JavaScript初学

1. 计算机基础

硬件：输入设备 ，输出设备，cpu（处理数据与运算），硬盘（永久存储数据），内存（暂时存储数据）

软件：系统软件,应用软件

数据存储：使用0.1二进制来表示数据，所有程序本质都是各种数据，

存储单位：位（bit）1b可以保存一个0或1，字节1B=8b

程序运行：硬盘-内存条-cpu（cpu运行太快了，如果只从硬盘中读取数据，浪费cpu性能，所以才使存取速度更快的内存来保存运行时的数据，内存条是电，硬盘是机械）

1. 初识JavaScript

在运行在客户端（自己的电脑）的脚本语言（不需要编译，由js解释器逐行来进行解释），也可以基于node.js技术进行服务器端编程

作用：表单动态验证，网页特效，服务端开发，桌面程序，app，游戏开发

Html/css属于描述类语言（样式方面），js属于编程类语言（行为方面）

渲染引擎：用来解析html与css，俗称内核。

Js引擎：也称js解释器，用来读取网页中的javascript代码，对其进行后处理，转换为机器语言，然后由计算机去执行。

组成：语法（ecmascript），页面文档对象（dom）对各种元素进行操作，浏览器对象模型（bom）对浏览器窗口进行操作。

1. js初体验

书写位置：行内，内嵌，外部

直接通过src直接调用即可，js中推荐使用单引号，在html中编写js大量代码时，不方便阅读，script中间不要写代码。

单行注释：// 快捷键：ctrl+/

多行注释：/\* \*/ 快捷键：shift+alt+a

1. JS输入输出语句

Alert—浏览器弹出警示框—浏览器（归属）

Console.log—浏览器控制台打印输出信息—浏览器

Prompt—浏览器弹出输入框，用户可以输入—浏览器

1. 变量

变量先声明（var），在赋值（=）

Var 变量名称； variable的缩写 输出结果：console.log(age)

初始化即声明变量的同时赋值，更新后会被覆盖，以最后一次覆盖为准

声明多个变量以逗号隔开；只声明不赋值则显示未被定义；不声明不赋值直接报错；不声明直接使用可以

1. 命名规则

由字母，数字，下划线，美元符号（就允许这两种符号）组成；严格区分大小写；不能以数字开头；不能是关键字，保留字；变量名必须有意义；遵守驼峰命名法

1. 数据类型

为了充分利用存储空间，定义不同的数据类型，js是一种弱类型或者说动态语言，不用提前声明变量类型，在程序运行过程中会被自动确定。

Infinity：无穷大；-infinity：无穷小;NaN：非数字的

转义符：\n换行，\b空格 \t缩进

1. 简单数字类型

Number：数字型-0

Boolean：布尔值类型-false（参与运算当0来看）true（参与运算当1来看）

String：字符串类型-‘’

Undefined：声明变量没有给赋值-undefined

Null：声明变量为空值

字符串类型和其他类型相拼接，最终的结果是字符串类型。

交互：用户输入（prompt），程序内部处理，输出结果（alert）

1. 获取数据类型

Typeof 加变量名称

通过控制台里的颜色，数字型蓝色，字符串型黑色，浅灰色null

字面量表示如何表达这个值。

转换为字符串方法：

1. 变量.toString()
2. String()
3. 利用+拼接字符串的方法：隐式转换

转换为数字型：

1. parselnt( )函数：转换为整数数值（地板取整）
2. parseFloat( )函数：转换为浮点数数值
3. Number( )强制转换函数:转换为数值型
4. Js隐式转换：- \* /

转换为布尔型：

1. Boolean( )函数：将其他值转换为布尔值
2. 解释性语言和编译型语言

翻译器翻译的方式有两种，编译（上完菜才吃）和解释（吃火锅，边吃边涮），两种方式之间的区别在于翻译时间点的不同，编译器是在代码执行之前进行编译，生成中间代码，解释是立即解释执行，没有中间代码生成。

