Javascript特点

解释性：它是一种解释性的程序语言，它的代码不需要经过编译，而是直接在浏览器中运行时被解释。

事件驱动：无须经过web服务程序，它对用户的响应，是以事件驱动的方式进行的。

跨平台：只要能运行浏览器的计算机，与操作环境无关。

安全性：它不允许访问本地的硬盘，而且不能将数据存入服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互，这样可以有效地防止数据的丢失。

Javascript在html中的使用

1. 在页面中直接嵌入javascript 代码

<script>标记常用的属性及说明

|  |  |
| --- | --- |
| 属性 | 说明 |
| Language | 设置所使用的脚本语言及版本 |
| src | 设置一个外部脚本文件的路径位置 |
| Type | 设置所使用的脚本语言，此属性已代替language属性 |
| defer | 表示当html文档加载完毕后再执行脚本语言 |

2．链接外部javascript文件

如果脚本代码比较复杂或是同一段代码可以被多个页面所使用，则可以将这些脚本代码放置在一个单独的文件中（保存文件的扩展名为.js）

在外部js文件中，不需要将脚本代码用<script>t和</script>扩起来。

**基本语法**

执行顺序：javascript程序按照在html文件中出现的顺序逐行执行，如果需要在整个html文件中执行，最好将其放在<head>…</head>。

Javascript严格区分字母大小写，但与之紧密相关的html是不区分大小写的。

获取时间代码

<script type="text/javascript">

var now=new Date();

var hour=now.getHours();

var minu=now.getMinutes();

alert ("您好！现在是"+hour+":"+minu+"\n欢迎访问本公司网站");

</script>

所有的javascript变量都由关键字var声明。

如果只声明了变量，未对其赋值，则其值默认为undefined。

在javascript中，所有的数字都是由浮点型表示的。

在任何数值直接量前加负号( - )可以构成它的负数，但是负号是一元求反运算符，它不是数值直接量语法的一部分。

16进制（以16为基数）数据：是以“OX”“ox”开头(貌似数字0？)，其后直接跟随十六进制数字串的直接量。16进制的数字可以用到0~9中的某个数字，也可以是a(A)到f(F)中的某个字母，它们用来表示0到15之间（包括0和15）的某个值。

单引号内出现双引号或双引号内出现单引号时，不需要进行转义。

但是双引号内出现双引号或单引号内出现单引号，则必须进行转义（在单（双）引号前）加” \ ”。

即同种引号需要转义

布尔型：有两个直接量true和false表示，用来说明某个事物是真还是假。

转义字符：以反斜杠开头的不可显示的特殊字符。

在“document.writeln();”语句中使用转义字符时，只有将其放在格式化文本中才会起作用，所有脚本必须放在<pre>和</pre>标签内。

例： document.writeln(“<pre>”);

document.writeln(“轻松学习\njavascript语言！”);

document.writeln(“</pre>”);

未定义类型的变量是undefined，表示变量还没有赋值（“var a;”），或者赋予一个不存在的属性值(“var a=String.notProperty;”)。

当在程序中由于某种原因发生计算错误后，会产生一个没有意义的数字，此时JavaScript返回的数字值是NaN（即非数字）。

如果试图引用一个没有定义的变量，则返回一个null值，表示为空值，用于定义空的或不存在的引用。注意：null不等同于空的字符串(“ ”)或0。

区别：null表示一个变量被赋予了一个空值，而undefined则表示该变量尚未被赋值。

比较运算符的操作过程：首先对操作数进行比较，该操作数既可以是数字也可以是字符串，然后返回一个布尔值true或false。

Javascript中的异于c语言的比较运算符

|  |  |  |
| --- | --- | --- |
| 运算符 | 描述 | 示例 |
| == | 等于。只根据表面值进行判断，不涉及数据类型 | “17”==17 //返回值为true |
| === | 绝对等于。根据表面值和数据类型同时进行判断 | “17”===17 //返回值为false |
| != | 不等于。只根据表面值进行判断，不涉及数据类型 | “17”!=17 //返回值为false |
| !== | 不绝对等于。根据表面值和数据类型同时进行判断 | “17”!==17 //返回值为true |

异或运算符

|  |  |
| --- | --- |
| a^=b //相当于a=a^b | |
| 0^0=0 | 0^1=1 |
| 1^1=0 | 1^0=1 |

口诀：相同值为0，不同为1。

**位操作运算符**

普通位运算符：在进行计算前，都先将操作数转换为32位的二进制整数，然后再进行相关运算，最后的输出结果将以十进制表示。

位操作运算符：对数值的位进行操作，如：向左或右移位。

**typeof运算符**：用于返回它的操作数当前所容纳的数据的类型。

例： var a=2035;

var b="accelworld";

var c=null;

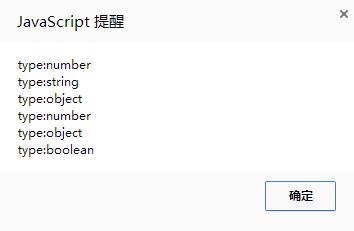
var d=20.35;

var e=Array(45);

var f=true;

alert("type:"+(typeof a)+"\ntype:"+typeof b+"\ntype:"+typeof

c+"\ntype:"+typeof d+"\ntype:"+typeof e+"\ntype:"+typeof f);



**new运算符**：通过new运算符来创建一个新对象。

语法：new constructor[(arguments)]

Constructor：必选项。对象的构造函数。如果构造函数没有参数，则可以省略圆括号。

arguments：可选项。任意传递给新对象构造函数的参数。

例： object1=new Object;

array2=new Array();

date3=new Date("august 8 2008");