第8讲 JavaScript基本语法

2018年5月17日 C17f38

JavaScript简称JS，在学习JS之前，我们先简单地了解几个关于JS的问题，以对这个语言有一个总体上的认识。

# 1、JavaScript概述

## 1.1、JavaScript是什么？

JavaScript是一种web前端的描述语言，也是一种基于对象（object）和事件驱动（Event Driven）的、安全性好的脚本语言。它运行在客户端。

javaScript主要用来向html页面中添加用户交互行为，在语法方面和c语言系列语言的语法类似，属弱类型语言。

javaScript是一种解释型语言，边执行边解释无需另外编译。

## 1.2、JavaScript的用途是什么？

JavaScript的用途是解决页面交互和数据交互，最终目的是丰富客户端效果以及数据的有效传递。

1. 实现页面交互，提升用户体验实现页面特效。即使用js操作html的dom节点或操作样式。
2. 在客户端对表单数据进行验证，即在数据送达服务端之前进行用户提交信息即时有效的检查，减轻服务器压力，这即是数据交互。

## 1.3、JavaScript由哪几部分组成？

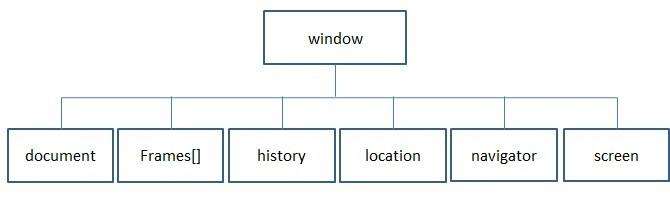
所谓JavaScript的组成部分，是指JavaScript的对象模型。通俗地说，就是利用JS进行程序设计时,JS内容各类对象的层次关系、结构关系。

打个比方，如果学校就是JS的话，学校由哪几部分组成呢？学校从机构设置上看，由行政部门、后勤部门、教学部门、管理部门、发展部门组成，而行政部门又包括具体的校长办公室、人事处、党委办公室，后勤部门具体又包括后勤处、书院管理中心、物业处、计财科……。如下图。这就是学校的OOM模型（Organization Object Model，组织对象模型）

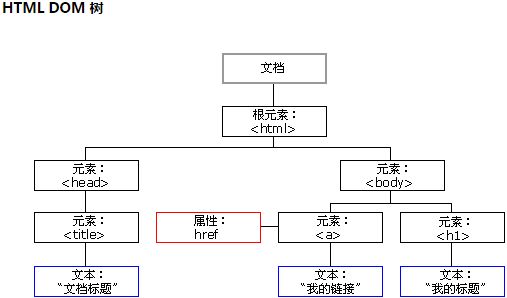
如果从职位身份的角度看，学校又由校长、副校长、主任（科长）、副主任（科长）、教师、学生组成。这是学校的IOM模型（identify object model，身份对象模型。

同理，JS语言也有它自己的对象模型，从不同的角度看，它有两个不同的模型——BOM与DOM。

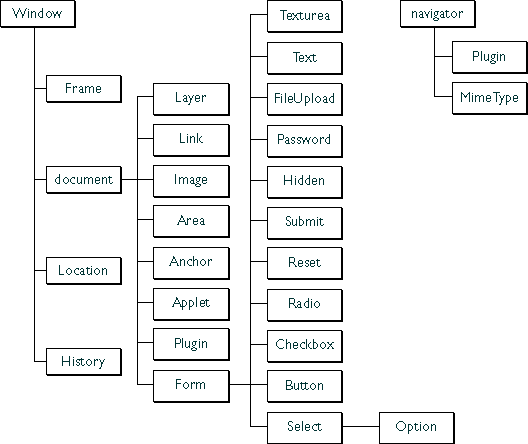
（1）BOM(Browser Object Model)，浏览器对象模型



（2）DOM（document object model），文档对象模型，是BOM模型中document对象的细化。



以上就是Js的组成。许多文献资料中，通常把这两个模型统一在一起，形成JS的OM（对象模型）



## 1.4、JavaScript的执行原理是怎样的？

JavaSript中基于事件机制的，且程序是在浏览器端运行的，因此，它由浏览器负责解释执行，不关乎服务器。用户通过浏览器提出访问URL，服务器将对应的文件返回到用户的浏览器，当适宜的事件触发了文件中的JS时，浏览器就执行所触发的JS程序。

