2017.6.28

一：

|----DOM和BOM

BOM是浏览器对象模型,由一系列对象构成,核心对象为window.

DOM是文档对象模型,由一系列对象构成,核心对象为document.

注意:DOM已经作为浏览器的一个标准.BOM尚未作为标准,每个浏览器都有自己的BOM实现.



|----Window对象

Window表示浏览器中打开的窗口.

常用方法和属性:







|----Document对象:

Document包含HTML文档的信息,表示文档对象.通过Document对象,访问HTML页面中的所有元素.

常用方法:

write() 或writeln():向文档中写入指定内容

节点操作:

节点信息:

节点对象.nodeName/nodeType/nodeValue: 获得节点名/类型/文本内容

注意:

节点类型.[1表示element[元素/标签],2表示attribute[属性],3表示text[文本]

层次关系属性:

标签对象.parentNode[parentElement(IE支持)]/firstChild/lastChild/childNodes: 获得当前标签的父节点/第一个子节点/最后一个子节点/所有子节点

引用节点:

document.getElementById("id值");

document.getElementsByName("name值");

document.getElementsByTagName("标签名");

创建节点:

document.createElement("节点名");

被复制标签对象.cloneNode(是否克隆所有后代)

添加节点:

父节点对象.appendChild(子节点对象)

父节点对象.insertBefore(子节点对象,原节点对象)

修改节点:

父节点对象.replaceChild(新节点对象,旧节点对象)

删除节点:

父节点对象.removeChild(子节点对象).

子节点对象.parentNode.removeChild(子节点对象)

设置/获得标签属性,标签对之间的内容

标签对象.属性名 获得标签对象的某属性值

标签对象.属性名=值 设置标签对象的某属性值

标签对象.getAttribute(“属性名"); 获得标签对象的某属性值

标签对象.setAttribute(“属性名","属性值"); 设置标签对象的某属性值

标签对象.innerHTML=值 或 标签对象.innerHTML : 设置或获得标签对之间的所有内容[html标签和文本]

标签对象.innerText=值 或 标签对象.innerText: 设置或获得标签对之间的文字[不含标签]

操作CSS:[设置/获得标签的CSS样式]

操作标签的style属性值:[操作行内式的CSS代码,行内式优先级最高,操作最准确]

标签对象.style.cssText 获得标签style属性的属性值

标签对象.style.样式属性名; 获得标签style属性值中指定样式的值,如果没有编写该样式,将返回空.

标签对象.style.样式属性名=值; 设置标签style属性值中指定样式的值

注意:CSS中由多个单词和中划线组成的样式名,在JS中省掉中划线,同时中划线后的单词首字母大写.

CSS样式名:background-image JS代码: 标签对象.style.backgroundColor="blue"

操作style标签及外部样式代码:[操作内嵌式,链接式的CSS代码]

获得所有style标签: document.styleSheets; 返回值为 存放style标签对象的数组

获取style对象: document.styleSheets[索引].rules[索引].style

获得第一个style标签中的第一个选择器的style对象: document.styleSheets[0].rules[0].style

获得style对象中的所有属性内容: style对象.cssText

获得style对象中指定样式的值: style对象.样式属性="值" 如果没有编写该样式,将返回空.

|-----Screen对象

Screen对象存储了有关客户端显示屏幕的信息.

常用方法和属性:



|----Location对象

Location对象存储了有关当前URL的信息.

常用方法和属性:



|-----Navigator对象

Navigation对象存储了有关浏览器的信息.

常用方法和属性:



|-----History对象

history对象,用于存储客户端最近访问过的网址.

常用方法和属性:



注意:

go(-3)相当于连续点击3次后退按钮.

window.html

<script type="text/javascript">

function test1(){

//弹出警告框: 只含关闭按钮 和 确定按钮

//window.alert("我是警告框!");

//alert("我是警告框!");

//alert("你想死吗!");

//弹出提醒窗口: 含有关闭按钮 和 确定 取消按钮. 当点击确定时,返回true. 否则[取消/关闭]返回false.

/\*var rs=window.confirm("您确定要删除吗?");

if(rs){

alert("删除成功!");

}else{

alert("暂不删除!");

}\*/

//弹出交互窗口:

/\*var rs=window.prompt("请输入您的性别:","男");

alert(rs);\*/

//弹出窗口 open(网页URL,窗口名,窗口特征)

window.open("pageForWindow.html","","width=500px,height=300px,left=300px,top=100px");

}

function temp(){

alert("我是temp()!");

}

function test2(){

//创建一个定时器

//timeout=window.setTimeout("temp()",3000);

timeout=window.setInterval("temp()",1000);

}

function test3(){

//window.clearTimeout(timeout);

window.clearInterval(timeout);

