**JavaScript介绍**

是一种小型的、轻量级、面向对象的（由属性和方法构成）、跨平台的客户脚本语言。嵌入到浏览器软件中的客户端（不能在服务器运行）脚本语言。

JavaScript功能

1表单验证：最基本功能（注册、）；2动态HTML可以实现动态、重复的效果（图片幻灯）；3交互式：人机交互。

<script></script>标记

JS代码嵌入HTML文档中。

常用的两个客户端的输出方法

document.write(“内容”);在body中输出文本；write()是document对象的一个输出方法，通过小数点来调用对象的方法。

Window.alert(str)；在当前窗口中弹出警告的内容；Window代表当前浏览器窗口对象；alert()方法：弹出一个对话框；

Js中的注释

HTML中的注释<!--注释内容-->；css注释：/\*单行注释\*/；JavaScript注释：// 或/\*多行注释\*/

变量：变量是一个变化的量，一个”未知数”；是一个程序的数据，在内存个中存在运行；临时存在的数据。

变量的数据类型：数值型、字符型、布尔型、underfined、null、array、object、function.

1数值型：可以进行算数运算的 包括整型和浮点型。特别：NaN(not a number不是一个数值)当其它数值类型，转成数值型，转不过去，但程序又不能报错，将返回一个NaN。

2字符型：用单引号或双引号引起来的一个字符串；字符型变量不能进行算术运算，只能进行“连接”运算。单引号和双引号之间可相互嵌套：单引号内只能嵌套双引号；双引号内只能嵌套单引号。

JS中的运算符

+:左右为数值，就执行加法运算；如果左右有一个为字符串（数值+字符），将执行“连接”运算；

赋值运算符：=、+=

=赋值运算符，将=右边的运算结果，赋给左边的变量；+=先加后等（a+=20 a=a+20）；

变量初始化：变量声明、变量赋值；程序的运行过程；输出结果。

只有同类型的变量，才能进行运算。

变量的命名:只能是字母、数字、下划线三类资费；不能以数字开头。JS是区分大小写的 ，即变量的命名。

基本数据型：数值、字符、布尔、未定义、空。一个变量名只能存一个值。

复合数据型：数组、对象、函数。一个变量，可能有多个值。

转义字符：如果想在双引号，内嵌套双引号，里面的双引号，一定进行转义（\”）.

转义符号：\’ \” \\ \r \n等。当浏览器遇到反斜杠时，将会对其后的一个字符进行特殊对待，当成特殊字符来对待。弹窗中的换行，只能用\n来实现；

布尔型：又称逻辑型。只有两个值：true真假、false假。常用于if条件判断语句；

if条件判断——只判断true不判断false：

if(条件判断：只有两个结果true或false )

{

条件为true,执行改行代码

}；

If为系统判断关键字，

If条件判断——既判断true 也判断false

If(条件判断：结果为true 或者为false)

{

条件为true,将执行代码;

}else

{

条件为false，将执行该代码;

}

未定义型：当一个变量定义，但未赋值时，将返回未定义型，未定义型的的值就只有一个undefined.当一个对象的属性不存在，也返回未定义型。

空型：当一个对象不存在是，将返回空型，空型的值只有一个null。是一个对象的占位符。

清除一个变量的值，可以给赋一个null值。

变量的数值类型转换

其他类型转换为布尔型

判断变量的类型：typeof()

使用一元运算符typeof()，可以测试一个变量的类型。结果是一个类型字符串。String、number、boolean、underfined、object（null、对象、数组返回object）、function。

