JavaScript是什么？

客户交互脚本语言。

客户：1、指的是用户本身；2、客户端；

交互：人机之间的动作与反馈。

脚本：需要按一定顺序执行的。

语言：计算机语言，JS是一种高级编程语言，跟HTML不一样，HTML严格上说并不是编程语言。

PS：所有编程语言必然会包括有两个特点：1、可以运算；2、可以进行判断。

JavaScript的作用：

在现在所有网页中都使用了该技术，它可以极大的丰富网页的用户体验与网站的功能效果。

开始使用JavaScript：

方法1：直接在HTML中使用<script>标签，在标签中所有内容都是JavaScript。

方法2：在HTML的<script>标签中加入src这个属性并连接到一个JS文件。

JavaScript基础：

1. 变量：

义定变量： var 变量名;

变量名的命名规范：

1. 不能使用纯数字；
2. 不能使用数字开头；
3. 不能使用除$以外的特殊字符命名； 对$a, $alex, \_a
4. 不要使用关键字进行命名；
5. JS是对大小写敏感的语言，所以变量A与变量a是不同的。

var a = '牛奶'; // 变量的赋值

变量的赋值永远是把右边的值赋给左边。

什么是变量：可以认为是个有名字用于存放数据的容器；

ES5 之后的一变量定义的改变：

let 代替 var 来进行普通变量的定义；

新增了const 这个关键用于定义常量，变量在程序中任何位置都可以把值进更改。常量只能在定义它的位置进行改变。也就是说常量是无法重新赋值的。另外我们约定俗成一般常量名都使用全大写来代表常量。

1. 变量的数据类型：
   1. 字符串（string）：

定义方法： var str = ‘值’;

字符串的类型 都 需要使用定界符时行包裹。 定界符有两个，单引号 ‘ , 双引号 “;

一般情况下单引号与双引号的使用上是没有区别的。

但是如果字符串中有单引号我们则使用双引号做定界符，反则一样如果字符串中有双引号则使用单引号做定界符。

除此以外，对于字符串比较特殊的字符，我们可以使用 转译进行转译防止出错。

转译符 \ ，作用有两个：

1. 把后一位字符变成纯字符输出；
2. 对于特殊字符的输出； 比方： \r 代表的是回车

对字符串与字符串或变量之间的拼接需要使用拼接符进行拼接。 拼接符 +

自拼接：

str += ‘，你好’; 等介于 str = str + ‘，你好’;

2、数字类型（Number）：

1）整型（integer）：整数，包含正负。

2）浮点（float）:小数，包含正负。

数字（数值）跟字符串就相差一对定界符。判断是否是数字或字符串时需要使用 typeof这个关键进行判断。

1. 运算符

n = i + j; // + 求和运算符

由于+号有多个不同作用，所使用上面需要留意。

只有当+的两边都数字类型时，它才是求和运算，只要有一边是字符串，那它就成了拼接符。

n = i - j; // - 求差运算符

n = i \* j; // \* 求积运算符

n = i / j; // / 求商运算符

n = i % j; // % 求余（模）运算符

求余一般用在对需要数字有规律的应用场景上。

PS：javascript是弱语言，其主体表现在变量的数据类型是由它的值来决定的。同时弱语言都可以把变量中的的值进行有规律的自由变换，从而达到更快的开发效率。

自增： i++, ++i。 这两种写法一般情况下是一样的。等介于 i = i+1

i++ 先调用，后自增。

++i 先自增，后调用。

自减： i--, --i。 这两种写法一般情况下是一样的。等介于 i = i-1

I-- 先调用，后自增。

--i 先自增，后调用。

自运算：

let i = 123;

i += 123; // i = i+123;

i -= 123; // i = i-123;

i \*= 123; // i = i\*123;

i /= 123; // i = i/123;

i %= 123; // i = i%123;

3、 布尔型（boolean）:

它只有两个值：

true: 真，对

false: 假，错

一般用于判断，在JS中的自由变换，true也代表为1，false代表为0；

PS：上面的三个类型，我们一般称作标量类型。

4、数组类型（Array）

定义的格式： let 数组名 = Array(值1, 值2......);

简写方法： let 数组名 = [值1, 值2.....];

