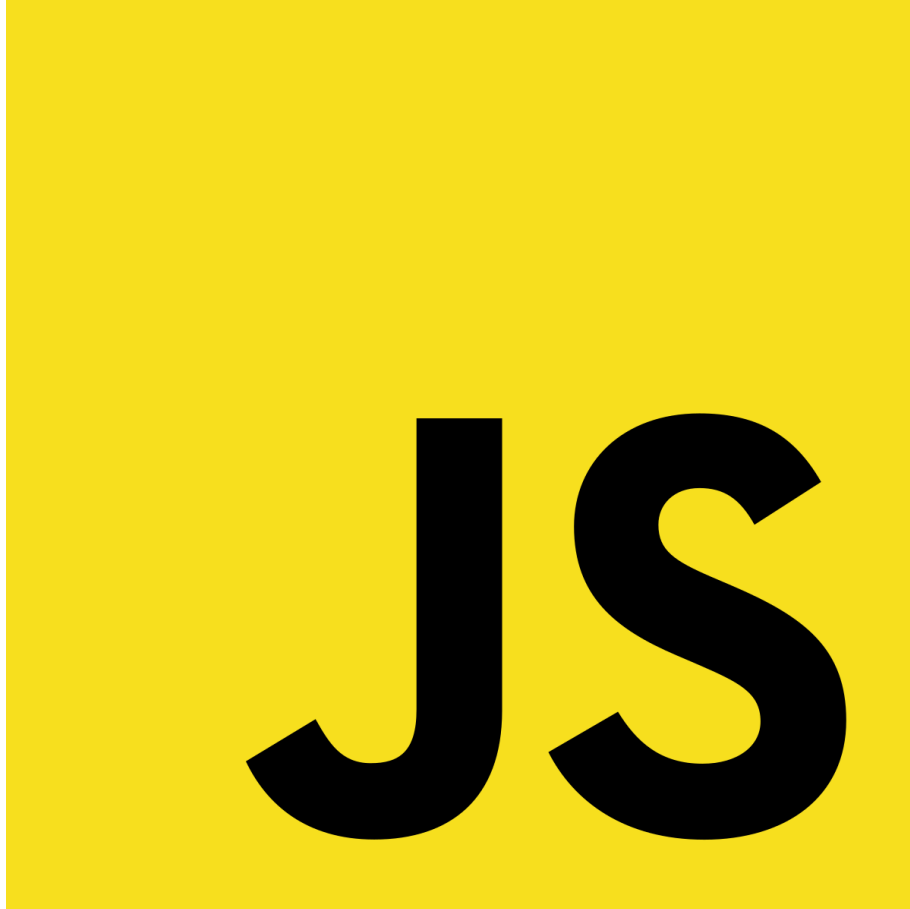


# MANUAL BASICO JAVASCRIPT



## Índice

Índice .....	2
Como usar javascript .....	4
1) Primer programa .....	4
2) Sumar valores .....	5
3) Modulo o restante de una division .....	5
4) Aumentar o disminuir valores .....	5
5) Longitud de cadenas .....	6
6) Índice de caracteres .....	6
7) Inmutabilidad de caracteres .....	6
8) Array (arreglos) .....	7
9) Array multidimensional (matriz), push, pop, shift y unshift .....	8
10) Funciones .....	9
11) Funciones 2 (numérico) .....	9
12) Variable local y global .....	10
13) Return en una función .....	11
14) Ejercicio lista (Queue) .....	12
15) Booleanos: true y false .....	12
16) Operadores de igualdad y desigualdad .....	13
17) Operadores mayores y menores .....	14
18) Objetos .....	16
19) Objetos complejos .....	18
20) Números aleatorios .....	19
21) Función parseInt() .....	21
22) Operador ternario .....	22
23) Variable var y let .....	22
24) Variable const .....	24
25) Objetos inmutables .....	25
26) Funciones flecha .....	25
27) Valores por defecto en parámetros en funciones .....	27
28) Operador rest .....	27

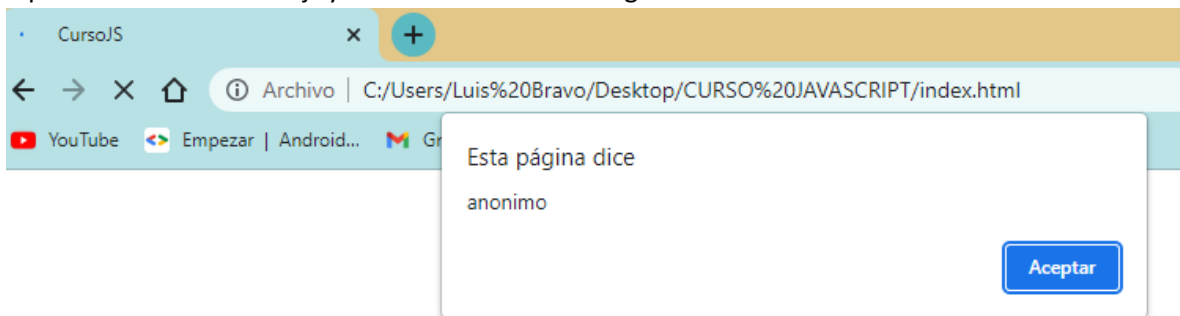
29)	Operador spread .....	27
30)	Destructuracion.....	28
31)	Objetos con forma concisa.....	32
32)	Plantillas de cadenas .....	33
33)	Metodos .....	34
34)	Clases.....	35
35)	Getter y Setter.....	36

