

1. `document.write(" ");` 输出语句
2. JS 中的注释为//
3. 传统的 HTML 文档顺序是:document->html->(head,body)
4. 一个浏览器窗口中的 DOM 顺序是:window->(navigator,screen,history,location,document)
5. 得到表单中元素的名称和值:`document.getElementById("表单中元素的 ID 號 ").name(或 value)`
6. 一个小写转大写的 JS: `document.getElementById("output ").value = document.getElementById("input ").value.toUpperCase();`
7. JS 中的值类型:String,Number,Boolean,Null,Object,Function
8. JS 中的字符型转换成数值型:`parseInt(),parseFloat()`
9. JS 中的数字转换成字符型:(" " 变量)
10. JS 中的取字符串长度是:(length)
11. JS 中的字符与字符相连接使用號.
12. JS 中的比较操作符有:==等于,!!=不等于,>,>=,<,<=
13. JS 中声明变量使用:var 来进行声明
14. JS 中的判定语句结构:`if(condition){}else{}`
15. JS 中的循环结构:`for([initial expression];[condition];[upadte expression]) {inside loop}`
16. 循环中止的命令是:break
17. JS 中的函数定义:`function functionName([parameter],...){statement[s]}`
18. 当文件中出现多个 form 表单时.可以用 `document.forms[0],document.forms[1]`来代替.
19. 窗口:打开窗口 `window.open()`, 关闭一个窗口:`window.close()`, 窗口本身:self
20. 状态栏的设置:`window.status= "字符 "`;
21. 弹出提示信息:`window.alert("字符 ");`
22. 弹出确认框:`window.confirm();`
23. 弹出输入提示框:`window.prompt();`
24. 指定当前显示链接的位置:`window.location.href= "URL "`
25. 取出窗体中的所有表单的数量:`document.forms.length`
26. 关闭文档的输出流:`document.close();`
27. 字符串追加连接符: =
28. 创建一个文档元素:`document.createElement(),document.createTextNode()`
29. 得到元素的方法:`document.getElementById()`
30. 设置表单中所有文本型的成员的值为空:

```
var form = window.document.forms[0]
for (var i = 0; i <form.elements.length;i ){
if (form.elements.type == "text"){
form.elements.value = " ";
}
}
```
31. 复选按钮在 JS 中判定是否选中:`document.forms[0].checkThis.checked` (checked 属性代表为是否选中返回 TRUE 或 FALSE)
32. 单选按钮组 (单选按钮的名称必须相同): 取单选按钮组的长度
`document.forms[0].groupName.length`
33. 单选按钮组判定是否被选中也是用 checked.
34. 下拉列表框的值:`document.forms[0].selectName.options[n].value` (n 有时用下拉列表框名称加上.selectedIndex 来确定被选中的值)
35. 字符串的定义:`var myString = new String("This is lightsword ");`
36. 字符串转成大写:`string.toUpperCase();` 字符串转成小写:`string.toLowerCase();`
37. 返回字符串 2 在字符串 1 中出现的位置:`String1.indexOf("String2 ")!= -1` 则说明没找到.
38. 取字符串中指定位置的一个字符:`StringA.charAt(9);`

39.取出字符串中指定起点和终点的子字符串:stringA.substring(2,6);
40.数学函数:Math.PI(返回圆周率),Math.SQRT2(返回开方),Math.max(value1,value2)返回两个数中的最大值,Math.pow(value1,10)返回

value1 的十次方,Math.round(value1)四舍五入函数,Math.floor(Math.random()*(n+1))返回随机数

41.定义日期型变量:var today = new Date();

42.日期函数列表:dateObj.getTime()得到时间,dateObj.getYear()得到年份,dateObj.getFullYear()得到四位的年份,dateObj.getMonth()得

到月份,dateObj.getDate()得到日,dateObj.getDay()得到日期几,dateObj.getHours()得到小时,dateObj.getMinutes()得到

分,dateObj.getSeconds()得到秒,dateObj.setTime(value)设置时间,dateObj.setYear(val)设置年,dateObj.setMonth(val)设置

月,dateObj.setDate(val)设置日,dateObj.setDay(val)设置星期几,dateObj.setHours 设置小时,dateObj.setMinutes(val)设置

分,dateObj.setSeconds(val)设置秒 [注重:此日期时间从 0 开始计]

43.FRAME 的表示方式: [window.]frames[n].ObjFuncVarName,frames["frameName"].ObjFuncVarName,frameName.ObjFuncVarName