数组中元素的值可以是任意类型。

作用：一般用于主观有联的数据集进行调用与整理。

调用: 数组名[下标（键值）] ，下标是从0开始计算

多维数组：数组中的元素同样为数组时，这样的数组我们叫做多维数组。

一般两层数组，我们叫做二维数组，三层数组叫做三给维数组，超过三层统称多维数组

正常的情况下，我们设计的数组结构不宜超过三层。

5、对象类型（Object）

程序语言中所谓的对象：指是使用程序语言对大千世界所有事物的描述。

定义的格式： let 对象名 = {属性名:’值’, 方法名:具体方法}

关于JS对象：

JS原本是就基于对象开发的一种语言。JS最开始的时候是没有类这一个概念的，所以它的类与对象概念是放在一起。

JS原概念中，对象指的是拥有相同属性与方法的事物集合。

属性：静态描述对象的东西。

方法：指的是对象可以做什么事情，或者别的东西可以对对象做什么事情。

一般来讲对象使用必须进行实例化。

实例化指的是从类（集合）中选出一个例子进行研究的过程。

在JS中表对象可以直接用于功能开发，也可以用于数据保存。

在JS中数组跟对象类型的数据进行转递赋值时它们的赋值是属于引用赋值。

PS：上面两个类型，我们一般称作复合类型。

6、特殊类型

NaN：非数字。它是特殊的地方是，它不能等自己。也就是说在使用比较运算符时

NaN == NaN 得到的结果是false

null:空。

undefined:值未定义。它并不是一个错误。

null 跟 undefined可以是进行自动转换false

1. 基础语法：

PS:比较运算符：把符号左右两边的值进行比较，如果得到的结果与符号所代表的结果一致则返回true，否则返回false。

逻辑运算符：把符号两边的值进行合并运算，它分别有两运算方法。1、与运算（&&）,2、（或运算）。

逻辑与优先于逻辑或。

1. 判断语句：

格式1：if(判断条件){当判断条件成立时执行的语句}

格式2：if(判断条件){当判断条件成立时执行的语句}else{当判断条件不成立时执行的语句}

PS：格式2也叫做二路选择语句

格式3：

if(条件1){

当条件1成立时执行

}else if(条件2){

当前面的条件不成立，并且条件2成立时执行

}else if(条件3)

....else{

所有条件不成立时执行

}

1. 多路径选择语句：

格式：

switch(判断值){

case 条件1:

当判断值等于条件1时执行的语句；

break; // 跳出

case 条件2:

当判断值等于条件2时执行的语句；

break; // 跳出

.....

default:

当所有条件与判断值都不等时执行的语句。

}

Switch能写的东西if..else if..都能写，反则则不一定。但是由于两者底机制不一样。Switch的本身语句效率要比if..esle if.. 要高，所以一般情况下能用switch就使用switch。

上面两种语句也叫做流程控制语句。

1. for循环语句

循环语句的作用：

1. 用于需要重复执行的语句。
2. 用于循环计算。
3. 用于其它的渐进计算。等等

4、对数组的遍历。

格式：

for(初始化; 循环条件; 循环结后执行的语句){当循环条件成立时执行的语句}

4、while循环语句

格式：

While(循环条件){当循环条件成立时执行的语句}

当循环条件永远为真时，也就是说该循环永远不可能跳出，这样的循环我们叫做死循环。

一般情况下都要避免写出死循环。

5、do..while循环语句

格式：

do{当循环条件成立时执行的语句}while(循环条件)

这个循环最大不同的地方是，第一次进行循环时是不需要对循环条件进行判断的。

使用该循环最少也会执行一次循环里的语句。

1. 自定义函数
   1. 内置函数：alert()等等这些本身系统自带的函数，就叫做内置函数。
   2. 自定义函数：这些函数是由开发者自开发。

函数，可以认为是为方便对经常使用的一段语句进行封装的方法。

3、作用：

1、对于经常使用的语句，我们可以进行封装成为函数。方便我们有需要的时候进行调用。

2、对于非常复杂的语句，我们可以使用封装函数的方法对这些语句进行分割。

4、定义格式：

function 函数名(参数1, 参数2,...){

执行语句

}

\*\*\*\*\*\*通过参数的改变，可以改变函数的结果。

定义函数时，里面的所有语句都不会执行。

5、调用格式：

函数名(参数1, 参数2....)