标识符：开发人员为变量，属性，函数，参数取得名字。

关键字：js本身已经使用的字

保留字：预留关键字，未来可能会成为关键字

1. 运算符

也称为操作符，实现赋值，比较和执行算数运算等功能

%：取余

避免用浮点数直接参与运算，因为他在进行运算时精度远远不如整数。不要直接判断两个浮点数是否相等。

表达式都会有一个返回值，任何式子都会有返回值。

++/--num：前置递增/减运算符，先自加/减后返回值

num++/--：后置递增/减运算符，先返回原值，在自增自减

前置自增后置自增，单独使用没有区别，目的是简化代码

比较运算符：返回的是布尔类型。全等===。

逻辑运算符：返回值是布尔类型，经常用于多个条件的判断。

逻辑与都为真才为真，否则为假。

逻辑或有真则为真，否则为假

逻辑与短路运算：表达式1&&表达式2，如果1为真，则返回表达式2，如果1为假，则返回表达式1, (返回假的)

逻辑或短路运算：表达式1||表达式2，如果1为真，则返回表达式1，如果1为假，则返回表达式2,(返回真的)

前面返回后后面直接中断，不再运行，空和0，null和undefined为假其余为真

运算符优先级：小括号，一元运算符，算术运算符，关系运算符，相等运算符，逻辑运算符(与大于或)

1. 流程控制

顺序流程控制：按照代码先后顺序，依次执行，else直接跟结果，else if中间有空格

分支流程控制：根据不同的条件执行不同的代码，从而得到不同的结果

Switch语句主要针对变量设置一系列的特定值得选项时，转化开关的意思，表达式和value值必须全等才能执行，如果当前case里面没有break，则不会退出switch，继续执行下一个case

Switch（表达式）{

Case value1：

执行语句；

Break；

Default：

执行最后的语句；

}

三元表达式：有三元运算符的表达式，条件表达式？表达式1：表达式2（如果条件表达式结果为真，则返回表达式1，如果表达式结果为假返回表达式2的结果）

分支比较少用if，分支多用switch

1. 循环流层控制

可以重复执行某些代码

For循环:重复执行某些代码，通常与计数有关系for(初始化变量；条件表达式;操作表达式)；因为有计数器变量i的存在，可以重复执行不同的代码

可以使用断点调试来观察程序的运行过程

双重for循环：循环嵌套在一个循环中在定义一个循环语法结构。当里层循环执行全部完成，外层循环执行一次

1. While循环

在条件为真的情况下执行程序，直到条件为假，里面应该有计数器，初始化变量，也应该有操作表达式，完成计数器的更新，防止死循环。

Do while循环 do{循环体}while（条件表达式），如果条件表达式为真则执行循环体，否则退出循环。至少执行一次，在进行条件判断。

Continue退出当前循环，后面的直接不用执行，break直接退出循环。

变量的名称一般是名词，函数的名称一般是动词。操作符左右两侧各保留一个空格

1. 数组

将一组数据存储在单个变量名下的方式

创建数组的方式：

1. 利用new创建数组：var 数组名=new Array( )
2. 利用数组字面量创建数组:var 数组名[ ]数据用逗号分隔

索引：用来访问数组元素的序号数组下标从0开始，数组名[索引号]

遍历数组：遍历是注意<=索引号的准确范围

数组名.length可以访问数组元素的数量（数组长度）

数组新增元素：（1）修改length长度

1. 不能直接给数组名赋值，否则会覆盖以前的数据
2. 判断是否为数组
3. Instanceof Array运算符
4. Array.isArry(arr)

添加删除数组的方法：push（）在我们数组的末尾添加一个或多个数组元素，push()里面直接写参数即可，返回的结果是新数组的长度，原数组也会发生变化；unshift在数组开头添加元素；pop()删除数组最后一个元素，把数组长度减1无参数、修改原数组；shift()删除数组的第一个元素，数组长度减1无参数，并返回第一个元素的值。

翻转数组：arr.reverse()即可。

数组排序：arr.sort()冒泡排序

arr1.sort(function (a, b) {

return a - b;

});

升序为a-b，降序为b-a

数组索引方法：indexOf()数组中查找给定元素的第一个索引，如果存在索引号返回，如果不存在，则返回-1；lastIndexOf()在数组中的最后一个索引，如果存在索引号返回，如果不存在则返回-1。（筛选重复元素时使用）

