

1. `document.write(" ");` 输出语句
`out.print()`接着在页面上打印
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("表单元素 ").tabIndex = 1

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

1. 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.状态栏文字的設置: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.dannyyg.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

1..提取表单元的方法:

(1) 为各个元素设置 ID, 用 document.getElementById("****").value 提取。

(2) 为表单设置 name: form, 为元素设置 name:username,用 document.form.username.value 来提取。

例子: if(document.usermsg.passwd.value.trim == document.usermsg.passwd2.value.trim){
 alert("两个密码一致");
 }else{
 alert("密码不一致");
 }

2.JavaScript 中参数传递

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

```
function demo(obj){
```

```
    var username = obj.username.value;
```

```

    if(username==" " || usenrmae==null){
        alert('请输入用户名');
        return false;//不提交
    }else{
        return true;//提交
    }
}
</script>
<body>
    This is my HTML page. <br>
    <form action="index.jsp" onsubmit="return demo(this);">
        用户名:<input type="text" name="username"><br>

        <input type="submit">

    </form>

```

3.得到单选按钮组（radio）中选中的值

```

javaScript:
    function check(){
        var r = document.getElementsByName("radio");
        for(var i = 0;i < r.length;i++){
            if(r[i].checked){
                alert(r[i].value);
            }
        }
    }

```

即使是复选框也可以用一些同样的 name 然后用上面函数去判断。

4.javaScript 中的一些事件。

（1）onload,onunload

eg: <body onload="demo()" onunload="alert('bye-bye')">

```

    function demo(){
        alert('welcome');}

```

（2）onmouseover,onmouseout

eg:<input type="button" value="提交"

```

    onmouseover="demo(this,'tijiao')"
    onmouseout="this.value='submit'">

    function demo(obj,val){

```

```
obj.value=val;
}
```

(3)onmouseup,onmousedown

eg: `<input type="button" value="click"`

`onmouseup="this.height=100" onmousedown="this.height=200">` --按钮按下的松开按钮的高度变化。

eg:当得到焦点时,文本框中的文字消失 -- onFocus

```
<input type="text" value="请在此处输入" name="te"
tabindex="2" onFocus="this.value="">
```

eg: 当选中文字时,文字内容发生变化 -- onselect

```
<input type="text" value="在此处输入文本" onselect="this.value='不可拷贝'">
```

eg:当选择改变时, alert 出选择内容 --onchange

```
<select onchange="alert(this.value)">
  <option value="北京">北京</option>
  <option value="上海">上海</option>
  <option value="广东">广东</option>
</select>
```

eg:得到焦点、失去焦点,按钮上文字发生变化 -- onFocus、onblur

```
<input type="button" value="聚焦和失去聚焦"
onfocus="this.value='聚焦'" onblur="this.value='失去聚焦'" tabindex="3">
```

eg:打开新窗口

```
<input type="button" value="打开新页面"
onclick="window.open('D.html','mywindow',
'height:200;width:300;toolbar=no;menubar=no;status=no;resizable=yes')">
<!-- toolbar:工具栏, menubar: 菜单栏,status:状态栏, -->
```

eg:关闭窗口

```
windows.close();
```

(4)location 转向

location.href 属性<input type="button" value="部门管理"

```
onclick="location.href='E.html'">
```

```
<br>
```

location.replace 方法<input type="button" value="部门管理"

```
onclick="location.replace('E.html')">
```

(5) 分别用隐含数组和表单名称取出文本框的内容

```
<input type="button" value="用隐函数组取表单一的内容"
onclick="alert(document.forms[0].username.value)">
<input type="button" value="用表单名取表单一的内容"
onclick="alert(document.a.username.value)">
```

(6) 提交表单内容的两种方式：使用 submit 按钮和使用 ‘submit()’ 方法

```
<input type="submit" value="用 submit 按钮提交">
<input type="button" value="用 submit()方法提交" onclick="document.b.submit()">
```

(7) 点击隐藏 div 中的内容

```
<div id="mydiv"><br><!-- style="display:none" -->
  This is my HTML page. <br>
</div>
```

```
<input type="button" value="隐藏" onclick="document.all('mydiv').style.display='none'">
```

(8) 点击显示 div 中的内容

```
<input type="button" value="点击" onclick="demo()">
<script type="text/javascript">
  function demo(){
    var v="<font color='red'>非常好</font>";
    document.all('mydiv').innerHTML=v;
  }
</script>
```

(9) 点击一次显示 div 中的内容，再次点击将隐藏

```
<div id="message" style="display:none;">
  <pre>  --pre 是保持文本原来的格式
  面朝大海
  春暖花开
  </pre>
</div>
<input type="button" value="显示"
onmousedown="document.all('message').style.display=""
onmouseup="document.all('message').style.display='none'"
>
```

