

SINTAXIS JAVASCRIPT PRIMEROS CONCEPTOS

NOELIA CÁMARA 2024

Contenido

PRIMEROS CONCEPTOS.....	4
ESTRUCTURAS	4
Sentencia condicional if else	4
If -else-if.....	4
SENTENCIA CONDICIONAL SWITCH CASE.....	4
REPETICIONES.....	5
BUCLE FOR.....	5
BUCLE FOR ...IN	6
BUCLE FOR ...OF.....	6
BUCLE DO ...WHILE	6
OPERADORES.....	7
OPERADORES BINARIOS	7
OPERADORES ARITMÉTICOS	7
OPERADORES DE ASIGNACION.....	7
OPERADORES DE CONCATENACION.....	8
OPERADORES LÓGICOS	9
TERNARIO	9
DE TIPOS.....	9
INSTANCIAS	9
AMBITO VARIABLES.....	10
VARIABLES LOCALES Y GLOBALES CON EL MISMO NOMBRE.....	10
VARIABLES DECLARADAS DENTRO DE UN BUCLE	10
FUNCIONES.....	10
FUNCIONES NORMALES	10
FUNCIONES ANÓNIMAS	11
FUNCIONES ANÓNIMAS AUTOINVOCADAS	12
FUNCIONES FLECHA	12
OBJETOS PREDEFINIDOS	13
BOOLEAN.....	14
DATE	15
MOMENT (IMPORTANDO)	17
NUMBERS	20
MATH.....	22
STRINGS.....	26
EJERCICIOS STRINGS	29

OBJETOS NAVEGADOR	30
BOM WINDOW	30
GEOPOSICIONAMIENTO	37
OBJETOS	39
EJERCICIOS OBJETOS	42
CLASES	46
ARRAYS	48
EJERCICIOS ARRAYS	56

PRIMEROS CONCEPTOS

ESTRUCTURAS

Sentencia condicional if else

```
// if-else. Evalua la parte del if, si es verdadero ejecuta el codigo dentro del if y en caso contrario entra en el else
```

```
if (edad > 0) {  
    alert("Mayor");  
} else {  
    alert("Menor");  
}
```

If -else-if

```
let nota = 8;  
if (nota < 5) {  
    alert("suspense");  
} else if (nota >= 5 && nota < 6) {  
    alert("bien");  
} else if (nota >= 7 && nota < 9) {  
    alert("notable");  
} else if (nota <= 10) {  
    alert("Sobresaliente");  
} else {  
    ("No evaluable");  
}*  

```

SENTENCIA CONDICIONAL SWITCH CASE

```
//Permite realizar distintas acciones segun las condiciones  
//la condicion se evalua 1 vez  
// break: sale del broque swich  
// default : se ejecuta en caso de que la expresion no coincida con  
ningun valor  
// let nota = 2;  
// switch (nota) {  
//     case 0:  
//     case 1:  
//     case 2:  
//     case 3:  
//     case 4:  
//         alert("suspense");  
//         break;  
//     case 5:  
//     case 6:  
//         alert("bien");  
//         break;  
// }
```

```

//     case 7:
//     case 8:
//         alert("notable");
//         break;
//     case 9:
//     case 10:
//         alert("sobresaliente");
//         break;
//     default:
//         alert("no evaluable");
// }

```

REPETICIONES

BUCLE FOR

```

// for (sentencia1;sentencia2;sentencia3){
//     Código
// }

// for (i=0 ; i<10; i++){
//     alert(i);
// }
// - variacion 1: omitir la primera sentencia --> debemos
inicializar la variable con anterioridad.

// let i= 0;
// for(; i<10; i++){
//     alert(i);
// }

// - variacion 2: omitir la segunda sentencia

// for (i = 0; ;i++){
//     alert(i);
//     if (i==5){
//         break;
//     }
// }

// - variacion 3: omitir la tercera sentencia --> incrementar la
variable dentro del bucle

// for (i = 0; i<5 ;){
//     alert(i);
//     i++;

// }

```

```
// - variacion 4:trabajar con dos o más variables dentro del bucle.
```

```
// for(i = 0, t=5; i<5 && t<10;i++, t++){  
//     alert ("i : " + i + " t : " + t) ;  
// }
```

```
// for(i = 0, t=5; i<5 , t<10;i++, t++){  
//     alert ("i : " + i + " t : " + t) ;  
// }
```

BUCLE FOR ...IN

```
let coches = ["opel", "seat", "volvo"];  
// for (coche in coches){  
//     alert(coches[coche]);  
// }  
//con este tienes que utilizar el indice para que te de el valor
```

BUCLE FOR ...OF

Directamente te da el valor al contrario que el anterior no necesitas el indice

```
// for(let coche of coches){  
//     alert (coche);  
// }  
  
// for (prop of fruta){  
//     alert (prop);  
// }
```

BUCLE DO ...WHILE

```
// Obligatoriamente se ejecuta al menos 1 vez  
let variableWhile = 1;  
  
do {  
    alert(variableWhile);  
    variableWhile++;  
} while (variableWhile < 10);
```

OPERADORES

Simbolos y palabras que realizan operaciones sobre uno o varios valores.