## Como usar javascript

Si utilizas un editor de texto como VScode puedes crear un archivo HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CursoJS</title>
</head>
<body>
  <script src="AQUICOLOCASELNOMBREDETUARCHIVO.js"></script>
</body>
</html>
```

Y puedes crear archivos .js y correrlos desde el navegador



```
JS 01Hola.js
JS 02Suma.js
JS 03modulo.js
```

### 1) Primer programa

```
//Titulo: 01Hola.js
window.alert("hola") //usamos window alert para mandar una alerta en la
pagina
```

## 2) Sumar valores

```
//Titulo: 02Suma.js  
var a=5;  
var b=2;  
var c = a+b;  
window.alert(c);
```

## 3) Modulo o restante de una division

```
//Titulo: 03modulo.js  
var a=10;  
var b=4;  
var c = a/b;  
window.alert("La divisiones: "+c);  
c= a % b;  
window.alert("El modulo o resto es "+c);
```

## 4) Aumentar o disminuir valores

```
//Titulo: 04aumentarDisminuirValorEnUno.js  
//aumento  
var i=10;  
var e=20;  
    //opcion 1  
    i=i+1;  
window.alert(i);  
  
    //opcion 2  
    e++;  
window.alert(e);  
  
//Disminuir  
  
i=i-1;  
e--;  
window.alert(i);  
window.alert(e);
```

## 5) Longitud de cadenas

```
//Titulo: 05LongitudCadena.js
var mensaje = "Soy luis";
//java script ceunta los espacios como caracter por lo que seria 8
window.alert("La longitud del mensaje es "+mensaje.length)
```

## 6) Índice de caracteres

```
//Titulo: 06IndiceDeCaracteres.js
var nombre="Luis Francisco";

//por lo tanto el mensaje sera L
window.alert("caracter posicion 1 " +nombre[0]);

//por lo tanto el mensaje sera F
window.alert("caracter posicion 6 " +nombre[5]);
//si ponemos una poscicion que no existe en el numero de caracteres nos
saldra undifine o indefinido

//Ultima posicion
window.alert("caracter posicion ultima " +nombre[nombre.length-1]);
```

## 7) Inmutabilidad de caracteres

```
//Titulo: 07InmutabilidadDeCaracteres.js
var cadena="Jola Mundo"

window.alert(cadena);
/*aquí tratamos de indicar que se hará el cambio de la j por la L pero no se
puede ya que las cadenas son
inmutables*/
cadena[0]="H";

window.alert(cadena); //Jola Mundo
```

## 8) Array (arreglos)

```
//Titulo: 08Array01

var array=["Luis",17,11,96];

window.alert("Primer array en javascript "+array);

//array anidado

var array2=[["Luis",17,11,96] , ["Paola",18,09,98]];
window.alert("Array anidado "+array2);

//imprimir indices

window.alert("Posicion 1 de array =" +array[0]);
window.alert("Posicion 2 de array =" +array[1]);
window.alert("Posicion 3 de array =" +array[2]);

//sumatoria de arreglo de numero
var suma= array[1]+array[2]+array[3];
window.alert("El resultado de sumar el array es:" +suma); // el resultado fue
124 por sumar 17+11+96

//sumatoria con cadena
suma= array[0]+array[2]+array[3];
window.alert("El resultado de sumar el array es:" +suma); /* el resultado fue
Luis1196 porque no suma cadena de
caracteres*/

//Actualizar valor o agregar
array[4]=10; //agregamos posicion 4 10
array[3]=1996; //modificamos posicion 3 de 96 a 1996
window.alert("Primer array modificado "+array);

//agregar o modificar a un arreglo anidado
var array3=[10,20,30];
window.alert("array anidado sin modificar "+array3);
array3[0]=[1,2,3];
window.alert("array anidado modificado "+array3);
```

## 9) Array multidimensional (matriz), push, pop, shift y unshift

```
//Titulo: 09ArrayMatriz.js
//EJEMPLO 1 imprimir matriz

var array=[[1,2,3],[4,5,6],[7,8,9]]

//imprimimos la pocision 00 y 11

window.alert(array[0][0]);
window.alert(array[1][1]);

//EJEMPLO 2 metodo PUSH Y POP, AGREGAR Y QUITAR ULTIMA POSICION

//Agregar UN ELEMENTO A UN ARRAY CON .PUSH Y REMOVE CON .POP
var estaciones=["invierno","Primavera","Verano"];
window.alert(estaciones);
estaciones.push("otoño");
window.alert(estaciones);
// si solo usamos nombreArray.pop(); solo eleiminamos el ultimo factor, pero
si ademas lo asignamos a una
//variable podemos usar esa variable despues por ejemplo en estacion4 se
guardo otoño
var estacion4=estaciones.pop();
window.alert(estaciones);
window.alert(estacion4);