例如下面的程序，是当用户单击页面中的按钮时，就运行myjs()函数。

<script language="javascript">

function myjs()

{

document.writeln("这是我的js程序，点击按钮就执行");

}

</script>

<input type="button" name="js" value="测试js" onclick="myjs()" />

由于JS程序是在浏览器端完成的，因此除非程序要与服务器之间进行数据交互，否则所有的程序，只对浏览器中的内容产生影响，不会影响到服务器上的原文件。并且，它的执行结果与浏览器紧密相关，可能会因为一些浏览器禁用JS而导致程序无法正常运行。

## 1.5、在页面文件中是如何引入javaScript文件的？

(1)使用<script>...</script>标签

<script type="text/javascript">

<!--　//javaScritpt语句； -->

</script>

(2)使用外部js文件

<html> <body>

<script src="myScript.js"></script>

</body> </html>

(3)直接在html标签中。

<a href="wod.html" onclick="alert('你单击到我了');">直接js</a>

# 2、JS的基本语法要求

**（1）区分大小写**

JavaScript语言区分字符大小写，两个字符串相同大小写不同，被认为是不同的字符串。JavaScript语言的关键字也区分大小写，按语法要求应小写。

**（2）书写格式**

JavaScript语言忽略语句间空白，即语句间的空格，空行，缩进等不影响程序的本质。为了提高程序的可读性，应当使用这些格式，使程序更加清晰，可读性更高。

**（3）注释语句**

为了提高程序的可维护性和可读性，应当有一定的注释语句，它是给读程序的人员看的，有单行注释和多行注释，单行注释以双斜线//开始，多行注意以/\*开始,以\*/结束。

**（4）分号的使用**

JavaScript语言中每句代码以分号结束。

**（5）JavaScript放置的位置**

JavaScript代码可以放在HTML页面中的<head></head>内，也可以放在<body></body>内，都需要用<script>标签声明是JavaScript脚本类型。如果JavaScript代码没有直接放在HTML页面，而在另一个文件 中，比如MyPage.js，可以使用<script>的src属性链接它：

<script type=”text/Javascript”src=” MyPage.js”>

**（6）系统关键字（也称系统保留字）**

有一些单词，是JS已经作为自己的内部对象、属性或方法使用的，如document，style，value等，这些称为保留字。保留字不能再作为用户程序自定义的内容使用。

# 3、JS的基本语法

## 3.1、变量

变量的声明：var varialName;

变量的赋值：varialName ="sunking"

变量名的组成：数字（0-9）、字母（a-z，A-Z）、下划线（\_）

例：在网页弹出显示“hello！My name is lsx”。

<html>

<html>

<body onLoad=**"showHello();"**>

<Script Language=**"JavaScript"**>

***function*** showHello**()**

**{**

***var*** usename**;**

usename**=**"lsx"**;**

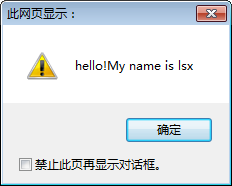
alert**(**"hello!My name is "**+**usename**);**

**}**

</script>

</body>

</html>



## 3.2、数据类型

JS是一种弱类型语言，它并不要求声明变量时就定义好数据类型，而是根据所赋予的变量值，决定变量的类型，一个变量可以在上一时刻是整型，而下一时刻是浮点型。

因此，js的数据类型也并不多，可以归纳为以下几个类型：

字符串、数字、布尔、数组、对象、Null。

### （1）字符串

字符串类型可以是引号中的任意文本，可以使用单引号或双引号，一般使用双引号；例：

var p=”this is a string”;

### （2）数字类型

JS的数字类型可以带小数点，也可以不带。

例：

var n1=12;

var n2=12.5;

### （3）布尔型

也称逻辑型，它只有两个值：true 或 false。

### （4）数组型

JS的数组用new Array()语句来声明，可以在声明的同时就给数组元素赋值，也可以声明结束以后，再用下标给指定的元素赋值。下标从0开始。

例：

var cars=new Array();

cars[0]="Audi";

cars[1]="BMW";

cars[2]="Volvo";