- Formado por operando y operador
- Operadores unarios(i++) y binarios(a+b)
- Aritméticos: + , - , * , / , % , **

OPERADORES BINARIOS

```
let a = 5 + 2;
document.write(a + "<br>");

a = 5 - 2;
document.write(a + "<br>");

a = 5 * 2;
document.write(a + "<br>");

a = 5 / 2;
document.write(a + "<br>");

a = 5 % 2;
document.write(a + "<br>");

a = 5 ** 2;
document.write(a + "<br>");
```

OPERADORES ARITMÉTICOS

```
let b = 1;
b++;
document.write(b + "<br>");

let c = 1;
c--;
document.write(c + "<br>");
```

OPERADORES DE ASIGNACION

```
=, +=, -=, *=, /=, %=
let d = 5 ;
document.write("= " + d + " <br>");

d += 2;
document.write("+ = " + d + " equivale a d = d + 2<br>");

d -= 2;
document.write("- =, d = d - 2 " + d + " equivale a d = d - 2 <br>");

d *= 2;
document.write("* =, d = d * 2 " + d + " equivale a d = d * 2 <br>");
```

```

d /= 2;
document.write("/ =, d = d / 2 "+d + " equivale a d = d / 2 <br>");

d %= 2;
document.write("% = , d = d % 2"+d + " equivale a d = d % 2 <br>")

```

OPERADORES DE CONCATENACION

DE CADENA

```

let nombre = "Juan";
let apellido = "Gonzalez";
let nombreCompleto = nombre + " " + apellido ;
document.write(nombreCompleto+ "<br>");
let texto = "ABCD";
texto += "EFGHI"; //texto = texto + "EFGHI"
document.write(texto + "<br>");

```

DE COMPARACIÓN

```

- Igualdad == (compara SOLO el valor)
  - Estricta === (compara tipo y valor)
  - Desigualdad != (compara SOLO el valor)
  - Desigualdad estricta !== (compara tipo y valor)
  - Mayor: >
  - Menor: <
  - Mayor o igual: >=
  - Menor o igual: <=
document.write(" DE COMPARACION <br>");
document.write(2 == 1 ); //false
document.write("<br>");
document.write(2 == 2 ); //true
document.write("<br>");
document.write(2 == '2'); //true
document.write("<br>");
document.write(2 === 1); //false
document.write("<br>");
document.write(2 === '2'); //false
document.write("<br>");
document.write(2 === 2); //true
document.write("<br>");
document.write(2 != 1); //true
document.write("<br>");
document.write(2 != '2'); //false
document.write("<br>");
document.write(2 !== '1'); //true
document.write("<br>");
document.write(2 !== 1); //true
document.write("<br>");

```


OPERADORES LÓGICOS

```
- AND : &&
- OR : ||
- NOT : !

let x = 4, y = 2;
let resultado = (x < 5 && y < 3 ); //true
document.write(resultado);
document.write("<br>");
resultado = (x > 5 && y < 3 ); //false
document.write(resultado);
document.write("<br>");
```

TERNARIO

```
let edad =18;
resultado= (edad >= 18) ? "Se cumple": "No se cumple";
document.write(resultado);
document.write("<br>");
```

DE TIPOS

```
document.write("<br> DE TIPOS <br>");
edad =true;
resultado= (edad >= 18) ? "Se cumple": "No se cumple";
document.write(typeof edad);
document.write("<br>");
```

INSTANCIAS

```
document.write("<br> INSTANCIAS <br>");
let vehiculos = ["coche","moto","quad"];
document.write(vehiculos instanceof Array);//true
document.write("<br>");

document.write(vehiculos instanceof Object);//true
document.write("<br>");

document.write(vehiculos instanceof String);//false
document.write("<br>");

document.write(vehiculos instanceof Number);//false
document.write("<br>");

document.write(typeof vehiculos[0]);
document.write("<br>");
```

AMBITO VARIABLES

VARIABLES LOCALES Y GLOBALES CON EL MISMO NOMBRE

```
let miVariable = "Fuera";
function ambito() {
  let miVariable = "Dentro";
  alert(miVariable);
}

console.log(miVariable); //fuera
ambito(); //dentro
console.log(miVariable); //fuera

// si cambia el valor de miFuncion (sin utilizar let dentro de la
funcion)
let miVariable2 = "Fuera";
function ambito() {
  miVariable2 = "Dentro";
  alert(miVariable2);
}

alert(miVariable2); //fuera
ambito(); //dentro
alert(miVariable2); //dentro
```

