Js中推荐使用单引号，

Js是脚本语言和解释语言，java是编译语言。

起名：不能用关键字，标识符起名。不用name 起名，关键字不能起名，for，

不能用数字开头，首字母小写，后面单词首字母大写。

alert(num2.length);只看字符串的长度，数字型看不了。

+ 加号：出现字符型就是拼接，没有字符型就是运算。加布尔型会把布尔型隐式转成数字。

- 减号：出现有数字的字符串会隐式转换为数字型，进行运算，/号和\*号也同理。

没有给变量赋值，变量显示是undefined结果，可以和字符串拼接，但和数字相加是NaN.

给变量赋值null,可以和字符串拼接，也可以和数字相加，null默认是0

基本数据类型：number string bloom undefined null.

Prompt返回的值是字符串型。

工具函数： isNaN(); typeof();isNaN（）：工具函数，判断是否是非数字，返回bloom。

字符串型： 和别的数据类型相加（拼接）会得到字符串类型数据。（隐式转换）

数字型： 字符型需要用+号时用number转换，减号，乘号，除号直接用，不需要转换，因 为有隐式转换

Boolean型：把空的数据转换为flase，非空转为true.

Var str = “”; var flag = false; var num2 = null; var a = undefined; var num = 0;

NaN也算.

以上都泛指的空。

Ture + 1 得2.

String(num)不影响变量本身的数据类型,下一行显示num的类型,还是他本身number的类型,

++a, a是先有返回值,返给了a, 改变了变量a 本身的值 .

a++ 是先返回值,再运算. ++a是先运算,再返回值 .

比较运算符: 返回值是boolean类型.

默认会把字符型的转换为数值型.

如果和数字型相比较,和会把布尔类型换成数字型,存在隐形转换.

数字型和数字的字符串类型比较，会把字符串换成数字型。在比较运算符中偏向数字。

===全等要求数据类型也相等,不会默认转换了.(单=是赋值,==是判断).

逻辑运算符:&&逻辑与有一个false就是false

||逻辑或有一个ture就true

!逻辑非,取相反值.

赋值运算符: = , ++ , - - , +=(可能出现拼接) , -= , \*= , /= , %=(余等)

运算优先级:

括号里面的级别最高,一元运算符有++,!, 逻辑运算符最低,逻辑与&&比逻辑或||级别高.

比较运算优选于逻辑运算.

Switch:

注意：switch后的变量必须和case的值全等。

没有break会发生遍历，不管满不满足case中条件。

If和switch的区别：if用于范围，switch用于几个值选择题。

用户输入的数据进行运算需要转换数字型。

 for中要用到的运算的变量要在for循环之前进行初始化。

转义符号要加引号。

Continue:是跳出本次循环，直接到i++中，就不管下面的代码了。

Break:是跳出整个循环。

封闭函数：

Return后面的代码不会被执行

调用函数时直接输入具体值，比如直接把数组[32,51,87,65,5]放在括号中

函数中不用把返回的值给return.调用是把函数的值给一个变量。

在块级作用域中不声明变量直接赋值，或者也没赋值的变量，为全局变量。

默认值是undefined.

拓展：可以在形参中设置默认值，没有导入实参时，会用到默认值，导入实参会 覆盖默认值，function sum(num1,num2=100) {};

复制函数：var 变量名 = 函数名;两个相等。

注意：作用域存在预解析。在声明变量之前调用，会存在变量提升，此时变量值 为Undefined.

1. 匿名函数，var fun = function() {...}只能在匿名函数后调用函数，因为 是 变 量名，在函数之前提升，存在变量提升，提升后值是undefined， 会 报 错。
2. 命名函数也存在函数提升，会把整个函数放在作用域最前面，不会影响 函数值。

有一个报错undefined没有找到变量，后面的代码就不会运行。

对象：

创建对象就是对象的实例化。

对象中调用属性也要带对象名，否则会成为全局变量，默认值是undefined.

New出来的都是对象Object,或者return{}返回值用{}包着也属于对象，

Var obj1=~~ 赋给一个对象名就创建出来一个对象了。

Js对象分为两种：属性和方法都是定死的。（不需要使用new关键字的）

属性和方法会改变，动态对象。（需要使用new关键字创建）

对象遍历：for(var k in 对象名) { console.log(对象名[k] }; //和别的for遍历相同

创建数组：

var 数组名 = new Array() ；

创建对象：

var 对象名 = new Object() ；

创建数组：

var 数组名 = []；

创建对象：

var 对象名 = {}；

创建数组：

var 数组名 = ['小白','小黑','大黄','瑞奇'];

创建对象：

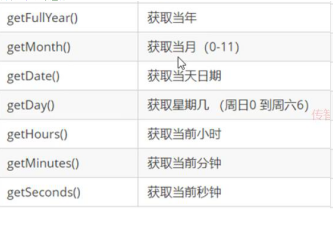
var 对象名 = {name:'小白' , age:16 };

Date

New出来的日期date，打印出来是统一格式，有参数返回的是输入的时间

Var date1=new Date();出来的是现在的时间; //在new之前放个+号,输出dat

Var date2=new Date(‘2020-1-9 11:00’);返回的是输入的时间。 //e1是毫秒数。



获取毫秒数的方法（1）date.valueOf (); //获取距1970年1.1日的毫秒数

1. var date1 = +new Date (); console.log (date1);
2. Console.log (Date.now ());

（4）date.getTime;

数组

判断是否是数组的工具函数，（1）数组名 instanceof Array

1. Array.isArray ( 数组名 )

加分隔符：arr.join(‘|’);

连接多个数组：arr.concat( [ 1,2,3,4,5,6 ] );整合成一个新数组。

排序：arr.sort();可以给数字和字母进行排序，默认升序

arr.sort(function (a, b) {

            return b - a;

        }); //降序

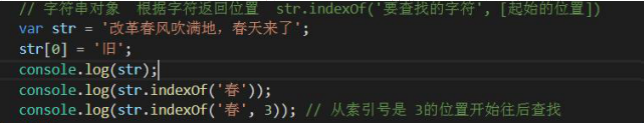
添加，push（后加）unshift（前加）返回值是数组的长度。例：arr.push(‘red’);

删除，pop（后删一个） shift（前删）返回值是删除的值。 例:arr.pop();

Splice（1，2）从索引1开始删2个元素，在原数组上删,返回删除的值。

Slice(2,5) 从索引2开始截取到索引值5，返回截取的数组。

查找索引值：arr.indexOf ( ‘值’ ，索引号) 从指定索引号开始查找



字符串

返回指定字符的索引值 str . indexOf（‘字符’）

返回指定位置的字符 Str . charAt ( index )：;

获取指定位置的ASC||码 str .charCodeAt ( index )：

获取指定位置处的字符 str. [ index ]：

连接字符串 str.concat(str1,str2);

字符串长度 length

转化为小写 toLowerCase()

转化为大写 toUpperCase()

substr(0,1) 返回从下标为0的字符开始长度为2 的字符串

str.substring(1,4)  返回从1到4的子串

提取字符串中的一部分并返回一个新的字符串，通substring() str.splice()

将字符串分成字符串数组 str.split()

将数组组成字符串 str.join( ‘’ )

replace返回的是整个被替换过的新数组，除了被替换的内容，其它内容不变，不在原数组上修改。

sort排序是在原数组上修改。

split和join括号里面必须有双引号，把有空格的字符串转换成数组，也要有空格。反之join亦然。