day08_JavaScript基础

JavaScript是一门客户端脚本语言，脚本语言是不需要编译的，每个浏览器都有一个JavaScript解析引擎，JavaScript代码(程序)可以直接被浏览器解析执行，所以JavaScript是运行在客户端浏览器中的。JavaScript可以用来控制 html 元素，让页面有一些动态的效果，增强用户和html页面的交互过程。JavaScript语言的发展史如下：

- 1992年，Nombase公司，开发出第一门客户端脚本语言，专门用于表单的校验，命名为：C--，后来更名为：ScriptEase
- 1995年，Netscape(网景)公司，开发了一门客户端脚本语言：LiveScript。后来请 SUN 公司的专家修改LiveScript，并将其命名为JavaScript
- 1996年，微软抄袭 JavaScript 语言开发出了Jscript语言，从此之后，这种运行在浏览器上的客户端脚本语言进入三足鼎立的时代
- 1997年，ECMA(欧洲计算机协会)为了统一所有客户端脚本语言的编码方式，以JavaScript为原型，制定出客户端脚本语言的标准：ECMAScript

所以：JavaScript = ECMAScript + JavaScript自己特有的东西(BOM、DOM对象)

一.ECMAScript的基本语法

1.与html的结合方式

(1).内部方式：定义<script>标签，标签体内容就是js代码

(2).外部方式：定义<script>标签，通过src属性引入外部的js文件(一般将所有的js文件都放在js目录下)

#.注意：
①.js代码的注释方式：与java一样，分为单行注释和多行注释
②.在同一个html页面内，<script>标签可以定义多个。并且<script>标签可以定义在html页面的任何地方，但是定义的位置会影响执行顺序：
在示例代码中，<script>标签定义在<input>标签的前面，所以会先执行<script>标签中的js代码，再执行后续的代码。并且由于alert语句的阻塞作用，所以不点击弹出对话框的确定按钮，body标签中的文本输入框不会显示在页面中。

2.数据类型、变量

在程序中定义一个变量，就是在内存中开辟一块空间用来存储数据，这块内存空间的名字就是该变量名。Java是一种强类型语言，而JavaScript是一种弱类型语言。在JavaScript中，定义一个变量的语法如下：　　var 变量名 = 初始化值;

- 强类型：在开辟变量的内存空间时，规定了该内存空间的数据类型，因此只能存储固定类型的数据(如：int a = 3; 变量a的内存空间规定为int类型，变量a只能存储int类型的数据)
- 弱类型：在开辟变量的内存空间时，不规定该内存空间的数据类型，因此可以存放任意类型的数据(如：var a = 3; 变量a的内存空间没有规定数据类型，变量a可以存储任意类型的数据)

#.注意：
①.document.write()语句可以将内容输出到页面上，其参数可以为变量、常量。并且可以将html标签作为参数传入，写入的时候会按照html进行解析(但是标签作为参数传入时，需要使用字符串形式)
②.变量的typeof()运算符：用来获取变量的数据类型

(1).原始数据类型(基本数据类型)

关键字	类型名称	解释
number	数字类型	整数、小数、NaN(not a number 一个不是数字的数字类型)
string	字符串类型	js中只有字符串、没有字符，使用单、双引号引起来的均为字符串类型：“abc”、“a”、‘abc’
boolean	布尔类型	true、false
null	空类型	若一个变量存储的是null，则代表该变量是空类型，意为：一个对象为空的占位符
undefined	未定义类型	若一个变量存储的是undefined、或者没有初始赋值，则代表该变量是未定义类型

(2).引用数据类型：对象

3.运算符

(1).一元运算符：只有一个运算数的运算符

①.自增运算符 ++、自减运算符 --
②.正号 +、负号 -

#.注意：

在js中，如果运算数不是运算符所要求的类型，那么js引擎会自动的将运算数进行类型转换。即：如果我在string、boolean类型的数据前面加一个正号，由于正号运算要求的是number类型数据，那么string、boolean类型的数据会按照以下规则转换为number类型，再参与正号运算。(其他运算符同理)

string转number：按照字面值转换。如果字面值不是数字，则转为NaN(不是数字的数字)

boolean转number：true转为1、false转为0

(2).算数运算符：加 +、减 -、乘 *、除 /、取模 %

(3).赋值运算符：基本赋值 =、加等于 +=、减等于 -+ ...