VARIABLES DECLARADAS DENTRO DE UN BUCLE

```
let letras = ["a", "b", "c"];
for (let letra of letras) {
  alert(letra);
}
```

FUNCIONES

FUNCIONES NORMALES

```
/ Funciones "normales"
// Sintaxis

// function nombreFuncion(parametros_separados_por_comas){
//    Codigo
// };

// Declaracion de una funcion sin parametros y no devuelve ningun
valor
function saludar (){
  console.log("Hola");
}
```

```

};

//Llamada a la funcion 1
saludar();

// Llamada a la funcion 2
let resultadoSaludar = saludar();
console.log(resultadoSaludar);

// Llamada sin parentesis (hace referencia al a funcion)
let resultadoSaludar2=saludar;
console.log(resultadoSaludar2);

// Funcion con parametros
function producto (num1, num2){
    console.log(num1 * num2);
};

producto(2,5);
function producto2 (num1, num2){
    // let resultado = num1 * num2;
    // return resultado;
    return num1*num2;
};

```

FUNCIONES ANÓNIMAS

```

// FUNCIONES ANONIMAS

// function nombre(parametros){instrucciones};

// SINTAXIS DE UNA FUNCION ANÓNIMA

let suma = function (a,b) {return a+b;};
console.log(suma(2,5));

let resultado = suma(3,5);
console.log(resultado);

// CON CONSTRUCTOR FUNCION ()
let miFuncion = new Function("a","b","return a+b");
console.log(miFuncion(10,10));
let resultadoConstructor = miFuncion(10,5);
console.log(resultadoConstructor);

```

FUNCIONES ANÓNIMAS AUTOINVOCADAS

```
//FUNCIONES ANÓNIMAS AUTOINVOCADAS
(function (){
    console.log("Hola buenas");
})();

(function (a,b){
    console.log(a+b);
}(4,5));
```

FUNCIONES FLECHA

```
//FUNCIONES FLECHA
// SINTAXIS
// (parametros) => {instrucciones;};

let suma = (a,b) => {return a+b;};
console.log(suma(5,6));

let suma2 = (a,b) => a+b;
console.log(suma(5,8));

// si no tiene parametros, obligados a poner paréntesis
let sinParametros = () => {console.log("Hola")};
sinParametros();

// Con un parámetro
let conUnParametro = (x) => x * x;
console.log(conUnParametro (3));

let conUnParametroB = x => x * x;
console.log(conUnParametro (4));

// valores por defecto

let suma3 = (a,b=0) => {return a+b;};
console.log(suma3(4));

let suma4 = (a=2,b=0) => {return a+b;};
console.log(suma4());

// funciones por parametro

const avisaUsuario = (fun,x) =>{
    alert(fun(x));
}

const saludarUsuario = (nombre = 'Amigo') =>{
```

```
    return `Hola ${nombre}`;  
}  
  
avisaUsuario(saludarUsuario, 'Sergio');
```

OBJETOS PREDEFINIDOS

```
// UTILIZACION DE LOS OBJETOS PREDEFINIDOS DE JAVASCRIPT  
// Objetivos:  
// -Identificar y conocer los objetos que nos proporciona js  
// -Propiedades y métodos de los principales objetos  
// -Aprender a manipular un documento HTML desde Javascript  
// -Manipular y gestionar la creación y apariencia de las ventanas del  
// navegador y la comunicación entre ellas  
  
// Datos primitivos  
// -String  
// -Bool  
// -Numero  
// -Undefinido  
// -Null  
  
// String  
let cadena = "DAW";  
// Number  
let num = 7.5;  
// Boolean  
let bool = true;  
// Undefined  
let indefinida = undefined;  
// Null  
let nulo = null;  
  
// OBJETOS NATIVOS: no dependen del navegador  
// -String  
// -Number  
// -Boolean  
// -Date  
// -Math  
// -Array  
// -RegExp  
// -Funcion  
// -Object  
  
// OBJETOS DE ALTO NIVEL: dependen del navegador  
// -Window  
// -Document  
// -History
```

```

// -Location
// -Navigator
// DEFINICION DE OBJETO
let lenguajesProg = { lenguaje: JSON, anio: 2000 };

// -Propiedades: se define en modo nombre: valor Ej: lenguaje, anio
// -Metodos: acciones que se pueden ejecutar sobre un objeto