(10) 在 javascript 中直接使用 document.write() 可以实现在页面上输入。

```
<script type="text/javascript">
  alert('欢迎使用系统');
  document.write('熊小心');

  prompt('输入生肖','虎');
```

```

prompt('输入家庭住址','XXXXXXX');
prompt('输入电话','XXXXXXX');
prompt('输入银行卡号码','XXXXXXX');
prompt('输入密码','XXXXXXXXXX');
confirm('确认输入? ');
document.write('所有信息被成功盗取');

```

```
</script>
```

(11) javascript 中定义数组

//JavaScript 是弱类型语言，以下可见，但是在自己写的时候最好将其规范化

```

var names = new Array();
names[0] = "MUTOMBE";
names[1] = 11;
names[2] = false;
names[4] = "jot";

for(var i = 0; i < names.length; i++){
    alert("names["+i+"]: "+names[i]);
}

```

(12) javascript 中参数传递

```

function demo(name){
return "你的姓名为:"+name;
}

```

```

v_res = demo('YAO');
document.write(v_res);

```

(13) 跑马灯效果

```

<marquee behavior=scroll scrollamount="1" scrolldelay="1">
    <a href="A.html" target="_blank">This is my HTML page. <br></a>
</marquee>

```

(14) 鼠标悬停、移除按钮颜色变化: onmouseover、onmouseout

```

<input type="button" value="函数" onclick="sum(10);"
    onmouseover="this.style.color='red'"
    onmouseout="this.style.color='black'">

```

(15) <div style="display: "" id="divs">隐藏</div>


```
<input type="button" name="but" value="hidden"
onclick="document.all('divs').style.display='none'">
<input type="button" name="but" value="display"
onclick="document.all('divs').style.display=''">
```

javascript:脚本语言

javascript 可以依赖于 html, jsp 等等
javascript 可以嵌在页面中的任何地方

1. javascript 可以放在 script 节点中间，自动被执行

```
<script>
    alert("欢迎使用本系统");
    document.write('加油')
</script>
```

也可以由函数来调用

```
<input type="button" value="点击" onclick="alert('大家好')">
```

2. javascript 语句的声明

```
<script>
    alert("欢迎使用本系统");
    document.write('加油')
</script>
```

3. 常用的几个输入提示方法

这些方法包括：

alert()警告对话框

confirm（）表示确认选择对话框

prompt（）提示输入框

4. 变量的声明

所有变量均以 var 来声明

使用时再根据数据的类型来确定其变量的类型

以下的写法是错误的

```
String name="zhangshan";
```

应该写为：

```
var name = "zhangshan";
```

5.运算符

三大类：

数字运算：

比较运算

逻辑运算

6. javascript 构成

变量

函数

语句块

等等

7. 在 javascript 中使用判断语句

```
if(boolean){
```

```
}else if(boolean){
```

```
}else{
```

```
}
```

其语法规则和 java 代码非常相似

8. 循环语句：

三类

for

```
for(var i = 0;i<20;i++){  
    if(i==5){  
        continue;  
    }  
}
```

```

        if(i==9){
            break;
        }
        alert(i);

    }

while
var i = 0;

    while(i<10){
        i++;
        alert(i);

    }

```

在循环里，我们也可以使用 **break** 和 **continue**、语句

javascript 是一种弱类型
java 就是一种强类型

9, javascript 数组

A。数组：在 java 中，数组是相同类型的数据集合
在 javascript 中，可以不是相同类型

```

var names= new Array(5);
    names[0] = "张三";
    names[1] = 123;

```

上面的语句完全是合法的

B。两种方式：

第一种方式：

```

//相当于 java 数组的动态初始化
var names= new Array(5);
names[0] = "张三";
names[1] = 123;
//    alert(names[0]);

```

```
for(var i=0;i<names.length;i++){  
    alert(names[i]);  
}
```

第二种方式:

```
//本方法相当于 java 代码中的数组静态初始化  
var names= new Array("smith","allen","clark");  
for(var i = 0;i<names.length;i++){  
    alert(names[i]);  
}
```

共同的属性 `length`，表示数组的长度

10. 函数:

函数就是一个实现某个功能的方法

语法规则:

```
function demo(age){  
    if(age>20){  
        return "123";  
    }else{  
        return true;  
    }  
}
```

注意下面的写法，是不正确的

```
function var demo(var age){  
  
  
}
```

在 javascript 中，返回值类型不需要声明，参数类型不需要声明

参数定义:

形式参数可以定义多个

但实际参数可以是 0 个，一个或多个

也就是说：调用函数的时候，不必为每一个参数指定一个值

```
function demo(name,age,mail,address,phone){
    //代表实参的个数
    alert(arguments.length);

}
//注意:javascript 允许传递不同的参数个数和类型
demo('zhangshanfeng',28);
```

通过 arguments.length 可以查看到实际参数的个数

函数的调用:

两种方式:

A.在 javascript 直接来调用

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

```
function demo(){
```

```
    alert("我被调用了");
```

```
}
```

```
//在 javascript 代码块中直接调用
```

```
demo();
```

```
demo();
```

```
demo();
```

```
</script>
```

B.通过事件来驱动

```
<input type="button" value="点击" onclick="demo()">
```

11.事件驱动

事件:动作

javascript 本身是一种事件驱动型语言,它支持通过事件进行程序的处理

事件处理的两种方式:

A.在事件处理函数中,直接编写 javascript 代码

B.在事件处理函数中,来调用一个 javascript 函数

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

```
function demo(){
```

```
    alert('abc');
```


请注意:

在 onsubmit 函数中,如果 onsubmit 的值为 return true;表示表单提交
如果 onsubmit 的值为 return false;表示表单不提交

实例:

```
<script type="text/javascript">
function demo(obj){
    var username = obj.username.value;
    if(username=="" || username==null){
        alert('请输入用户名');
        return false;//不提交
    }else{
        return true;//提交
    }
}
</script>
<body>
    This is my HTML page. <br>
    <form action="index.jsp" onsubmit="return demo(this);">
        用户名:<input type="text" name="username"><br>

        <input type="submit">

    </form>
```

12.window 窗口对象

alert:警告

confirm:选择

prompt:输入

close:关闭

print:打印

open:打开新页

window.open 用法

```
<input type="button" value="部门管理" onclick="window.open('E.html','mywindow',
    'height=200,width=200,toolbar=no,menubar=no,status=no,resizable=yes')">
```

13.输入输出

输入:window.prompt();

输出:document.write();

14.转向:

location.href 属性

location.replace()方法

document 对象

代表整个文档信息

15.表单的访问方式

两种方式:

A.使用表单名称

```
<input type="button" value="点击" onclick="alert(document.demoform.username.value)">
```

B.第二种方式:

```
<input type="button" value="点击" onclick="alert(document.forms[1].username.value)">
```

使用 forms 隐含数组来实现

表单的提交方式:

两种方式

A.使用 submit 提交按钮

```
<form action="index.jsp" name="demoform">  
    <input type="text" name="username" value="zhangshan">  
    <input type="submit">  
</form>
```

B.使用 submit()方法

```
<input type="button" value="立即注册" onclick="document.demoform.submit()">
```

16.改变元素的源代码

innerHTML:表示对象内的源代码


```
var v = "<font color='red'>非常好</font>";  
document.all('message').innerHTML=v;
```

17.在 frameset 中,不同的 frame 之间如何取值

两种方式:

第一种方式:使用 frames 隐含数组

```
<input type="button" value="取 值"   
onclick="alert(parent.frames[2].document.all('username').value);">
```

第二种方式:你可以使用一个 frame 的具体名称

```
<input type="button" value="第二种方式取值"   
onclick="alert(parent.rightframe.document.all('username').value);">
```

这个 rightframe 名称来源于我们在 frameset 中的定义

这里,parent 代表父对象

18.iframe:inline frame

内嵌页面

src:表示页面的源路径

```
<iframe id="myframe" src="sport_index.html" frameborder="0"></iframe>
```

相互取值

在子页面中,如何获取父页面的内容

```
<input type="button" value="提 取 父 页 面 值"   
onclick="alert(parent.document.myform.username.value);">
```

在父页面中,如何来获取子页面的内容

```
<iframe id="myframe" src="E.html" height="300"></iframe>   
<input type="button" value="点击" onclick="alert(myframe.message.innerHTML);">
```

19.为控件来设置快捷键

```
<input type="text" value="" name="username" > <br>
```

```
<input type="button" value="提取" onclick="alert(document.all('username').value);"
accessKey="t">
```

使用时,alt+t

20.如何来判断用户按下了哪个键

event.keyCode

使用了 keycode 可以帮助我们实现一种快捷的效果

实例:

```
function demo(){
    var v= event.keyCode;
    /**
    if(v>=48 && v<=57){

    }else{
        alert('请输入整数');
    }*/

    if(v==13){
        //location.replace("E.html");
        document.myform.submit();
    }

}
```

21:setInterval 定时处理

表示在制定的时间段内,执行一个具体的操作

两个参数

A.要执行的函数

B.延迟时间

```
function demo(){
//表示每隔一秒来做一件事情
    setInterval("process()",1000);
}
```