//EJEMPLO 3 METODO SHIFT UNSHIFT AGREGAR Y QUITAR PRIMERA POSICION
/**/
var array3= ["invierno","Primavera","Verano","Otoño"];
window.alert(array3);
array3.shift(); //se elimina invierno
window.alert(array3);
array3.unshift("invierno");//regresamos invierno
window.alert(array3);
```



## 10) Funciones

```
//Titulo: 10Funciones.js

function mostrarMensaje(){
    window.alert("Hola mundo");
}

mostrarMensaje();
mostrarMensaje();
mostrarMensaje();
//se mostrara 3 veces el mensaje
```

## 11) Funciones 2 (numérico)

```
//Titulo: 11Funciones2.js
//entre parentesis vemos a y b son parametros
function sumar(a,b){
    var suma = a + b;
    window.alert("El resultado es "+suma);
}

sumar(5,3);

function concatenarCadena(cadena1,cadena2,cadena3){
    window.alert(cadena1+" "+cadena2+" "+cadena3);
}

concatenarCadena("Estoy","Bien","Cabron");
```

## 12) Variable local y global

```
//Titulo: 12VaribaleGlobalYLocal.js

var varGlobal=5;
var varGlobal2=10;
window.alert(varGlobal);
function miFuncion(){
    var varLocal=6;
    window.alert(varGlobal);
    varGlobal=7;
    window.alert(varLocal);
    window.alert(varGlobal2);
    window.alert(varGlobal);
}

miFuncion();
window.alert(varGlobal2);
window.alert(varGlobal);

window.alert(varLocal);//esta avrible no funciona porque se definio dentro
de
//la funcion
```

### 13) Return en una función

```
//Titulo: 13RetornarEnUnaFuncion.js

function sumar(a,b){
    return a+b;
    // indicamos que es lo que se va a retornar en este caso seria la
    sumatoria
}

window.alert("La suma es: "+sumar(5,3)); //mandamos los valores y como
llamamos
//la funcion y la funcion nos regresa la sumatoria de los numeros

//Asignar valor retornado
var resultado = sumar(5,4);
window.alert("La suma es: "+resultado);

function restar(a,b){
    window.alert("La resta es "+(a-b));
    //en este caso estamos imprimiendo y no regresamos nada
}

window.alert("La resta es "+restar(5,3)); // como no tenemos return en
nuestra
//funcion no podemos imprimir y nos saldra el mensaje undefined es decir
//indefinido
```

## 14) Ejercicio lista (Queue)

```
//Titulo: 14EjercicioFilaQueue.js

/*Crea un programa que modifique un arreglo(array) eliminando el primer
valor y
agregando uno nuevo ademas de mostrar el orden antes y despues */

function proximoEnLaFila(arreglo,lugar){
    arreglo.push(lugar); //Agregamos lugar al final del array
    return arreglo.shift(); //Quitamos y retornamos el primer valor del
array
}
//definimos nuestro arreglo
var array=[1,2,3,4,5];
//imprimimos nuestro array antes de modificar
window.alert("Antes"+ JSON.stringify(array));
//ConJSON.stringify convertimos el array en una cadena

//mandamos array, nuevo lugar y tambien imprimimos el lugar que se elimina
window.alert(proximoEnLaFila(array,6));

//imprimimos el nuevo array
window.alert("Despues"+ JSON.stringify(array));
```

## 15) Booleanos: true y false

```
//Titulo: 15Booleanos.js

window.alert(true); //forma correcta de escribir true
window.alert(false); //forma correcta false

window.alert(True); //forma incorrecta marca error o no muestra nada
```

## 16) Operadores de igualdad y desigualdad

```
//Titulo: 16OperadoresIgualDesigual.js

//Operador de igualdad ==
function igualdad(){
window.alert(5==5); //true
window.alert(5==6); //false
window.alert(5=="5"); //true aunque este en comillas
window.alert("Hola"=="Hola"); //true
window.alert("Hola"=="Jola"); //false
window.alert([1,2,3]==[1,2,3]); //aunque son iguales marca false
//no podemos usar el operador para comparar array
}

//Operador de igualdad estricta ===
function igualdadEstricta(){
    window.alert(9===9); //true
    window.alert(9===9); //false a diferencia de la igualdad simple el
operador no los reconoce como iguales
}

//operador de desigualdad != , en pocas palabras es lo inverso al operador
de igualdad
function desigualdad(){
    window.alert(5!=5); //false por que si son iguales
    window.alert(5!=6); //true porque son diferentes
    window.alert(5!="5"); //false
    window.alert("Hola"!="Hola"); //false
    window.alert("Hola"!="Jola"); //true
    window.alert([1,2,3]![1,2,3]); //true
}

function desigualdadEstricta(){
    window.alert(5!==5); //false por que si son iguales
    window.alert(5!==6); //true porque son diferentes
    window.alert(5!==5); //true porque no esta convirtiendo el numero
entre comillas
    window.alert("Hola"!==5); //false
    window.alert("Hola"!==5); //true
    window.alert([1,2,3]!==[1,2,3]); //true
}