```

BOOLEAN

```

// Objetos nativos, no dependen del navegador
// String, number, boolean, date, array, math, function, object,
// regex
// Objetos de alto nivel, si dependen del navegador
// Window,-- estas cuelgan de window: document, history,
// location, navigator, screen

// BOOLEAN
// function Boolean ()
document.write(Boolean(10>9));
document.write("<br>");
document.write(10<9);
document.write("<br>");

// Valores son ciertos (true)

document.write("90 es "+Boolean(90));
document.write("<br>");
document.write("4.5 es "+Boolean(4.5));
document.write("<br>");
document.write("-10 es "+Boolean(-10));
document.write("<br>");
document.write("1 es "+Boolean(1));
document.write("<br>");
document.write("String es "+Boolean("String"));
document.write("<br>");
document.write("false es "+Boolean("false"));
document.write("<br>");
document.write("true "+Boolean(true));
document.write("<br>");
document.write("<br>");

// Valores son falsos (false)
document.write("0 es "+Boolean(0));
document.write("<br>");
document.write("NaN es "+Boolean(NaN));
document.write("<br>");
document.write("Null es "+Boolean(null));

```

```

document.write("<br>");
document.write("Undefined es "+Boolean(undefined));
document.write("<br>");
document.write("Cadena vacía es "+Boolean(""));
document.write("<br>");
document.write("false "+Boolean(false));
document.write("<br>");

// Ejemplos de usos
if(90){
    document.write("Es verdadero");
    document.write("<br>");
}
let p;
if (p){
    document.write("Es verdadero");
    document.write("<br>");
    // Aqui no entra por ser undefined
}
if (!p){
    document.write("Es verdadero");
    document.write("<br>");
    // Ahora si que entra
}

```

DATE

```

let fecha1 = new Date();
document.write("Fecha actual " + fecha1 + "<br>");

// Creacion de fechas mediante cadenas/string
let fecha2 = new Date(
    "Mon Oct 15 2024 12:30:37 GMT+0200 (hora de verano de Europa
central)"
); //Ignora el dia de la semana en caso de que este mal o lo que
hay en los paréntesis.
document.write("Fecha 2 " + fecha2 + "<br>");
let fecha3 = new Date("October 14, 2024 12:00:00");
document.write("Fecha 3 " + fecha3 + "<br>");
let fecha4 = new Date("October 14 2024");
document.write("Fecha 4 " + fecha4 + "<br>");
let fecha5 = new Date("October 14 ");
document.write("Fecha 5 " + fecha5 + "<br>");
let fecha6 = new Date("2024 ");
document.write("Fecha 6 " + fecha6 + "<br>");
let fecha7 = new Date("2024-11-25");
document.write("Fecha 7 " + fecha7 + "<br>");

```

```

    // Con formato numerico , tanto los meses como los dias de la
    semana comienzan por 0
    // Meses (0-11) 0= Enero, 11=Diciembre
    // let f1 = new Date (año,mes,dia,horas,minutos,segundos y
    milisegundos)
    // Dias de la semana (0-6) 0= domingo y 6= sabado
    let f1 = new Date(2024, 9, 15);
    document.write("F1 " + f1 + "<br>");
    let f2 = new Date(2024, 11, 15, 10, 1, 40, 50);
    document.write("F1 " + f2 + "<br>");

    // Creacion de fechas con milisegundos
    let f3 = new Date(1524875247);
    document.write("F1 " + f3 + "<br>");

    // Formatos de visualizar la fecha
    let fechaHora = new Date();
    document.write("Fecha " + fechaHora + "<br>");
    document.write("toString " + fechaHora.toString() + "<br>");
    document.write("toUTCString " + fechaHora.toUTCString() + "<br>");
    document.write("toDateString " + fechaHora.toDateString() +
    "<br>");

    // METODOS GET

    document.write("getDay " + fechaHora.getDay() + "<br>"); //Dia de
    la semana del 0-6
    document.write("getDate " + fechaHora.getDate() + "<br>"); //Dia
    del mes
    document.write("getMonth " + fechaHora.getMonth() + "<br>"); //Mes
    del 0-11
    document.write("getFullYear " + fechaHora.getFullYear() + "<br>");
    //Año
    document.write("getYear " + fechaHora.getMinutes() + "<br>");
    //Minutos
    document.write("getSeconds " + fechaHora.getSeconds() + "<br>");
    //Segundos

    // METODOS SET

    fechaHora.setDate(10);
    fechaHora.setMonth(8);
    fechaHora.setFullYear(2023);
    fechaHora.setMinutes(26);
    fechaHora.setSeconds(15);
    fechaHora.setMilliseconds(40);

    document.write("Fecha "+fechaHora.toString()+"<br>");

```



```
// Suma de fechas
```

```
fechaHora.setMonth(fechaHora.getMonth()+2);  
document.write("Fecha "+fechaHora.toString()+"<br>");
```