特殊字符

New运算符：创建一个对象。如：var today=new Date();//创建一个当前日期时间对象

Delete运算符：删除数组的元素，或者对象的属性。

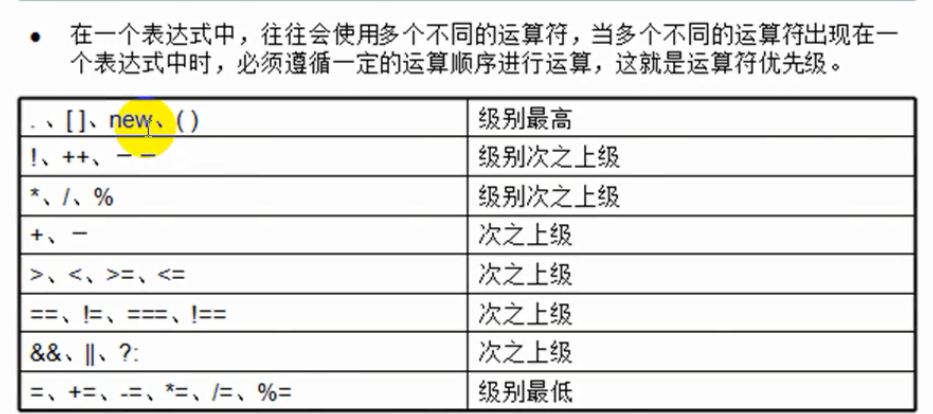
Typeof运算符：一元运算符，主要功能：判断变量的类型。Typeofname 或 typeof(name)

点运算符（.）：应用在对象中，通过点来运算符去调用属性或方法。Window.alert(“OK”)

[]中括号运算符：主要用来访问数组元素的。Arr[0] = 100; //找到下标为0的数组元素，并重新赋值 对数组中第一个值 换成100。

运算符的优先级

特殊运算符>算术运算符>比较运算符>逻辑运算符>赋值运算符



If（括号为布尔型）条件判断语句：条件成立，执行什么代码；不成立，执行什么代码

结构：1值判断真（true），条件为假，不执行if(条件判断：判断结果为布尔型) { 条件为真true 执行的代码} ；2既判断真，也判断假 if(条件判断) {条件为真，执行的代码 } else{ 条件为假，执行的代码 }；3多条件判断 if(条件1){ 代码1； } else if (条件2) { 代码2； } else if ( 条件3 ) { 代码3 ； } ...... else { 如果上面条件都不成立执行该代码; } 各调价你是“或”的关系。

Switch分支语句：根据一个变量的不同值，来执行不同的代码。

语法 ：switch(变量) { case 值1:代码1;break; case 值2: 代码2;break; case 值3: 代码3;break; default:如果上面条件都不满足，则执行改代码 }

1 switch后的小括号内是一个变量名，这个变量可能有不同的取值 ；2每个cased的值，与变量的值进行对比，如果一致就执行该case后的代码；3所有的case都是或关系，只有一个case满足；4每个case都是用break来结束，结束之后会跳到switch结束大括号之后运行。

区别：if与switch if中的条件一般是一个范围，而switch的条件一般是一个变量的不同取值。

While循环：只要条件成立，就重复不断的执行循环体代码

结构：While(条件判断) { 如果为true,则执行循环体代码 }

说明：1循环开始前，必须对变量初始化，声明变量，给变量给个初始值；2如果while的条件为true,则重复不断执行{}中代码；3循环体中，必须要有变量更新的语句。

For( 初始化变量 ; 条件判断 ; 变量更新) { 循环体代码 }

break和continue语句

Break语句：哦用于无条件结束各种循环（退出）和swith

一般情况，需要在break语句前加一个条件判断。条件成立，退出循环。

不能一次跳出多重循环

Continue语句：结束本次循环，而开始下一次循环。

数组：一组数的集合。

数组元素：数组中的每一个值，称之为“数组元素”。

数组中的下标号是从0开始的正整数。第一个为0；第二个为1；第三个为3；以此类推。

使用数组的目的，就是使用循环便利数组很方便。

访问的方法是：数组变量名，后跟一个中括号[],[]中括号内为元素的小标。

数组的创建方法

1使用new关键字和Array()来创建个数组

var arr = new Array(); //创建一个空数组

var arr = new Array(“周更生”,”男”,”30”); //创建数组，并初始化数组的元素

2使用[]来创建var arr = [10,20,30]

数组的操作

增加：增加一个不存在的下标元素

删除:使用关键字delete,只删除元素的值，而下标还在。

修改:修改已经存在下标的元素（重新辅助）

读取:读取已经存在的小标元素。

数组的对象属性length

一个数组，其实是一个数组对象，把数组看成对象，是为了使用数组对象的属性或方法

legth就是数组对象的一个属性，如：varlen = arrObrj.length;

legth属性，可以动态获取数组的长度(个数)