(4).比较运算符：大于 >、小于 <、大于等于 >=、小于等于 <=、等于 ==、全等于 ===

①.类型相同：直接比较

②.类型不同：先进行类型转换，再比较

③.全等于 ===：在比较之前，先判断类型，如果类型不一样，则直接返回false

(5).逻辑运算符：与 &&、或 ||、非 !

①.逻辑运算符要求的运算数必须是boolean类型，当运算数是其他类型的时候，会按照如下规则将其转换为boolean类型，再参与逻辑运算

number转boolean：0、NaN为false，其他为真

string转boolean：除了空字符串("")为false，其它都是true

null、undefined转boolean：都是false

对象转boolean：所有对象都为true

②.由以上数据类型转换成boolean的规则，我们可以延伸出以下使用技巧：

在java中死循环的写法：while(true){ }　　在js中死循环可以写为：while(1){ }

在java中防止对象的空指针异常：if(obj != null){ }　　在js中防止对象的空指针异常：if(obj){ }

在java中防止字符串为空：if(str !=null && str.length > 0){ }　　在js中防止字符串为空：if(str){ }

(6).三元运算符：表达式 ? 值1 : 值2;　　判断表达式的值，如果是true则返回值1，如果是false则返回值2

4.流程控制语句

在js中，流程控制语句的使用语法与java无异。

(1).if语句、标准if-else语句、扩展if-else语句

(2).switch语句：在js中，switch语句的表达式可以接受任意的原始数据类型

(3).for循环、while循环、do...while循环

5.练习：输出99乘法表，需要实现的效果见 "参考\99乘法表案例效果.png"

#.特殊语法：

- 每条语句以分号结尾，如果一行只有一条语句，则分号可以省略不写(不建议省略)
- 变量的定义可以使用var关键字，也可以不使用，不加var定义的变量必须对其进行初始赋值(建议使用var关键字)

在方法体内，加var为局部变量，不加var则是全局变量

在方法体外，加不加var都是全局变量

二.ECMAScript的基本对象

1.Function对象

Function对象：函数、方法对象

(1).方法对象的创建

法一：　var 方法名 = new Function(形式参数列表，方法体);　　基本不使用，可以忘记了

法二：　function 方法名(形式参数列表) {
　　方法体;
}

法三：　var 方法名 = function(形式参数列表) {
　　方法体;
}

(2).方法的调用：方法名称(实际参数列表);

(3).方法对象的属性：

length属性：返回形参的个数

...

#.注意：

①.由于方法是一个对象，所以方法可以被重新赋值。并且如果定义名称相同的方法，并不会报错、会覆盖。

②.在js中，方法的调用只与方法的名称有关，和参数列表无关

③.在方法声明中有一个隐藏的内置对象：arguments数组对象，用来封装传递进去的所有实际参数

④.方法定义时，由于形式参数、返回值的类型都是var，所以不需要写，写了反而会报错

2.Array对象

Array对象：数组对象

(1).数组对象的创建

法一： var 数组名 = new Array(元素列表);

法二： var 数组名 = new Array(数组长度);

法三： var 数组名 = [元素列表];

(2).数组对象的使用：和java中使用下标来访问数组元素的方法一样

(3).数组对象的属性：

length属性：返回数组的长度

...

(4).数组对象的方法：

join(分隔符参数)：将数组中的元素按照指定的分隔符拼接为字符串并返回

push()：向数组的末尾添加一个或更多元素，并返回新的长度

toString()：把数组转换为字符串，并返回结果

reverse()：颠倒数组中元素的顺序

...