MOMENT (IMPORTANDO)

```
// document.write(moment().format('MMMM Do yyyy,h:mm:ss a '));  
  
// Creacion de fecha  
let fechaActual = moment();  
document.write(fechaActual);  
document.write("<br>");  
document.write(fechaActual.format());  
document.write("<br>");  
document.write(fechaActual.format("DD/MM/YY"));  
document.write("<br>");  
document.write(fechaActual.format("LLL"));  
document.write("<br>");  
document.write(fechaActual.format("LL"));  
document.write("<br>");  
let fecha2 = moment("15-12-2024", "DD-MM-YYYY");  
document.write(fecha2);  
document.write("<br>");  
document.write(fecha2.format("YY/MM/DD"));  
document.write("<br>");  
// Suma y resta de fechas  
// sumar  
fechaActual = moment().add(7, 'Y').add(3, 'd').format("DD-MM-YYYY");  
document.write(fechaActual);  
document.write("<br>");  
// restar  
fechaActual = moment().subtract(7, 'Y').format("DD-MM-YYYY");  
document.write(fechaActual);  
// diferencia entre dos fechas  
let fechaInicio=moment("2024-01-01", "YYYY-MM-DD");  
let fechaFin= moment();  
let diferenciaFecha= fechaFin.diff(fechaInicio, 'months');  
document.write("<br>");  
document.write(diferenciaFecha);  
document.write("<br>");  
// Inicio y fin de periodos  
document.write("<br>");  
let inicioDia=moment().startOf('day').format('DD/MM/YYYY  
HH:mm:ss');  
document.write(inicioDia);  
document.write("<br>");
```

```

        let finoDia=moment().endOf('months').format('DD/MM/YYYY
HH:mm:ss');
        document.write(finoDia);
        document.write("<br>");
        // Para validar fechas

        let fechaValidar=moment('2022-02-30','YYYY-MM-DD').isValid();
        document.write(fechaValidar);
        document.write("<br>");

        // Comparar fechas
        let fecha10= moment('2024-10-15');
        let fecha11= moment();
        document.write("<br>");
        document.write(fecha10.isBefore(fecha11));
        document.write("<br>");
        document.write(fecha10.isSame(fecha11));
        document.write("<br>");
        document.write(fecha10.isAfter(fecha11));

        // Conversion de duracion

        let duracion=moment.duration(2,'months');
        document.write("<br>");
        document.write(duracion.asHours());

```

EJERCICIOS MOMENT

```

<!-- // Crear fecha actual y foramto personalizado, el que se quiera que
no sea día mes y año solo
        // Sumar o restar a un periodo una fecha
        // Calcular la dif de los días que han pasado desde que empezamos
el curso (10/09) hasta el día de hoy
        // Verificar alguna fecha si es anterior
        // Formatear una fecha a texto legible, pe 10/10/2024 a es día 10
de octubre de 2024
        // Encontrar el principio y fin de la semana de la fecha que
queramos
        // Calcular nuestra edad a partir de nuestra fecha de nacimiento --
>

        <!-- Pagina donde encontrar documentacion o poder importar codigo -
- www.momentjs.com -->
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>
        <title>Ejercicios moment</title>

```

```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.30.1/moment.min.js"></script>
</head>
<body>
  <script>
    // Crear fecha actual y formato personalizado, el que se quiera que
    no sea día mes y año solo

    let fechaActual = moment().format("DD-MM-YYYY HH:mm");
    document.write(fechaActual);
    document.write("<br>");
    // Sumar o restar a un periodo una fecha

    let fechaSuma = moment().add(8, "Y");
    document.write(fechaSuma);
    document.write("<br>");

    // Calcular la dif de los días que han pasado desde que empezamos
    el curso (10/09) hasta el día de hoy
    let fechaInicio = moment("2024-09-10", "YYYY-MM-DD");
    let fechaActual2 = moment();
    let diferenciaFecha = fechaActual2.diff(fechaInicio, "days");
    document.write("<br>");

    // Verificar alguna fecha si es anterior
    let primeraFecha = moment("2022-02-30");
    let segundaFecha = moment("2024-04-01");
    document.write(primeraFecha.isBefore(segundaFecha));
    document.write("<br>");

    // Formatear una fecha a texto legible, pe 10/10/2024 a es día 10
    de octubre de 2024

    let fecha = moment("2022-02-09", "YYYY-MM-DD");
    document.write(
      " Es día " +
      fecha.get("date") +
      " del " +
      fecha.get("month") +
      " de " +
      fecha.get("year")
    );
    document.write("<br>");

    // Encontrar el principio y fin de la semana de la fecha que
    queramos

    let fecha4 = moment("2024-10-21");

