JavaScript

1.导入外部script： <script src="myScript.js"></script>

2. JavaScript 可以通过不同的方式来输出数据：

.使用 window.alert() 弹出警告框。

.使用 document.write() 方法将内容写到 HTML 文档中。

.使用 innerHTML 写入到 HTML 元素。

.使用 console.log() 写入到浏览器的控制台。

3. JavaScript是弱类型编程语言,定义变量都使用 var 定义,与 Java 这种强类型语言有区别.

在定义后可以通过 typeOf() 来获取JavaScript中变量的数据类型.

// Number 通过数字字面量赋值

// Number 通过表达式字面量赋值

// String 通过字符串字面量赋值

// Array 通过数组字面量赋值

// Object 通过对象字面量赋值

4.javascrip的变量可以存放值、表达式；最好字母开头

当向变量分配文本值时，应该用双引号或单引号包围这个值。

当向变量赋的值是数值时，不要使用引号。如果用引号包围数值，该值会被作为文本来处理。

如果重新声明 JavaScript 变量，该变量的值不会丢失

5.数据类型

字符串（String）、数字(Number)、布尔(Boolean)、数组(Array)、对象(Object)、空（Null）、未定义（Undefined）。

对象：对象由花括号分隔。在括号内部，对象的属性以名称和值对的形式 (name : value) 来定义。属性由逗号分隔

对象属性有两种寻址方式：

name=person.lastname; 对象名.属性名

name=person["lastname"]; 对象名[”属性名”]

对象方法

创建方法：methodName : function() { code lines }

使用方法：对象.methodName

internetbigbang

函数内未声明即使用的变量情况：

function func(){ undefined\_var=110}

在 func() 被第一次调用之前， undefined\_var 变量是不存在的即 undefined。func() 被调用过之后，undefined\_var 成为全局变量。

6.常见的事件

事件 描述

onchange HTML 元素改变

onclick 用户点击 HTML 元素

onmouseover 用户在一个HTML元素上移动鼠标

onmouseout 用户从一个HTML元素上移开鼠标

onkeydown 用户按下键盘按键

onload 浏览器已完成页面的加载

两个数字相加，返回数字相加的和，如果数字与字符串相加，先把数字转为字符串，再拼接返回字符串

7.正则表达式

修饰符 描述

i 执行对大小写不敏感的匹配。

g 执行全局匹配（查找所有匹配而非在找到第一个匹配后停止）。

m 执行多行匹配。

\d 查找数字。

\s 查找空白字符。

\b 匹配单词边界。

\uxxxx 查找以十六进制数 xxxx 规定的 Unicode 字符。

字符^表示匹配的字符必须在最前边.

字符$与^类似,匹配最末的字符.

test() 方法是一个正则表达式方法。

test() 方法用于检测一个字符串是否匹配某个模式，如果字符串中含有匹配的文本，则返回 true，否则返回 false。

exec() 方法是一个正则表达式方法。

exec() 方法用于检索字符串中的正则表达式的匹配。

该函数返回一个数组，其中存放匹配的结果。如果未找到匹配，则返回值为 null。

8.变量提升

变量提升：函数声明和变量声明总是会被解释器悄悄地被"提升"到方法体的最顶部。

9.严格模式

eval() 函数可计算某个字符串，并执行其中的的 JavaScript 代码。

"use strict"; 使用严格模式

10.void(0)

javascript:void(0) 中最关键的是 void 关键字， void 是 JavaScript 中非常重要的关键字，该操作符指定要计算一个表达式但是不返回值。

