/>");

}

d.write('<body><html>')

}

</script>

</head>

<body>

<form name="isForm" >

<h4>请输入您的基本信息</h4>

你的姓名：<input type="text" name="username" value="请输入你的用户名，在1-12为内" width="20"/><br />

你的性别：<input type="radio" name="man" />&nbsp;&nbsp;&nbsp;&nbsp;

<input type="radio" name="feman" /><br />

你的年龄：<input type="text" name="age" width="20"/><br />

工作单位：<input type="text" name="work" width="20" /><br />

联系方式：<input type="text" name="contact" width="20" /><br />

<h4>请输入您的其他信息（选择性）</h4>

你向往的职业：<select name="profession" >

<option value="教师" >教师</option>

<option value="软件工程师">软件工程师</option>

<option value="公务员">公务员</option>

</select><br />

您喜欢的运动：

<input type="checkbox" name="football" value="foorball" />足球&nbsp;&nbsp;

<input type="checkbox" name="basketball" value="basketbll" />篮球&nbsp;&nbsp;

<input type="checkbox" name="vollyball" value="vollyball" />排球<br />

<input type="checkbox" name="badmintoon" value="badmintoon" />羽毛&nbsp;&nbsp; <input type="checkbox" name="pingpong" value="pingpong" />乒乓球<br />

求给我们留言：<br />&nbsp;&nbsp;&nbsp;&nbsp;&nsbp;

<textarea name="content" rows=5 cols=15 ></textarea><br /><br />

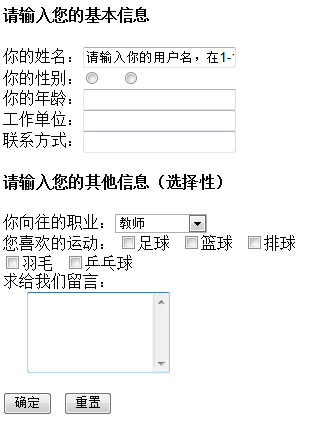
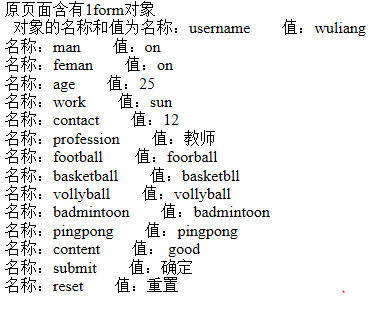
<input type="submit" name="submit" value="确定" onclick="openWin()" />&nbsp;&nbsp;

<input type="reset" name="reset" value="重置" />

</form>

</body>

</html>

Form元素中的元素对象：

Button 按钮（多个按钮同时只有一个有效） type，name，value blur() focus()

Submit 提交 reset 重置 type，name，value blur() focus()

**Submit 和reset 都可以被 document.form[0].submit() 或 reset()的方法创建**

**Text 对象**

**Textarea 多行文本 cols rows**

**Password**

**共同属性：defaultValue name type value size maxLength**

**Blur() focus() select() 该方法可以选中这对象的所有内容**