#.注意：

①.在js中，数组元素的类型是可变。即：数组中的元素类型可以是不统一的任意类型。

②.在js中，数组的长度是可变的。即：访问不存在的下标元素时，不会报错，数组会自动扩容。

3.Date对象

Date对象：日期对象

(1).日期对象的创建： var 对象名 = new Date(); 创建的Date对象会自动把当前日期和时间保存为其初始值

(2).日期对象的方法：

toLocaleString()：以本地时间的字符串格式，返回当前date对象对应的时间

getTime()：获取date对象的毫秒值。即：当前date对象描述的时间到1970年1月1日零点的毫秒值之差

...

4.Math对象

Math对象：数学对象，类似于java中的Math工具类，Math对象不用创建，可以直接使用其中的属性和方法

(1).Math对象的属性：

PI：返回圆周率(约等于3.14159)

E：返回算术常量e，即自然对数的底数(约等于2.718)

(2).Math对象的方法：

random()：返回[0, 1)之间的随机数

ceil(x)：向上取整

floor(x)：向下取整

round(x)：四舍五入

...

5.RegExp对象

RegExp对象：正则表达式对象，在js中可以根据正则表达式创建一个正则表达式对象来对其进行使用

(1).正则表达式对象的创建：

法一： var reg = new RegExp("正则表达式");

法二： var reg = /正则表达式/;

(2).正则表达式对象的方法：

test(字符串参数)：验证指定的字符串是否符合正则定义的规范

(3).正则表达式：定义字符串的组成规则(所有语言通用的一种规则)

①.单个字符

[ ]：匹配含有方括号之间的任意一个字符的字符串

举例如下：

[a]：匹配含有a的字符串

[ab]：匹配含有a或者b的字符串

[a-z]：匹配含有a-z其中一个字母的字符串

[a-zA-Z0-9]：匹配含有a-z、A-Z、0-9其中一个字符的字符串

②.量词符号

?：表示出现0次、1次

*：表示出现0次、多次

+：表示出现1次、多次

{m，n}：表示数量在[m，n]之间。m如果缺省：{ ,n} 最多n次；n如果缺省：{m，} 最少m次

举例如下：

[a]?：匹配出现0次、1次a的字符串

[ab]*：匹配出现0次、多次a或者出现0次、多次b的字符串

[a-z]{6，12}：匹配a-z其中一个字母出现6~12次的字符串

③.特殊符号：代表特殊含义的单个字符

\d：单个数字字符，相当于[0-9]

\w：单个单词字符，相当于[a-zA-Z0-9_]（单词字符包括：a-z、A-Z、0-9 以及下划线）

④.开始、结束符号：一般我们会给正则表达式加上开始、结束符号，来表示一个正则表达式的开始和结束

^：开始符号

$：结束符号

6.Global对象

Global对象：全局对象，意味着Global对象中封装的方法不需要创建对象就可以直接调用

(1).URL编码、解码的方法

首先我们要知道什么是URL编码，前端页面在与服务器进行通信的时候，要符合HTTP协议，而HTTP协议是不支持中文字符传输的，所以在前端页面向服务器传输一些中文数据时，需要将这些中文字符进行URL编码后再进行传输。服务器接收到中文字符URL的编码格式后，再对其进行URL解码即可获取到传输过来的中文数据。URL编码的规则就是：将中文字符的字节形式转换为十六进制，前面再加上百分号%。例如：传智播客 = %E4%BC%A0%E6%99%BA%E6%92%AD%E5%AE%A2

在js中，提供了一些将字符串进行URL编码、解码的方法，如下：

encodeURI()：URL编码（仅编码中文字符）

decodeURI()：URL解码

encodeURIComponent()：URL编码（仅编码除了字母、数字以外的字符）

decodeURIComponent()：URL解码

(2).parseInt()方法：将字符串转为数字。逐一判断字符串中的每一个字符是否是数字，直到不是数字为止，将前边数字部分转为number。若第一个字符就不是数字，则直接转换为NaN。

(3).isNaN()方法：判断一个值是否是NaN。由于NaN参与的 == 比较全部为false，所以需要使用此方法来判断一个值是否是NaN。(NaN六亲不认，连自己都不认)

(4).eval()方法：将字符串作为JavaScript脚本代码来执行

#.除此之外，还有 Number、String、Boolean 等基本对象，类似于java中基本类型对应的包装类，在此不再赘述