# 来定位页面的具体位置，格式为：# + id

**11.函数**

1）**函数参数：**

JavaScript 函数定义时显式参数没有指定数据类型。

JavaScript 函数对隐式参数没有进行类型检测。

JavaScript 函数对隐式参数的个数没有进行检测。

隐式参数，有个内置的对象 arguments 对象。argument 对象包含了函数调用的参数数组。

2）**函数调用**：

一般而言，在Javascript中，this指向函数执行时的当前对象

当一个函数不属于任何对象时。在 JavaScript 中它始终是默认的全局对象。在 HTML 中默认的全局对象是 HTML 页面本身，所以函数是属于 HTML 页面。在浏览器中的页面对象是浏览器窗口(window 对象)。以上函数会自动变为 window 对象的函数。myFunction() 和 window.myFunction() 是一样的；

函数作为全局对象调用，会使 this 的值成为全局对象。使用 window 对象作为一个变量容易造成程序崩溃。

this指的是，调用函数的那个对象。

3）**闭包：**

变量声明时如果不使用 var 关键字，那么它就是一个全局变量，即便它在函数内定义。

所有函数都能访问全局变量。实际上，在 JavaScript 中，所有函数都能访问它们上一层的作用域。JavaScript 支持嵌套函数。嵌套函数可以访问上一层的函数变量。

闭包就是能够读取其他函数内部变量的函数。由于在Javascript语言中，只有函数内部的子函数才能读取局部变量，因此可以把闭包简单理解成"定义在一个函数内部的函数"。所以，在本质上，闭包就是将函数内部和函数外部连接起来的一座桥梁。

最大用处有两个，一个是前面提到的可以读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中。

如果是一个表达式后面添加()就会执行，所以 在匿名函数外面添加()然后再在括号后面添加()就可以进行调用了 javascript中()括起来的部分是不能包含语句的，会被当作表达式处理，所以如果function被()括起来以后就被javascript引擎作为function表达式处理了。因此jquery最外层就是一个自调用匿名函数。但是他这么做的最大原因是想定义一个命名空间，这样就不会和全局的变量出现冲突

**12.JavaScript HTML DOM**

1）查找元素：

. 通过 id 查找 HTML 元素: document.getElementById("IDName");

. 通过标签名查找 HTML 元素：getElementsByTagName("p")

. 通过类名找到 HTML 元素：document.getElementsByClassName("intro")

2）改变HTML：

.改变 HTML 输出流：document.write() 可用于直接向 HTML 输出流写内容；

.改变 HTML 元素的内容：document.getElementById(id).innerHTML=新的 HTML；

. 改变 HTML 元素的属性：document.getElementById(id).属性名=新属性值；

3）改变CSS

document.getElementById(id).style.property=新样式

document.getElementById("p2").style.color="blue";

4）addEventListener() 方法用于向指定元素添加事件句柄。

addEventListener() 方法添加的事件句柄不会覆盖已存在的事件句柄。

你可以向一个元素添加多个事件句柄。

你可以向同个元素添加多个同类型的事件句柄，如：两个 "click" 事件。

你可以向任何 DOM 对象添加事件监听，不仅仅是 HTML 元素。如： window 对象。

事件传递有两种方式：冒泡与捕获。

事件传递定义了元素事件触发的顺序。 如果你将 <p> 元素插入到 <div> 元素中，用户点击 <p> 元素, 哪个元素的 "click" 事件先被触发呢？

在 冒泡 中，内部元素的事件会先被触发，然后再触发外部元素，即： <p> 元素的点击事件先触发，然后会触发 <div> 元素的点击事件。

在 捕获 中，外部元素的事件会先被触发，然后才会触发内部元素的事件，即： <div> 元素的点击事件先触发 ，然后再触发 <p> 元素的点击事件。

5）创建新的 HTML 元素

var para=document.createElement("p");

如需向 <p> 元素添加文本，您必须首先创建文本节点。这段代码创建了一个文本节点：

var node=document.createTextNode("这是一个新段落。");

然后您必须向 <p> 元素追加这个文本节点：

para.appendChild(node);

最后您必须向一个已有的元素追加这个新元素。

这段代码找到一个已有的元素：

var element=document.getElementById("div1");

以下代码在已存在的元素后添加新元素：

element.appendChild(para);