调用函数时加入参数的值方法，叫做传参

传参的的参数是一个真实的值。这种参数叫做实参。

传参的的参数是一个变量。这种参数叫做形参。

必要参数：如果该函数调用时不传这个参数就函数就会错。

可选参数：如果该函数调用时不传这个参数也可以执行，一般来讲这个参数有默认值。

\*设置参数时需要注意的是必要需要设置前置位置，可选放到后置位置。

返回值：带返回值的函数，指的是这个函数最终会返回一个结果出来。

一般在函数的结束时使用return关键把结果返回出函数外部。

return一般的用法有两种：

1. 在函数中让值返回出去。

2、结束当前语句。同样如果放到最外层结程序

1. 变量的可作用域

变量的作用域:变量有效的区域。

变量直接在script下并且使用了var或者let的关键字进行定义话，这个变量在其下面的所有函数都可以自由调用。这样的变量我们叫做全局变量。

局部变量：在函数中使用关键字let或var进行定义的变量，只有在该函数域才有效。

不过，由于函数中也可以定义函数，那么这个变量在该函数里也可以看作全局变量。

1. 系统内置对象

指的系统已经进行归纳好，并且是开发中必然经常使用的基础对象。我们可以使用这些对象里的属性与方法完成我们要想功能。

由于JS是基对象的编程语言，所以在某些操作或者数据类型中的操作，JS会默认所操作直接当成对象进行操作。

1. 字符串对象
   1. 属性：
      1. lenght: 字符串对象的长度

PS：在对象调用中，使用对象属性格式 ： 对象名.属性名

* 1. 方法：
     1. indexOf: 返回原字符串中目标字符串第一次出现的位置（下标）
     2. lastIndexOf:返回原字符串中目标字符串最后一次出现的位置（下标）
     3. substr:截取字符的方法，参1是开始截取位置，参2是截取的长度
     4. Substring:截取字符的方法，参1是始截取位置，参2是截取结的位置

在对象调用中，使用对象方法格式 ： 对象名.方法名()

1. 数组对象
   1. 属性：
      1. length:返回数组元素个数
   2. 方法：
      1. concat ：返回数组对象与其它数组或数据的合并后的数组。
      2. pop:移除数组对象最后一个元素，并且返回该移除的元素。
      3. Shift：移除数组对象第一个元素，并且返回该移除的元素。
      4. push:给数组对象添加一个新的元素，在最后。
      5. reverse ：返回一个对象顺序倒过来的数组对象。
      6. sort:数组排序方法。

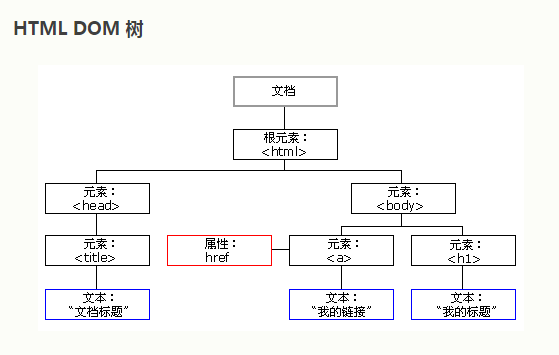
PS：实际上字符也可以当成数组进行操作。如果需要在字符串中直接获取一个字符，我们可以直接的使用下标来获取。

1. Math对象（数学对象）
   1. 属性：（主要提数学上常数）
      1. PI：返回圆周率
   2. 方法：
      1. ceil：向上取整
      2. floor：向下取整
      3. round:四舍五入到整数
      4. random:0~1之间的伪随机数，可以为零，不可能为1
2. Date对象（日期对象）
   1. 属性：（没有特殊的属性）
   2. 方法：
      1. getDate:获取日期对象中的日期（多少号）
      2. getDay:获取日期对象中的星期几，星期天是0。
      3. getFullYear:获取日期对象中的年份。
      4. getMonth ：获取日期对象中的月份，但月份是从零开始算的。
      5. \*\*getTime：获取日期对象的时间戳。时间戳从1970年开始计算到目标日期所经过的毫秒。所以我个一般来讲只要时间的记录都会使用时间戳。同时也是多个不同的语言对时间戳是通用的。
      6. toLocaleString：把日期对象转化成本地格式的日期字符串。
3. DOM与BOM

BOM：指的是浏览器操作对象。它本身可以直接操作浏览器本身各种各样的工作。

DOM：指的是本档（页面）操作对象。

DOM的操作实际上就是对页面当所有元素的获取与更改。



对于DOM的直接节点操作是需要一层层往HTML中的元素去找的。

常用的页面元素获取方法：

1. 通过ID名来获取元素对象；

格式：

document.getElementById(元素的ID名);

1. 通过标签名获取元素对象（集合）

格式：

document.getElementsByTagName(标签名)

1. 通过name的值获取元素对象（集合）

格式：

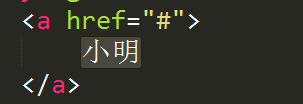
document.getElementsByName(name值);

1. 通过class名来获取元素对象（集合）

格式：

document.getElementsByClassName();

PS：在DOM直接的操作中，由于文本也属于节点，所以有时候由于html编辑的格式问题比如下面这样



直接a标签中的首节点会是一个空文本节点。

JS的事件绑定

方法1：直接在标签中加入事件属性。调用来函数来实现。

1. 实际JS功能开发的流程
   1. 功能流程：指的是用户在使用该功能时候的流程；
   2. 数据流程：指的是程序在实现该功能时数据流程；

比方：计算器的功能流程

1. 获取用户输入的两个数字
2. 使用用户选择的计算方法计算两个数字的结果
3. 把计算好的结果放到页面中对应的位置去

\*\*\*所有JS功能都有的三个核心步骤\*\*\*

1. 获取：按功能要求把需要的数据获取得到。
2. 处理：按功能要求把获取到的数据进行处理。
3. 反馈：按功能要求把处理完的结果反馈给用户或服务器。