desigualdadEstricta();
```

## 17) Operadores mayores y menores

```
//Titulo: 17OperadoresMayorMenor.js

function mayorQue(){
    window.alert(5>4); //true
    window.alert(5>6); //false
    window.alert(5>5); //false
    window.alert("A">"A");//False son iguales
    window.alert("A">"B"); //false porque en el alfabeto B esta despues por lo
    cual es mayor
    window.alert("B">"A"); //true
    window.alert("ACB">"ABC"); //True porque como A esta en primera pocision
    compara la segunda y tenemos
    // C que es mayor que B
    window.alert("AB">"A"); //true porque compara la segunda posicion y como
    no hay lo toma como 0
    window.alert("Mundo">"Hola"); //true M esta despues de H en el alfabeto
    window.alert("M">"H"); //true M esta despues de H en el alfabeto
    window.alert("Fin Mayor que");
    window.alert();
}

function menorQue(){
    window.alert(5<4); //false
    window.alert(5<6); //true
    window.alert(5<5); // false
    window.alert("A"<"A");//False son iguales
    window.alert("A"<"B"); //true porque en el alfabeto B esta despues por
    lo cual es mayor
    window.alert("B"<"A"); //false
    window.alert("ACB"<"ABC"); //false porque como A esta en primera
    pocision compara la segunda y tenemos
    // C que es mayor que B
    window.alert("AB"<"A"); //false porque compara la segunda posicion y
    como no hay lo toma como 0
    window.alert("Mundo"<"Hola"); //false M esta despues de H en el alfabeto
    window.alert("M"<"H"); //false M esta despues de H en el alfabeto
    window.alert("Fin Menor que");
    window.alert();
}

function mayorIgualQue(){
    window.alert(5>=4); //true
```

```
    window.alert(5>=6); //false
    window.alert(5>=5); //true
    window.alert("A">="A");//True
    window.alert("A">="B"); //false porque en el alfabeto B esta despues por
lo cual es mayor
    window.alert("B">="A"); //true
    window.alert("ACB">="ABC"); //True porque como A esta en primera
posicion compara la segunda y tenemos
    // C que es mayor que B
    window.alert("AB">="A"); //true porque compara la segunda posicion y
como no hay lo toma como 0
    window.alert("Mundo">="Hola"); //true M esta despues de H en el alfabeto
    window.alert("M">="H"); //true M esta despues de H en el alfabeto
    window.alert("Fin Mayor igual que");
    window.alert();
}

function menorIgualQue(){
    window.alert(5<=4); //false
    window.alert(5<=6); //true
    window.alert(5<=5); //true
    window.alert("A"<="A");//true son iguales
    window.alert("A"<="B"); //true porque en el alfabeto B esta despues por
lo cual es mayor
    window.alert("B"<="A"); //false
    window.alert("ACB"<="ABC"); //false porque como A esta en primera
posicion compara la segunda y tenemos
    // C que es mayor que B
    window.alert("AB"<="A"); //false porque compara la segunda posición y
como no hay lo toma como 0
    window.alert("Mundo"<="Hola"); //false M esta despues de H en el
alfabeto
    window.alert("M"<="H"); //false M esta despues de H en el alfabeto
    window.alert("Fin Menor igual que");
    window.alert();
}
//mayorQue();
//menorQue();
//mayorIgualQue();
menorIgualQue();
```

## 18) Objetos

```
//Titulo:19Objetos.js

//crear un objeto con atributos
var miPerro = {
  "nombre": "Luna",
  "edad": 8,
  "peso": 5.5,
  "raza": "Schnauzer",
  1 : ["bonita","gordita","alegre"]
};

//consulta con punto nombreObjeto.Atributo

window.alert("Nombre: "+miPerro.nombre);

//consulta con corchete nombreObjeto[Atributo] si un atributo es numero
usamos este

window.alert("Edad en años normales: "+miPerro["edad"]);

//usar y actualizar propiedad

miPerro.nombre="Luna Hdez Cortes";
window.alert("Nombre: "+miPerro.nombre); //luna
var edadAnosPerro= miPerro.edad * 7; //54
miPerro.edad = edadAnosPerro;
window.alert("Edad en años perro: "+miPerro.edad); //54
window.alert("Caracteristica: "+miPerro[1]); //bonita,gordita,alegre
var caracteristicaEliminada=miPerro[1].pop(); //alegre eliminada
window.alert(caracteristicaEliminada); //alegre
miPerro[1].push("Dormilona"); //dormilona
window.alert("Caracteristica: "+miPerro[1]); //bonita,gordita,dormilona
//Lo mismo aplica con unshift y shift al tratarse de un array

//agregar y eliminar atributo

miPerro.comidaFav = "pollito"; //agregar propiedad
window.alert(miPerro.comidaFav); //pollito
delete miPerro.comidaFav; //eliminar propiedad
window.alert(miPerro.comidaFav); //undefined
```



```
//Verificar si un valor existe

window.alert(miPerro.hasOwnProperty(1)); //true
window.alert(miPerro.hasOwnProperty("peso")); //true
window.alert(miPerro.hasOwnProperty(2)); //false
window.alert(miPerro.hasOwnProperty("poder")); //false