}

</script>

</head>

<body>

Window对象测试 <br>

<!-- 弹出窗口和关闭窗口 -->

<input type="button" value="窗口" onclick="test1()">

<!-- 延迟或定时执行 -->

<input type="button" value="延迟执行" onclick="test2()">

<!-- 取消延迟或定时执行 -->

<input type="button" value="取消执行" onclick="test3()">

</body>

</html>

newPage.html

<script type="text/javascript">

function test1(){

window.close();

}

function test2(){

//获得父窗口的地址栏对象,更改地址并连接

window.opener.location.assign("http://www.baidu.com");

//关闭当前窗口

window.close();

}

function test3(){

//调整窗口大小: resizeBy()增加XX像素 resizeTo()调整为XX像素

//window.resizeBy(-100,-100);

window.resizeTo(300,300);

//移动窗口: moveBy() moveto()

window.moveBy(-50,-10);

}

</script>

</head>

<body>

操作成功!

<!-- 关闭当前窗口 -->

<input type="button" value="关闭窗口" onclick="test1()">

<!-- 改变父窗口信息 -->

<input type="button" value="关闭窗口[改变父窗口]" onclick="test2()">

<!-- 调整窗口 -->

<input type="button" value="调整窗口" onclick="test3()">

</body>

</html>

二:

Screen对象测试

<script type="text/javascript">

function test(){

document.write("显示器分辨率: "+screen.width + "\*" + screen.height+"<br />");

document.write("桌面尺寸: "+screen.availWidth + "\*" + screen.availHeight+"<br />");

document.write("目标设置或缓冲器 调色板比特深度: "+screen.colorDepth+"<br />")

document.write("调色板比特深度: "+screen.bufferDepth+"<br />")

document.write("每英寸点数\_X: "+screen.deviceXDPI+"<br />")

document.write("每英寸点数\_Y: "+screen.deviceYDPI+"<br />")

document.write("常规点数\_X: "+screen.logicalXDPI+"<br />")

document.write("常规点数\_Y: "+screen.logicalYDPI+"<br />")

document.write("是否启动字体平滑: "+screen.fontSmoothingEnabled+"<br />")

document.write("颜色分辨率: "+screen.pixelDepth+"<br />")

document.write("屏幕刷新率: "+screen.updateInterval+"<br />")

}

</script>

</head>

<body>

Screen对象测试<br>

<input type="button" value="确定" onclick="test()">

</body>

</html>

居中窗口

<script type="text/javascript">

function test(){

//获得显示器的宽度

var sw=screen.width;

//获得显示器的高度

var sh=screen.height;

//设置窗口宽高

var w=500;

var h=400;

//计算中间位置

var cw=(sw-w)/2;

var ch=(sh-h)/2

//"width="+w+"px,height="+h+"px,top="+ch+"px,left="+cw+"px"

window.open("pageForScreen.html","","width="+w+"px,height="+h+"px,top="+ch+"px,left="+cw+"px");

}

</script>

</head>

<body>

Screen对象使用<br>

<input type="button" value="居中窗口" onclick="test()"><br>

</body>

</html>

三:

历史对象

one

<body>

History\_one.html页面!<br>

<a href="History\_two.html">History\_two.html</a>

</body>

</html>

two

<script type="text/javascript">

function test1(){

//history.go();//相当浏览器的刷新按钮

history.go(-1);//1 点击1次 前进按钮 3点击3次前进 -1点击一次后退

}

function test2(){

history.forward();

}

function test3(){

history.back();

}

</script>

</head>

<body>

<!--

one---two---three: two进入three[前进] 回到one[后退]

-->

History\_two.html页面!<br>

<input type="button" value="刷新" onclick="test1()"><br>

<input type="button" value="前进" onclick="test2()"><br>

<input type="button" value="后退" onclick="test3()"><br>

<a href="History\_three.html">History\_three.html</a>

</body>

</html>

three

<body>

History\_three.html页面!<br>

<a href="History\_one.html">History\_one.html</a>

</body>

</html>

四:

Navigator对象

<script type="text/javascript">

function test(){

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

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

document.write("<p>代码："+navigator.appCodeName + "</p>");

document.write("<p>平台："+navigator.platform + "</p>");

document.write("<p>Cookies 启用："+navigator.cookieEnabled + "</p>");

document.write("<p>插件数："+navigator.plugins.length + "</p>");

}

</script>

</head>

<body>

Navigator对象测试<br>

<input type="button" value="确定" onclick="test()">

</body>

</html>

五:

Location对象

<script type="text/javascript">

function test(){

//href实质可以理解为 超链接a的href

/\*var nowHref=location.href;

alert("当前url:"+nowHref);\*/

//location.href="pageForLocation.html";

//location.reload();//刷新

//location.replace("pageForLocation.html");//在当前历史记录的基础上,替换当前页面内容为指定网页内容

location.assign("pageForLocation.html");//实质相当于 超链接

}

</script>

</head>

<body>

Location对象测试<br>

<input type="button" value="确定" onclick="test()"><br>

<a href="pageForLocation.html">pageForLocation.html</a>

</body>

</html>