```

```

document.write(fecha4.startOf('week'));
document.write("<br>");
document.write(fecha4.endOf('week'));
document.write("<br>");
// con el start of

// Calcular nuestra edad a partir de nuestra fecha de nacimiento

let fechaNacimiento = moment("1995-02-03");
let fechaActual3 = moment();
document.write(
    "Tienes " + fechaActual3.diff(fechaNacimiento, "years") + "
años."
);
document.write("<br>");
// Si una fecha es válida
let fechaValidar1 = moment("2024-11-21", "YYYY-MM-DD").isValid();
document.write(fechaValidar1);
document.write("<br>");

let fechaValidar2 = moment("2024-11-32", "YYYY-MM-DD").isValid();
document.write(fechaValidar2);
document.write("<br>");

```

NUMBERS

```

/CARACTERISTICAS
// -Solo hay un tipo de datos de tipo numerico
// -Representacion. Todas son tipo number
let numEntero = 5;
let numDecimal = 5.5;
let numExp = 5e8;
let numExpNegativo = 5e-8;
// document.write(numExpPos)
let numExpPost = 5e8;
// document.write(numExpNeg)
let hexadecimal = 0xff;
//document.write(hexadecimal);

// Cambio de base entre numeros
// -toString()
let baseDecimal = 128;
document.write("Binario " + baseDecimal.toString(2) + "<br>");
document.write("Octal " + baseDecimal.toString(8) + "<br>");
document.write("Hexadecimal " + baseDecimal.toString(16) + "<br>");

// REPRESENTACION DE INFINITO Y -INFINITO
let inf = 2;
while (inf !== Infinity) {

```

```

    inf *= inf;
    document.write(inf + "<br>");
}
let div1 = 2,
    div2 = 0;
document.write("Division 2/0 : " + div1 / div2 + "<br>");

// NOT A NUMBER: NaN
let notANumber = 100 / "hola";
document.write(notANumber + "<br>");
let notANumber2 = 100 / "10";
document.write(notANumber2 + "<br>") + "<br>";
document.write(isNaN(notANumber2) + "<br>");
document.write(isNaN(notANumber) + "<br>");
document.write("Es de tipo " + typeof notANumber);

// NUMERO COMO OBJETO
let num1 = 456;
let num2 = new Number(456);
let num3 = new Number(456);
document.write("Num1 == Num2 "+(num1==num2)+"<br>");
document.write("Num1 === Num2 "+(num1===num2)+"<br>");
document.write("Num2 === Num3 "+(num1===num2)+"<br>");

// PROPIEDADES

let maximo=Number.MAX_VALUE;
document.write("Maximo "+maximo+"<br>");
let minimo=Number.MIN_VALUE;
document.write("Mínimo "+minimo+"<br>");
let neginf=Number.NEGATIVE_INFINITY;
document.write("Neginf "+neginf+"<br>");
let postinf=Number.POSITIVE_INFINITY;
document.write("Postinf "+postinf+"<br>");
let numNaN=Number.NaN;
document.write("NaN "+numNaN+"<br>");

// METODOS
// - toFixed(num decimales): cadena con el numero específico de
decimales
let decimales=7.8956;
document.write("ToFixed "+decimales.toFixed(2)+"<br>");
document.write("ToFixed "+decimales.toFixed(0)+"<br>");
document.write("ToFixed "+decimales.toFixed(6)+"<br>");
// - toPrecision(num cifras):
document.write("ToPrecision "+decimales.toPrecision(2)+"<br>");
document.write("ToPrecision "+decimales.toPrecision(1)+"<br>");
document.write("ToPrecision "+decimales.toPrecision(6)+"<br>");

```

```

// - toExponential():
let exp=1234567
document.write("ToExponential "+exp.toExponential(6)+"<br>");
document.write("ToExponential "+exp.toExponential(10)+"<br>");
document.write("ToExponential "+exp.toExponential(1)+"<br>");

// parseInt() / parseFloat()
document.write("ParseInt: "+ parseInt("10.6")+"<br>");
document.write("ParseInt: "+ parseInt("10 casa")+"<br>");
document.write("ParseInt: "+ parseInt("10 20")+"<br>");

// METODOS GLOBALES PARA CONVERTIR VARIABLES EN NUMEROS

document.write("Number (true): "+Number (true)+ "<br>");
document.write("Number(false): "+ Number (false)+ "<br>");
document.write("Number (new Date ()): "+ Number (new Date())+"<br>");
// document.write("Number ('casa+)' "+ parseInt("10 20")+"<br>");

```

MATH

```

// MATH
// Permite realizar operaciones matemáticas
// No es un constructor
// Tanto sus propiedades como los métodos deben ser llamados usando
directamente el nombre del objeto Math.

// PROPIEDADES
let euler = Math.E;
document.write("Número Euler " + euler + "<br>");
let logaritmo2E = Math.LOG2E;
document.write("Log2 " + euler + "<br>");
let logaritmo10E = Math.LOG10E;
document.write("Log10 " + euler + "<br>");
let pi = Math.pi;
document.write("Número PI " + euler + "<br>");
let raizCuadrada = Math.SQRT2;
document.write("Raíz cuadrada " + euler + "<br>");

// MÉTODOS
// Nos devuelve un numero aleatorio entre 0 y 1
let aleatorio = Math.random();
document.write("Número aleatorio " + aleatorio + "<br>");
// Raiz del número
let raizNum = Math.sqrt(36);
document.write("Raíz cuadrada " + raizNum + "<br>");

// max(lista de valores)/min(lista valores)
let minimo = Math.min(1, 3, 6, 4, 8, 9);
let maximo = Math.max(1, 3, 6, 4, 8, 9);