//ejemplo con condicional
function verificarPropiedad(obj, propiedad){
    if(obj.hasOwnProperty(propiedad)){
        return "propiedad: " + obj[propiedad];
    } else {
        return "Propiedad inexistente"
    }
}

window.alert(verificarPropiedad(miPerro,1)); //Propiedad:
bonita,gordita,alegre
window.alert(verificarPropiedad(miPerro,"color")); // Propiedad inexistente
window.alert(verificarPropiedad(miPerro,"nombre")); // Propiedad: Luna
```

## 19) Objetos complejos

```
// Titulo: 20ObjetosComplejos.js

var ordenesDePizzas=[ //objeto tipo array por []
    //las llaves nos sirven para andiar atributos(objetos) y dentro de esos
    otros
    //atributos podemos crear otras {} para anidar mas atributos como si
    fuera un array
    { //primer objeto
        "tipo": "pastor",
        "tamaño":"individual",
        "precio":55.67,
        "ingredientes":["queso","pastor","piña"],
        "paraLlevar": true
    }//fin primer objeto
    , //coma para indicar cambio de objeto
    { //segundo objeto
        "tipo": "quesos",
        "tamaño":"mediana",
        "precio":125.35,
        "ingredientes":["queso1","queso2","queso3"],
        "paraLlevar": false
    }//fin segundo objeto
]

window.alert(ordenesDePizzas[0]); //llamamos obeejto 1
window.alert(ordenesDePizzas[1]); //llamamos obeejto

window,alert(ordenesDePizzas[0].tipo); //pastor
window,alert(ordenesDePizzas[1].tipo); //quesos

window,alert(ordenesDePizzas[0].ingredientes[2]); //piña
window,alert(ordenesDePizzas[1].ingredientes[0]); //queso1
```

## 20) Números aleatorios

```
//Titulo: 21NumerosAleatorios.js

//Num decimal-----

//Por medio de funcion
function numDecimalAleatorio(){
    return "Primer decimal"+Math.random();
}
window.alert(numDecimalAleatorio());

//Por medio de variable
var a = Math.random();
window.alert("Segundo decimal"+a);

//Numero entero entre 0 y 19-----

//Por medio de funcion
function numEnteroAleatorio(){
    return "Primer entero"+Math.floor(Math.random()*20);
    //floor solo toma el numero entero y deja a un lado el decimal
}
window.alert(numEnteroAleatorio());

//Por medio de variable
var b = Math.floor(Math.random()*20);
window.alert("Segundo entero: "+b);

//Ejercicio numero aleatorio dependiendo del limite que se defina en un
bucle for

function numAleatorio(limiteSuperior){
    return Math.floor(Math.random()*(limiteSuperior+1));
}
for(var i=0; i<=5;i++){
    window.alert(numAleatorio(10));
}

var a =1;
var b= 6;

function rangoAleatorio(limiteSuperior,limiteInferior){
```

```
        return Math.floor(Math.random()*(limiteSuperior-  
limiteInferior+1))+limiteInferior;  
    }  
    for(var i=0; i<=5;i++){  
        window.alert(rangoAleatorio(a,b));  
    }  
}
```

## 21) Función parseInt()

```
//Titulo: 22FuncionParseInt().js

//Ejemplo1

window.alert(parseInt("6")); //6
window.alert(parseInt("8.7")); //lo deja como 8
window.alert(parseInt(9.9)); //redondea a 9
window.alert(parseInt("AA")); //NAN = not a number= no es un numero

//Ejemplo2 operacion

var a = "5";
var b = "6";

window.alert(parseInt(a)+parseInt(b));

//Ejemplo sistema binario
window.alert(parseInt("110111",2));
//Ejemplo sistema hexadecimal
window.alert(parseInt("3E0A",16));
```

## 22) Operador ternario

```
//Titulo: 23OperadorTernario.js

//Ejemplo base

function retornarMinimo(a,b){
    return a<b ? a:b ;
    // condicion a<b si se cumple regresa a y si no b
}

window.alert(retornarMinimo(2,3)); //

//Operador ternario anidados

function compararNumeros(a,b){
    return a==b ? "a = b " : a<b ? "a < b": "a > b";
}

window.alert(compararNumeros(4,3)); //a > b
window.alert(compararNumeros(1,6)); // a < b
window.alert(compararNumeros(2,2)); // a == b
```

## 23) Variable var y let

```
//Titulo: 24variableVarYLet.js

//Diferencia entre var y let

var nombre = "luis";
var nombre = "paola";
//podemos definir muchas veces la misma variable y asignar valores

let apellido ="Hernandez";
//let apellido="Bravo"; //si dejamos el let nos marca error por que la
variable ya se declaro
apellido="Bravo";
// si quitamos let, ahora podemos modificar el valor de la variable

//Ambitos de var y let
```

```
//Ejemplo con var-----
var miVariableGlobal=5;
window.alert(miVariableGlobal);

function miFuncion(){
    window.alert(miVariableGlobal);

    var miVariableLocal=8;
    window.alert(miVariableLocal);
}

miFuncion();