数组转换为字符串：toString( )将我们的数组转换为字符串；join（‘分隔符’）返回的都是一个字符串

1. 字符串对象

基本包装类型就是把简单数据类型包装成为复杂数据类型，这样基本数据类型就有了属性和方法

对象和复杂数据类型才有属性和方法。通过基本包装类型就把简单数据类型包装成为复杂数据类型

字符串的不可变指的是里面的值不可变，虽然看上去可以改变内容，但其实是地址变了，内存中新开辟了以个内存空间。由于字符串的不可变，在大量拼接字符串的时候会有效率问题。

根据字符返回位置：字符串的所有方法，都不会修改字符串本身（字符串是不可变的），操作完成会返回一个新的字符串。str.indexOf(‘要查找的字符’，[起始的位置])

charAt（index）-返回指定位置的字符-str.chartAt()

charCodeAt(index)-获取指定位置的ASC||码-str.charCodeAt()-判断用户按下了哪个键

str[index]-获取指定位置字符串。

字符串操作方法：concat(str1,str2,str3):用于连接多个字符串，等效于+

Substr(start,length):从开始位置开始索引，length取得是个数

Slice（start，end）:从start位置开始，截取到end位置

Substring（start，end）：基本和slice相同，但是不接收负数

Replac（‘替换的字符串’，‘替换的字符’）

Split(‘分隔符’)：把字符转换为数组

1. 简单数据类型和复杂数据类型

简单类型又叫基本数据类型或者值类型，复杂类型又叫引用类型；

值类型：在存储时变量中存储的是值本身

引用类型:在存储变量中存储的仅仅是地址（引用），因此叫做引用数据类型

简单数据类型存放到栈中，里面直接开辟一个空间存放的是值；复杂数据类型存放到堆中，首先在栈里存放地址，十六进制表示，然后这个地址指向堆里面的地址。

1. 传参

（1）简单类型传参：函数的形参也可以看做是一个变量，当我们把一个值类型变量作为参数传递给函数的形参时，其实是把变量在栈空间里的值复制了一份给形参，name在方法内部对形参做任何修改，都不会影响外部变量。

（2）复杂数据类型传参：函数的形参也可以看做是一个变量，当我们把引用类型变量传给形参时，其实是把变量在栈空间里保存的堆地址复制给了形参，形参和实参其实保存的是同一个堆地址，所以操作的是同一个对象。

1. 对象

对象是一个具体的事物，万物皆对象，js中对象表达结构更清晰更强大

Js中，对象本质是一组无序相关属性和方法的集合，例如字符串，数值，数组，函数等

属性：事务的特征

方法：事物的行为

1. 函数

声明函数然后调用函数。（1）Function是声明函数的关键字，全部小写；函数是做某件事，函数名必须是动词；函数不调用，自己不执行；调用函数时别忘了加小括号。

（2）函数表达式声明方式 var 变量名=function（）{ }；

形参和实参：function 函数名（形参），调用函数里面的参数为实参，实际的参数。

如果实参的个数多于形参的个数，会取到形参的个数；如果实参个数小于形参个数，多于的形参定义为undefined，最终结果就是NAN

函数返回值return语句：最终的结果返回给函数的调用者，只要函数遇到return就把后面的结果返回给（只能返回一个值，多个值以最后一个为准）给调用者，并且函数会被终结，后面的代码不会被执行；在实际开发中，通常使用一个变量来接收函数的返回结果，使用更简单。

如果没有return则返回undefined。

Arguments：当不确定有多少个参数传递的时候可以用arguments来获取，即函数的一个内置对象，arguments对象中存储了传递的所有实参。

函数可以调用另外一个函数。

1. Js中的作用域

所用到的名字并不是有效和可用的，变量在某个范围内起作用和效果，而限定这个名字可用性的代码范围就是这个名字的作用域。作用域提高了程序逻辑的局部性，增强了程序的可靠性，减少了名字冲突。