44.parent 代表父亲对象,top 代表最顶端对象

45.打开子窗口的父窗口为:opener

46.表示当前所属的位置:this

47.当在超链接中调用 JS 函数时用:(javascript :)来开头后面加函数名

48.在老的浏览器中不执行此 JS: <!-- //-->

49.引用一个文件式的 JS: <script type="text/javascript" src="aaa.js"></script>

50.指定在不支持脚本的浏览器显示的 HTML: <noscript></noscript>

51.当超链和 onCLICK 事件都有时,则老版本的浏览器转向 a.html,否则转向 b.html.例: dfsadf

52.JS 的内建对象

有

:Array,Boolean,Date,Error,EvalError,Function,Math,Number,Object,RangeError,ReferenceError,RegExp,String,SyntaxError,TypeError

or,URIError

53.JS 中的换行:\n

54.窗口全屏大小: <script>function fullScreen(){ this.moveTo

(0,0);this.outerWidth=screen.availWidth;this.outerHeight=screen.availHeight;}window.maximize=fullScreen; </script>

55.JS 中的 all 代表其下层的全部元素

56.JS 中的焦点顺序:document.getElementById("表元素 ").tabIndex = 1

57.innerHTML 的值是表元素的值:如 <p id="para">"how are you "</p>,则 innerHTML 的值就是:how are you

2. 58.innerTEXT 的值和上面的一样,只不过不会把 这种标记显示出来.

59.contentEditable 可设置元素是否可被修改,isContentEditable 返回是否可修改的状态.

60.isDisabled 判定是否为禁止状态.disabled 设置禁止状态
61.length 取得长度,返回整型数值
62.addBehavior()是一种 JS 调用的外部函数文件其扩展名为.htc
63.window.focus()使当前的窗口在所有窗口之前.
64.blur()指失去焦点.与 FOCUS()相反.
65.select()指元素为选中状态.
66.防止用户对文本框中输入文本:
67.取出该元素在页面中出现的数量:document.all.tags("div(或其它 HTML 标记符) ").length
68.JS 中分为两种窗体输出:模态和非模态.window.showModalDialog(),window.showModeless()
69. 状态栏文字的 设置 :window.status= ' 文字 '; 默认 的状态栏文字 设置 :window.defaultStatus = '文字. ';
70.添加到收藏夹:external.AddFavorite("http://www.dannyg.com ";, "jaskdlf");
71.JS 中碰到脚本错误时不做任何操作:window.onerror = doNothing; 指定错误句柄的语法为:window.onerror = handleError;
72.JS 中指定当前打开窗口的父窗口:window.opener,支持 opener.opener...的多重继续.
73.JS 中的 self 指的是当前的窗口
74.JS 中状态栏显示内容:window.status= "内容 "
75.JS 中的 top 指的是框架集中最顶层的框架
76.JS 中关闭当前的窗口:window.close();
77.JS 中提出是否确认的框:if(confirm("Are you sure? ")){alert("ok ");}else{alert("Not Ok ");}
78.JS 中的窗口重定向:window.navigate("http://www.sina.com.cn ");
79.JS 中的打印:window.print()
80.JS 中的提示输入框:window.prompt("message ", "defaultReply ");
81.JS 中的窗口滚动条:window.scroll(x,y)
82.JS 中的窗口滚动到位置:window.scrollby
83.JS 中 设置 时间 间隔 :setInterval("expr ",msecDelay) 或 setInterval(funcRef,msecDelay) 或 setTimeout
84.JS 中的模态显示在 IE4 行,在 NN 中不行:showModalDialog("URL "[,arguments][,features]);
85.JS 中的退出之前使用的句柄:function verifyClose(){event.returnValue= "we really like you and hope you will stay longer. "};}

window.=verifyClose;

86.当窗体第一次调用时使用的文件句柄:onload()
87.当窗体关闭时调用的文件句柄:onunload()
88.window.location 的 属 性 :
protocol(http:),hostname(www.example.com),port(80),host(www.example.com:80),pathname ("/a/a.html "),hash("#giantGizmo ",指跳转到相应的锚记),href(全部的信息)
89.window.location.reload()刷新当前页面.
90.window.history.back()返回上一页,window.history.forward()返回下一页,window.history.go(返回第几页,也可以使用访问过的 URL)
91.document.write()不换行的输出,document.writeln()换行输出
92.document.body.noWrap=true;防止链接文字折行.
93.变量名.charAt(第几位),取该变量的第几位的字符.
94. "abc ".charCodeAt(第几个),返回第几个字符的 ASCII 码值.
95.字符串连接:string.concat(string2),或用 +=进行连接
96.变量.indexOf("字符 ",起始位置),返回第一个出现的位置(从 0 开始计算)
97.string.lastIndexOf(searchString[,startIndex])最后一次出现的位置.
98.string.match(regExpression),判定字符是否匹配.

- 99.string.replace(regExpression,replaceString)替换现有字符串.
- 100.string.split(分隔符)返回一个数组存储值.
- 101.string.substr(start[,length])取从第几位到指定长度的字符串.
- 102.string.toLowerCase()使字符串全部变为小写.
- 103.string.toUpperCase()使全部字符变为大写.
- 104.parseInt(string[,radix(代表进制)])强制转换成整型.
- 105.parseFloat(string[,radix])强制转换成浮点型.
- 106.isNaN(变量):测试是否为数值型.
- 107.定义常量的要害字:const,定义变量的要害字:var