//window.alert(miVariableLocal); //error de referencia porque esta en la
funcion
window.alert(miVariableGlobal);// con la variable global no hay problema

for (var i=0; i<3; i++){ //aquí i trabaja como una variable global
    window.alert(i);
}

window.alert("Variable i:"+i); // aquí podemos usar i por ser global


//Ejemplo con let-----
for (let j=0; j<3; j++){
    window.alert(j);
}

//window.alert("Variable j:"+j); //marca error porque let vuelve a declarar j
solo en el bucle for
```

## 24) Variable const

```
//Titulo: 25Variableconst.js

/*const hace referencia a constant o constante por lo que una vez se
inicializo
o se asigno un valor ya no se puede modificar*/
/*
const miConstante= 35;
window.alert(miConstante);
//const se debe crear e inicializar en una sola linea es decir
//no podemos hacer esto:
    //const miConstante2; // creamos variable
    //miConstante2=15;    //inicializamos, nos saldra error al hacerlo asi
miConstante=15; //error no podemos cambiar el valor
window.alert(miConstante);
*/

//CAMBIAR VALORES SOLO EN ARRAY
/**/
const MI_ARRAY=[1,2,3,4];
for(var i=0; i<MI_ARRAY.length; i++){
window.alert("Primer array: "+MI_ARRAY[i]);
}

MI_ARRAY[0]=5;
MI_ARRAY[1]=6;
MI_ARRAY[2]=7;
MI_ARRAY[3]=8;

for(var i=0; i<MI_ARRAY.length; i++){
    window.alert("Segundo array: "+MI_ARRAY[i]);
}
```



## 25) Objetos inmutables

```
//Titulo: 26ObjetoInmutable.js

let colores={
  "verde":"#10e04b",
  "azul": "#1b50e0",
  "negro": "#000000",
  "blanco": "#ffffff"
};

Object.freeze(colores);//con esto evitamos el cambio o adición de atributos

colores.amarillo ="#ffff200"; //error no se puede agregar
colores.verde ="#ffff200"; //error no se puede modificar
delete colores.verde; //error no se puede eliminar
```

## 26) Funciones flecha

```
//Titulo: 27FuncionesFlecha.js

//son funciones compactas
//Diferencia de función normal y flecha

//Así definimos de forma normal una función anónima después de function no
hay nombre
//pero si esta asignada a una variable
const fecha= function(){
  return new Date();
};
//con la función flecha
const fecha = () => new Date();

//Ejemplo1

//función normal anónima
const sumarTres = function(x){
  return x+3;
}

//función flecha
```

```
const sumarTres2 = (y) => x+3;

window.alert(sumarTres2(5)); //8

//Ejemplo con array

//funcion normal anonima
const concatenarArray = function(arr1,arr2){
    return arr1.concat( arr2); //concat nos sirve para concatenar o unir 2
array
};

var arr3 = concatenarArray([1,2],[3,4,5]);

for(var i=0; i <arr3.length; i++){
    window.alert("Funcion normal: "+arr3[i]);
}

//funcion flecha
const concatenarArray2 = (arr1,arr2) => arr1.concat( arr2);
var arr4 = concatenarArray2([2,3],[4,5,6]);

for(var i=0; i <arr4.length; i++){
    window.alert("Funcion flecha: "+arr4[i]);
}

//Ejemplo quitando palabra function y poniendo =>
//funcion normal
//const sumar = function(a,b){
//    let num =6;
//    return a+b+num;
//};

//funcion flecha
const sumar =(a,b) => {
    let num =6;
    return a+b+num;
};

window.alert(sumar(2,2)); //10
```

## 27) Valores por defecto en parámetros en funciones

```
//Titulo: 28ValoresPorDefectoEnParametrosEnFunciones.js

const incrementar = (num, valor=1) => { return num +valor };

//como el parametro valor se definio con uno el resultado es
window.alert(incrementar(5)); //6
window.alert(incrementar(5,3)); //8
//como mandamos el valor 3 se asigna a la variable valor aunque ya tenga 1
```

## 28) Operador rest

```
//Titulo: 29OperadorRest.js

function miFuncion(...args){ //operador rest ...args (areglo)
    window.alert(args);
    window.alert(args.length);
}

//miFuncion(1,2,3,4); //asi podemos agregar en parametros los valores que
necesitemos
miFuncion([1,2,3],[4,5,6]); //tambien en forma de array si es necesario
```

## 29) Operador spread

```
//Titulo: 30OperadorSpread.js

//es lo contrario a Rest este operador descompone los parametros

const numeros = [1,2,3];

function sumar (x,y,z){
    return x + y + z;
}

window.alert(sumar(...numeros)); //tomamos los valores del array y separamos
el resultado es 6
// ...nombreVariable
```

### 30) Destructuración

```
//Titulo: 31Destructuracion.js

//Funcionamiento

const usuairo ={
  nombre: "Luis",
  edad: 26
};

//si quisieramos asignar los valores de uno en uno lo hacemos como se
muestra a continuacion
//const nom = usuairo.nombre;
//const ed =usuairo.edad;
//observemos que podemos colocar el nombre que se nos antoje a la variable

const {nombre, edad} =usuairo;
//tipoVar{ var1,var2}=objeto;
//con esto creamos las variables y asignamos dichos valores pero las
variables deben
//llamarse igual que el atributo dentro del objeto como nombre y edad