也可以这样写：

var cars=new Array("Audi","BMW","Volvo");

var cars=new Array(4);

### （5）对象型

对象是一种更加丰富的数据类型，它其实可以看作是一组不同数据类型的组合。

对象由花括号分隔。在括号内部，就是数据，这些数据可以看作是对象的不同属性的集合，以名称和值对的形式 (name : value) 来出现。属性之间用逗号分隔。

例如定义一个man变量，数据类型为对象型

var man=

{

sname:”张三”,

sage:18,

sjob:”saler”,

};

使用对象型变量的值时，可以按照数组的形式对变量中的某个属性进行使用，也可以按“对象名.属性名”的形式使用。例：

var fname;

fname=man.sname;

//或者 fname=man[“sname”];

### （6）null型

null型也可称为空型，它只有一个值，就是null。

var a=null;

在定义一变量时，可以通过直接赋值确定变量的类型，也可不指定数据类型，也不赋值，这时变量的类型相当于“undefined”（未定义）。还可以用new关键字来声明其数据类型，但不赋值。

var b;

例：

var carname=new String; //字符串型

var x= new Number; //数字型

var y= new Boolean; //布尔型

var cars=new Array;//数组型

var person= new Object; //对象型

## 3.3、运算符

JavaScript的运算符，按不同的运算分类如下：

### （1）赋值运算符

|  |  |  |
| --- | --- | --- |
| **运算符** | **例子** | **等价于** |
| = | x=y |  |
| += | x+=y | x=x+y |
| -= | x-=y | x=x-y |
| \*= | x\*=y | x=x\*y |
| /= | x/=y | x=x/y |
| %= | x%=y | x=x%y |

### （2）数学运算符

+：数学加

-：数学减

\*：数学乘

/：数学除

%：求余，A%B表示A整除B的余数

++：递增1，A++相当于A=A+1 ++A

--：递减1，A—相当于A=A-1 --A

### （3）字符串运算符

在字符串操作中，“+”表示连接，A=”hello!”;B=”world”,

A+B的结果是”hello!world”

### （4）关系运算符

关系运算符也称比较运算符，有四个:==(相等)，！=（不等于）， =>（大于或等于）， =<（小于或等于）

关于运算符的结果是逻辑值，即要么true,要么false。

例如：

var A=5;

var B=3;

var C=8;

A+B==C; //结果为true

A+B<=2; //结果为false

### （5）逻辑运算符

逻辑运算有三种：&& （与），||（或），！（非），并且这类运算符的操作数必须是布尔型或者结果为布尔型的表达式。

逻辑运算的结果，依然是逻辑型。

&&（与）：只有两个操作数都为true，运算结果才是true

||(或)：只要两个操作数中任意一个为true，运算结果即为true

!（非）：只有一个操作数，!true==false,!false==true;

# 4、js对象的属性及方法

## 4.1、什么是对象(object)

JavaScript中是基于对象的编程，但不是完全的面向对象的编程。

什么是对象呢？这是一个不容易通俗表达的概念，还是回到上文所说的学校例子。

学校是不是对象呢？不是。因为全中国全世界都有学校，但学校是什么样的呢？我们会通过一系列的特征来说明，如校名、校史、办学层次、地址、办学规模……。但问题是，无法对这些特征进行统一的表达，比如我们不能说校名叫“清华大学”，因为并不是所有的学校都叫清华大学。其它的各个特征值也都无法统一。因此，“学校”其实只是一类事物的统称，并不是某个具体的事物。程序设计中，往往把这样一个统称性的东西，称为“类”。

但“惠州城市职业学院”就是一个对象了，是“学校”这个类的一个具体化存在了，学校的一切特征，在这个存在身上，都可以找到具体的值，例如校名叫“惠州城市学院”，校史也有一个具体的描述，学校层次是大专，地址是惠州三栋镇……。

每个对象有很多特征，我们称之为对象的属性，如上面说的“校名”“校址”就是属性。每个对象也可以完成一些事情，即具备一些功能，称为对象的方法。例如学校可以招生，也可以培训，也可以考试。这些就是学校的方法。

