Se usan o llaves {

Corchetes [

Paréntesis ()

Alertas

Tipos de alertas

* Alert (“hola mundo”)

//abre una ventana del navegador e imprime un elemento

* Prompt abre una pequeña ventana en el navegador para el usuario

Introduzca datos

* Console.log .log (“mira aquí”)

//lanza un mensaje en la consola desarrollo y se utiliza para testear código

Declarar una variable

Var nombre variable= valor;

Puedes imprimir con

Console.log (nombre variable);

Tipos de variables

* miVariable =200; //numéricos
* caracteres
* miVariable=” Hola”; //String
* miVariable=’Hola’; //String
* miVariable=”200”;//srtring

tipos de datos booleanos

estos tipos de datos se usan en sentencias condicionales

if las cuales admiten dos valores(verdadero/Falso)

arrays =es una variable que sirve para guardar varios datos dentro de una misma variable.

Var nombreArray= “Maria”, “Carlos ”,”pepe” ;

Las posiciones los array parte de la posición 0

Variables

Son un dato guardado en memoria

Se puede sobre escribir

Ejemplo \_numero=2;

$nombre=’pepe’

Ejemplos

var cantidad=10;

undefined

console.log(cantidad);

VM962:1 10

undefined

var zapatos=16;

console.log(zapatos);

zapatos=20;

console.log(zapatos);

VM1170:2 16

VM1170:4 20

undefined

var caja=25;

console.log(caja);

VM1286:2 25

undefined

var -caja=12;

error

VM1408:1 Uncaught SyntaxError: Unexpected token '-'

var\_caja=12;

12

var\_caja=1;

1

pelota=2

2

Funciones

Las funciones se declaran de la siguiente manera.

Funtion nombre\_funcion(){

Var x=500;

}

// error pot antes de inprimir la variable x debe estar fuera de la funcion

Console.log(x)

Variables global o local

Esta se pueden declarar/ Re declarar

Con **var se permite el uso del hoisting que es la forma de js para llamar a una variable función al inicio del scope antes de que se ejecute el código**

**La variable let está limitada que funcione dentro de un bloque de porción de código que este entre llaves {}**

**Esta puede ser actualizada pero no Re declarada**

**Las variables const /constantes mantienen su valor durante todo código no pueden re declararse ni actualizarse**

**Tipos y usos de los operadores**

**Operadores = son símbolos que indican como se debe manipular los datos**

**También tenemos los operadores aritméticos como**

* **Suma (+)**
* **Resta (-)**
* **Multiplicación (\*)**
* **División (/)**

**Ejemplo**

**var a=500;**

**var b=500;**

**var resultado;**

**var resultado=a+b;**

**console.log(resultado);**

**ejemplo 2**

**var a=500;**

**var b=500;**

**var resultado;**

**var resultado=a\*b;**

**console.log(resultado);**

**claro que si una de las variables es un String lo toma como concatena las variables a y b**

**var a=500;**

**var b='500';**

**var resultado;**

**var resultado=a+b;**

**console.log(resultado);**

**500500**

**Otro caso**

**var a=500;**

**var b='b';**

**var resultado;**

**var resultado=a\*b;**

**console.log(resultado);**

**NaN**

**Variables de incremento y decremento**

**a=a+1;**

**a+=|1**

**a++**

**a=a+1;**

**a+=|1**

**a—también esta el decremento**

**significado de operadores**

* **=asignación**
* **==igualdad se usa para ver dos valores son iguales**
* **=== igualdad estricta encaso que una de esas condiciones no se cumpla el resultado es false**
* **¡= distinto compara si dos valores son distintos**
* **< menor que**
* **> mayor que**
* **<= menor igual**
* **>= mayor o igual**
* **&& implica que se deben cumplir las dos condiciones**
* **|| implica que se debe cumplir o 1 condición o la otra**

**Operaciones con String**