//Ejemplo con coordenadas

var coordenadas = {
  x:5,
  y:6,
  z:8
};

var {x,y,z}=coordenadas;
window.alert("Cordenada x :"+x);
window.alert("Cordenada y :"+y);
window.alert("Cordenada z :"+z);

//Ejemplo1 con objetos anidados

const usuario={
  luis: {
    nombre: "luis",
    edad:27,
    correo: "luisgmail.com"
```

```
    },
    paola:{
      hobie:{
        op1: "jugar",
        op2: "leer"
      }
    }
  };

const {luis:{nombre,edad,correo}}=usuario;
//tipoVar {atributoAExtraer:{subatributo,subatributo}}=objeto;
window.alert(nombre+" edad: "+edad);
window.alert(nombre+" correo "+correo);

const {paola:{hobie:{op1}}}=usuario;
window.alert("hobie paola "+op1);

//Ejemplo2 con objetos anidados cambiando nombre de variable

const usuario={
  luis: {
    nombre: "luiss",
    edad:27,
    correo: "luisgmail.com"
  },
  paola:{
    nombre: "paola",
    hobie:{
      op1: "jugar",
      op2: "leer"
    }
  }
};

const
{luis:{nombre:nombreUsuario,edad:edadUsuario,correo:correoUsuario}}=usuario;
//tipoVar
{atributoAExtraer:{subatributo:nuevoNombreVar,subatributo:nuevoNombreVar}}=o
bjeto;
window.alert(nombreUsuario+" edad: "+edadUsuario);
window.alert(nombreUsuario+" correo "+correoUsuario);

const {paola:{nombre:nombreUsuario2,hobie:{op1:hobbie1}}}=usuario;
//Para facilitar despues del nombre del atributo:nuevoNombraVar
```

```
window.alert("hobie de "+nombreUsuario2+": "+hobbie1);

//Ejemplo con array

var a , b, c ; //declaramos variables
[a,b,c]=[1,2,3]; //podemos asignar valores de esta manera
window.alert(a+" "+b+" "+c); //podemos imprimir 1 2 3

//asignar variables a menos
var x, y ;
[x,y]=[4,5,6,7,8,9];

window.alert(x+" "+y); //x= 4 y =5

//asignar otro valor de otra posicion por ejemplo del arreglo
[6,7,8,9,10,11,12,13]
// asignar a l el valor 10 que es la posicion 4
var j,k,l;
[j,k,,l]=[6,7,8,9,10,11,12,13]//escribimos la l dejando el numero de
espacios es decir poniendo comas ,
//j=posicion 0, k= posicion 1, l= posicion 10
window.alert(j+" "+k+" "+l); //j=6, k=7 l=10

//Intercambiar valores en un array

var a =8, b= 6;
[b,a]=[a,b] //b=a por lo que b= 8, a=b por lo que a=6

window.alert(a+" "+b); //a=6, b=8

//Ejemplo con operador rest en array

var a, b;
// ... nombreVariableArray
[a,b,...array] = [1,2,3,4,5,6,7] //a=1, b=2, array= [3,4,5,6,7]
window.alert(a); //1
window.alert(b); //2
window.alert(array); //3,4,5,6,7

//remove primeros elementos
const arrayInicial =[1,2,3,4,5,6,7,8];

function removerTresPrimerosElementos(array2){
    const [,,,...arrayLocal]=array2;
```

```
        return arrayLocal;
    }
    window.alert("array inicial: "+arrayInicial)
    window.alert("Quitando los primeros 3: "+
    removerTresPrimerosElementos(arrayInicial));

//Ejemplo pasar objeto como argumento

//definimos objeto
var nuevoPerfilCliente={
    nombre: "Luis",
    edad: 26,
    origen: "Mexico",
    ubicacion: "Pachuca"
};
//creamos metodo o funcion flecha para destructural como lo hicimos hasta
ahora
const actualizarPerfil = (informacionDePerfil) =>{
    const{nombre,edad,origen,ubicacion}= informacionDePerfil;
    window.alert(nombre);
    window.alert(edad);
    window.alert(origen);
    window.alert(ubicacion);
};
//llamamos la funcion y mandamos nuestro objeto
actualizarPerfil(nuevoPerfilCliente);

//ahora lo haremos como destructuracion
//definimos objeto
var nuevoPerfilCliente2={
    nombre: "Paola",
    edad: 24,
    origen: "Mexico",
    ubicacion: "Pachuca"
}
//creamos metodo pero ahora nuestro argumento o variable son los atributos
del objeto en automatico
//js hace la asignacion
const actualizarPerfil2 = ({nombre,edad,origen,ubicacion})=>{
    window.alert(nombre);
    window.alert(edad);
    window.alert(origen);
    window.alert(ubicacion);
}
```

```
actualizarPerfil2(nuevoPerfilCliente2);

//Ejemplo numerico y funcion flecha

const estadictisas={
  media:10,
  mediana:5,
  minimo: 4,
  maximo: 10
}

const promedio = ({maximo,minimo})=> (maximo+minimo)/2;

window.alert(promedio(estadictisas)); // (10+4)/2=7
```

### 31) Objetos con forma concisa