```

```
document.write("Mínimo " + minimo + "<br>");
document.write("Máximo " + maximo + "<br>");

// en el caso de arrays
let numeros = [1, 3, 6, 4, 8, 9];
// con (...numeros) se lista el array dentro del metodo Math.min o
max
let minimoArray = Math.min(...numeros);
let maximoArray = Math.max(...numeros);
document.write("Mínimo " + minimoArray + "<br>");
document.write("Máximo " + maximoArray + "<br>");

let listaNum = [4, 8, 9];
let listaNum2 = [1, 3, 5];
let resultado = listaNum.concat(listaNum2);
let resultado2 = [...listaNum, ...listaNum2];
document.write("Concat " + resultado + "<br>");
document.write("Concat2 " + resultado2 + "<br>");

// round redondea tanto al alza como a la baja
let redondeo = Math.round(7.499);
document.write("Redondea " + redondeo + "<br>");
// ceil redondea siempre al alza
let alta = Math.ceil(7.1);
document.write("Redondea a la alza " + alta + "<br>");
let alta1 = Math.ceil(7.8);
document.write("Redondea alza " + alta1 + "<br>");

// floor redondea siempre a la baja
let baja = Math.floor(7.8);
document.write("Redondea baja" + baja + "<br>");
let baja1 = Math.floor(7.1);
document.write("Redondea baja" + baja + "<br>");

// trunc devuelve la parte entera
let truncar = Math.trunc(7.87777777);
document.write("Truncar " + truncar + "<br>");
let trunc1 = Math.floor(7.19999999);
document.write("Truncar " + baja + "<br>");

// pow potencias
let potencia = Math.pow(3, 2);
document.write("Potencia de 3^2 = " + potencia + "<br>");
let potencia1 = Math.pow(5, 10);
document.write("Potencia de 5^10 = " + potencia1 + "<br>");

// abs(número) valores absolutos
let absoluto=Math.abs(-8);
document.write("Absoluto = " + absoluto + "<br>");
```

```
// Métodos trigonométricos
// acos(x)
// asin(x)
// atan2(x,y)
// sin(x)
// cos(x)
// tan(x)
```

EJERCICIOS NUMBERS

```
// Redondear un número a su valor entero más cercano (q tenga un valor decimal y lo redondee al entero más proximo).
function redondeo (num){
  let result=Math.round(num);
  return result;
}
console.log(redondeo(4.8));
// Redondear hacia arriba o hacia abajo pasar en la funcion el numero y elegir para donde quieres que redondee
function redondeo2 (num,select){
  if (select=="arriba"){
    let result=Math.ceil(num);

  }
  if (select=="abajo") {
    let result=Math.floor(num);
  }
  return result;
}
console.log(redondeo2(4.8,"arriba"));
console.log(redondeo2(4.8,"abajo"));

// Obtener el valor absoluto de un numero

function absoluto(num){
  let result=Math.abs(num);
  return result;
}
console.log(absoluto(-50));
// Numero aleatorio entre 0 y 50

function aleatorio (num){
  let result = Math.random()*num;
  return result;
}
// Convertir un numero decimal a entero
function conversion (num){
  let result=Math.trunc(num);
  return result;
}
```



```

}

console.log(conversion(4.8910));

// Obtener el numero más pequeño o más grande de un array

function pequeGrande (array,select){
  if (select == "grande"){
    result=Math.max(...array);
  }
  if (select == "pequeño"){
    result=Math.min(...array);
  }
  return result;
}

console.log(pequeGrande([1,8,9,5,10],"grande"));
console.log(pequeGrande([1,8,9,5,10],"pequeño"));
// Redondear a un número específico de decimales( entrada, un numero y
otro parametro diciendo a que decimal redondear)
function redondeo3 (num,decimal){
  let result=num.toFixed(decimal);
  return result;
}

console.log(redondeo3(4.597,2));

// Comprobar si un número es finito o NaN
function finitoNaN (num){
  if (isFinite(num)){
    console.log(num +" es un número finito");
  }
  if (isNaN(num)){
    console.log(num +" es NaN");
  }
}

finitoNaN('hola');
finitoNaN(8);
// Convertir una cadena a número
function numero (cadena){
  result = parseInt(cadena);
  return result;
}

let entero = numero ('5978');
console.log(result);

// Redondear un numero grande usando la notacion exponencial
function expo (num){
  result=num.toExponential(2);