## 4.2、JS的对象

JS也有很多对象，主要有以下几种：

(1)、由浏览器根据web页面的内容自动提供的对象，如document、form、div、button这些html标签，在js中，都可以看作是文档对象document的子对象

(2)、JavaScript的内置对象，如Date（日期）,Math（数学），string（字符串）等等。

(3)、用户自定义的对象，例如前面说的对象型数据，就是用户自定义的对象。

JavaScript中对象的方法就是该对象中的函数，一个函数因为能够完成一系列事情而具备某些功能，因此就是一个方法。

访问一个对象的属性或调用一个对象的方法，用以下语法格式：

对象名.属性名

对象名.方法名

例如，我们要知道某个字符串的长度，就可以这样写：

<script language="javascript">

var str="this is my javascript";

var len=str.length; // 长度属性

</script>

而把字符串转成大写，这就是需要使用到字符串对象的功能了，因此调用它的方法

<script language="javascript">

var str="this is my javascript";

var str1=str.toUpperCase();//调用方法

</script>

以后将专题介绍js的对象。

# 5、JS的事件机制

事件是浏览器响应用户交互操作的一种机制，它既包括浏览器自身的一系列动作（例如打开、关闭、获取焦点等等），也包括用户的一系列操作（例如单击、双击、移动鼠标等等）。

为了响应某个事件（操作）而进行的处理过程，叫做事件处理。 　　事件定义了用户与页面交互时产生的各种操作，例如单击超级连接或按钮时，就产生一个单击（click）操作事件。

浏览器在程序运行的大部分时间都等待交互事件的发生，并在事件发生时，自动调用事件处理函数，完成事件处理过程。

前文说过，事件不仅可以是用户的操作，也可以是浏览器自己的一些动作，例：当载入一个页面时，就会发生load事件，卸载一个页面时，就会发生unload事件等。同一个处理过程，定义在不同的事件中，效果是完全不一样的。

例如下面的两个函数，myfun1()是在浏览器刚载入文档时就起作用了（被调用），而myfun2()要在用户单击按钮时，才会被调用。

<script language="javascript">

function myfun1()

{

//myfun1函数体

}

function myfun2()

{

//myfun2函数体

}

</script>

<body onload="myfun1();">

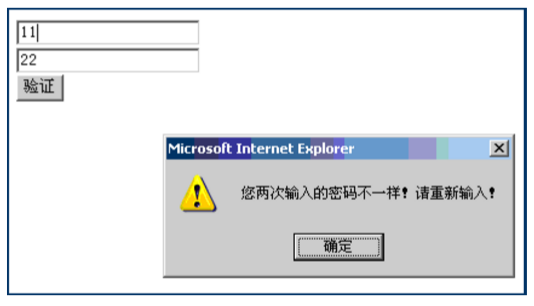
<input type="button" name="js" value="调用js" onclick="myfun2();" />

</body>

# 6、课堂练习：

（1）验证密码：

在文档中设计一个密码确认程序，在两个文本框中分别输入密码与确认密码， 点击“验证“按钮时，程序对两个文本框的内容进行检查，必须两次密码都完全相同，并且密码长度要大于6位，否则给出提示。如图所示：



（2）验证用户名是否为空：

设计界面如下图，当用户名文本框失去焦点时,检查用户名是否为空,如果为空则在用户名文本框后面给出红色提示“请输入你的用户名“，如果已经填写内容，则不显示任何提示。

C:\Users\Administrator\AppData\Roaming\Tencent\Users\522621691\QQ\WinTemp\RichOle\$PH0G8TA54`}K8A_)CEY7SI.png

C:\Users\Administrator\AppData\Roaming\Tencent\Users\522621691\QQ\WinTemp\RichOle\2TM~8A)5__B68AB`9@8DG%Q.png

（3）提交用户信息

请在html文档中实现以下效果：页面中提供一个“提交”按钮以及一个性别单选按钮组，（图1）

当选择“男”并“提交”后，显示“你是男的”（图2），选择“女”并提交后，显示“你是女的”（图3）