```
//Titulo:33ObjetosConFormaConcisa.js

//Creamos una funcion flecha para crear un objeto
//despues de la flecha abrimos llaves {}
const crearUsusairo=(nombre,edad,lengua)=>{
return {
  nombre: nombre,
  edad: edad,
  lengua: lengua,
};
};

//window.alert(crearUsusairo("Luis",26,"Español"));
//al usar window.alert saldra el mensaje [object Object] dando a entender
que funciona
//esto se puede procesar mejor en la terminal y en lugar de window.alert()
usar console.log()
console.log(crearUsusairo("Luis",26,"Español"));

//Ahora creamos un objeto colocando parentesis antes y despues de los
corchetes
const crearUsusairo2=(nombre,edad,lengua)=>({nombre,edad,lengua});
//tenemos entonces que la funcion creara un objeto con los mismo nombres que
tenemos como argumentos
window.alert(crearUsusairo2("Paola",24,"Español"));
```



## 32) Plantillas de cadenas

```
//Titulo: 32PlantillasDeCadenas.js

/*A la hora de imprimir podemos cambiar las comillas "" para escribir y usar
acento invertido `
Para colocar texto (alt+tecla esta a un lado del botón borrar)
Podemos cambiar el concatenamiento por ${} dentro de las llaves podemos
escribir variables o
Podemos escribir valores diferentes*/

//Ejemplo variables

var nombre= "Luis";
var edad=26;
window.alert(`Nombre: ${nombre} y tengo ${edad}`);

//Ejemplo array
var array=[1,2,3,4];
window.alert(`El arreglo es ${array}`); //se imprime asi= 1,2,3,4
window.alert(`El arreglo es ${JSON.stringify(array)}`); //Se imprime con
corchetes= [1,2,3,4]

//Ejemplo con objeto
var persona ={
    nombre: "Paola",
    edad: 24
}
//declaramos variable con mensaje
const saludo= `Hola mi nombre es ${persona.nombre} y tengo ${persona.edad}`;
window.alert(saludo);
```

### 33) Metodos

```
//Titulo: 34Metodos.js

//Podemos decir que cuando un objeto tiene una funcion como propiedad es un
metodo
//Opcion 1

const persona={
  nombre: "Paola",
  presentarse: function(){ //dejamos function
    return `Hola mi nombre es ${this.nombre}`; //this para indicar que
usamos la variable del objeto
  }
};

//opcion 2
const persona={
  nombre: "Paola",
  presentarse(){ //quitamos function el resultado es el mismo
    return `Hola mi nombre es ${this.nombre}`; //this para indicar que
usamos la variable del objeto
  }
};

window.alert(persona.presentarse());

//a diferencia de java en javascript tenem
```

## 34) Clases

```
//Titulo: 35Clases.js

/*a diferencia de java no tenemos todas la capacidades que podemos tener en
una clase, en realida
lo que hacemos es crear un objeto*/
//Ejemplo 1

class TransboradorEspacial{
    constructor(planeta){ //funcion o metodo que se ejecuta siempre y al
iniciar
        this.planeta = planeta; //indicamos que el planeta de this clase se le
asignara el valor de
        // planeta enviado al constructor
    }
}

var zeus = new TransboradorEspacial("Jupiter"); //creamos mensajero similar
a java y mandamos var planeta
window.alert(zeus.planeta); //llamamos al mensajero y que nos de la variable

var prometeus = new TransboradorEspacial("Nova 4");
window.alert(prometeus.planeta);

//Ejemplo2
class Mascota {
    constructor (nombre, edad){
        this.nombre= nombre;
        this.edad=edad;
    }
}

var luna=new Mascota("Luna",8);
window.alert(luna.nombre + luna.edad);
window.alert(`Nombre: ${luna.nombre} edad: ${luna.edad}`);
window.alert("Nombre: "+luna.nombre+" edad: " +luna.edad);
```

### 35) Getter y Setter

```
//Titulo: 36GetterYSetter.js

class Libro{
    constructor(autor){ //argumento del constructor entre parentesis
        //this.autor = autor; //variable accesible para cualquiera y poder
        //modificar
        this._autor = autor; //con el _ ahora la variable es privada como en
        //java
        //en toeria sigue siendo accesible la variable pero el guion bajo
        //indica a otro programador que no
        //se debe modificar, es decir que es privada
        //entonces para poder interactuar con la variable al igual que en
        //java necesitamos getter y setter
    }
    //con getter accedemos al valor sin modificarlo es decir retorna para poder
    //usarlo
    get autor(){ //nombre del argumento del constructor
        return this._autor;
    }
    //con setter nos deja actualizar el valor
    set autor(nuevoAutor){ //nombre del argumento del constructor
        //nuevoArgumento
        window.alert("setter");//ponemos mensaje para saber que usamos el
        //setter
        this._autor=nuevoAutor; //podemos poner condicionales para asegurar
        //cierto valor
    }
}

const libro1= new Libro("anonimo"); //creamos mensajero mandamos el nombre
window.alert(libro1.autor); //llamamos al mensajero con la variables sin _
//es decir
//el argumento del constructor

//usando setter para actualizar valor
libro1.autor="Luis";
window.alert(libro1.autor);
```