1. 全局作用域：整个script标签，或者是一个单独的js文件，
2. 局部作用域（函数作用域）：在函数内部就是局部作用域，只在函数内部起效果和作用

变量作用域：（1） 全局变量：在全局作用域下的变量

（2） 局部变量：在局部（函数）作用域下的变量

如果在函数内部没有直接声明直接赋值的变量也属于全局变量；从执行效率来看，全局变量只有浏览器关闭时才会销毁，比较占内存资源。局部变量程序执行完毕就会销毁，比较节约内存资源

作用域链:只要是代码，就会有作用域链；写在函数内部的局部作用域；如果函数中还有函数，那么这个作用域中就可以诞生一个作用域；根据内部函数可以访问外部函数变量的这种机制，用链式查找决定那些数据能被内部函数访问，就称做作用域链（就近原则，层级关系就近，站在目标出发，一层一层往外找）。

1. Js中的预解析

Js引擎运行js分为两步：（1）预解析：js引擎会把js里面所有的var还有function提升到当前作用域的最前面（2）代码执行：按照代码书写的顺序从上往下执行。

预解析分为变量预解析（变量声明）和函数预解析（函数提升），变量提升就是把所有的变量声明提升到当前作用域的最前面，不提升赋值操作。函数提升就是把所有的函数声明把当前作用域提升到最前面；

var a = b = c = 9;相当于var a，b和c为全局变量

1. 对象

对象是具体的事务，对象是一组无序的相关属性和方法的集合

属性：事物的特征，在对象中用属性来表示（常用名词）；方法：事务的行为，在对象中用方法来表示（常用动词）

为什么需要对象：保存一个值时，可以使用变量；保存多个值时，可以使用数组；保存一个人的完整信息，可以使用对象；让结构更加清晰更加强大。

1. 创建对象
2. 利用字面量来创建对象：{ }里面包含对象的属性和方法
3. 利用new object来创建对象：var obj=new object（）；利用=赋值的方法，添加对象的属性和方法，每个属性之间用分号结束，属性和方法采用键值对的形式。方法后面跟的是匿名函数
4. 利用构造函数来创建对象：前面两种只能创建一个对象，这个函数里面封装的是对象，把对象里面相同的属性和方法抽象出来封装到函数里面。

调用属性：（1）对象名.属性名.理解为的即可（2）对象名[‘属性名’]

调用方法：（1）对象名.方法名()必须加小括号

1. 变量属性方法的区别

变量和属性都是用来存储数据的；变量单独声明并赋值，使用时直接写变量名，单独存在；属性是在对象里面的，是不需要声明的使用时必须是对象.属性；函数和方法都是实现某种功能，做某件事，外面的叫函数单独声明调用，里面的叫方法调用的时候对象.方法( )。

1. 利用构造函数创建对象

前面的方式只能创建一个，里面的属性和方法是大量相同的，利用函数把重复的代码封装进去，就是把对象封装进去，构造函数名字首字母大写，不需要return就会返回结果，调用构造函数必须使用new，属性和方法前面必须添加this

语法格式：function 构造函数名（）{

This.属性=值

}

使用时用new 构造函数名（）；

1. 构造函数与对象之间的联系

构造函数相当于某一大类，对象实实在在存在相当于某一个，通过构造函数创建对象的过程我们也称为对象的实例化。

遇见New之后创建一个空的对象，this都会指向这个对象，执行构造函数代码，返回这个对象（相当于return）

1. 遍历对象

For in用于对数组或者对象的属性进行循环操作。

For（变量 in 对象），（）输出的是属性名，[ ]里面输出的是属性值

1. 内置对象

自定义对象，内置对象，浏览器对象

内置对象就是js语言自带的一些对象，提供基本属性和方法，帮助我们快速开发

1. Mdn

先查看功能，再看参数的意义和类型，最后查看返回值的类型和意义

1. w3c
2. 数学对象

Math数学对象不是一个构造对象，不需要new来调用，直接使用里面的方法和属性即可

Math.abs( )取绝对值，math.floor( )向下取整，math.ceil( )向上取整，math.round( )就近取整其他都是四舍五入，.5往大了取

Random（ ）随机返回一个浮点数，在0-1之间