```

```
    return result;
}

let exponencial = expo(8561789621456);
console.log(exponencial);
```

STRINGS

```
// Strings
let double = "comillas dobles";
let single = "comillas simples";
let backticks = `backticks`;
let str = 'comillas "dobles"';
let str2 = "comillas 'dobles'";
let str3 = `comillas "dobles"`;

console.log(double);
console.log(str);
console.log(str2);
console.log(str3);

let str4 = "comillas" + " dobles";
let str5 = `Esto es un String ${double} ${single}`;
let str6 = `Esto es un
String ${double} ${single}`;

console.log(str4);
console.log(str5);
console.log(str6);

// PROPIEDADES
// -length
console.log(str.length);

// MÉTODOS
// -Accediendo a las posiciones
let prueba = `Hola`;
// -Primeras posiciones
console.log(prueba[0]);
console.log(prueba.charAt(0));
console.log(prueba.at(0));

// Últimas posiciones

console.log(prueba[-1]);
console.log(prueba.charAt(-1));
console.log(prueba.at(-1));
console.log(prueba[prueba.length - 1]);
```

```
// ITERANDO POR LOS ELEMENTOS DEL STRING
for (let char of "hola") {
  console.log(char);
}

// INMUTABILIDAD
console.log(prueba[0]);
prueba[0] = "h";
console.log(prueba[0]);

prueba = "h" + prueba[1] + prueba[2] + prueba[3];
console.log(prueba);
console.log(prueba[0]);

// CAPITALIZACION
console.log(prueba.toUpperCase());
console.log(prueba.toLowerCase());

// BUSCANDO UNA SUBCADENA DE CARACTERES
let prueba2 = "Mañana tenemos examen tenemos";

// -indexOf (substring, [pos])
console.log(prueba2.indexOf("tenemos"));
console.log(prueba2.indexOf("en", 10));

// -lastIndexOf()
console.log(prueba2.lastIndexOf("tenemos"));
console.log(prueba2.lastIndexOf("en"));

// -search(cadena/expresion)
console.log(prueba2.search("Mañana"));

// -match(expresiones regulares)

// includes (substring, [pos])
console.log(prueba2.includes("nemos"));
console.log(prueba2.includes("tnemos"));

// startsWith() y endsWith()

console.log(prueba2.startsWith("Mañana"));
console.log(prueba2.startsWith("tenemos"));

// OBTENIENDO UN SUBSTRING
// -slice(posicion inicial, posicion final) //Solo el primero coge desde
ese hasta el final, si no hay que poner tambien la posicion final
console.log(prueba2.slice(4));
console.log(prueba2.slice(0, 8));
console.log(prueba2.slice(-1));
```

```

console.log(prueba2.slice(4,8));
console.log(prueba2.slice(0));

// -substring(posicion inicial, posicion final) //no admite negativos
console.log(prueba2.substring(4));
console.log(prueba2.substring(-1));
console.log(prueba2.substring(-4, -1));

// -substr(posicion inicial, largo), desde donde y que cantidad de
caracteres

console.log(prueba2.substr(1,5));

// - split recibe un caracter , [num veces]

console.log(prueba2.split(" "));
console.log(prueba2.split(" ",1)); //hará la separacion desde el primero
console.log(prueba2.split(" ",2)); //cogerá los dos primeros
console.log(prueba2.split(" ",6));// aunque pongas un índice mayor a lo
que hay te da el máximo

// -trim(cadena), te elimina los espacios en blanco que hay al principio
y al final

console.log("  Hola buenos dias  ".trim()); //te genera la cadena sin
espacios
console.log("  Hola buenos dias  ".trimStart());// quita desde el ppo
pero los del final te los deja
console.log("  Hola buenos dias  ".trimEnd());// quita los del final
pero los del principio los deja

// CONCATENACION

console.log(prueba2.concat(",y hoy tambien"));

// REPETICION // te repite la variable el numero de veces que le digas

let variable = "*";
console.log(variable.repeat(10));

// COMPARACION

// localCompare(cadena) : -1 (antes), 0 (igual), 1 (después). Compara
según el código ASCII
let frase1 = "Buenas";
let frase2 = "Adios";
let frase3= "Buenas";
let frase4= "buenas";

```

```

console.log(frase1.localeCompare(frase2));
console.log(frase2.localeCompare(frase1));
console.log(frase1.localeCompare(frase3));
console.log(frase1.localeCompare(frase4)); // en el ascii las minisculas
van antes

// - codePointAt() Para saber la posición ASCII

console.log("B".codePointAt());
console.log("b".codePointAt());
console.log("z".codePointAt().toString(2)); // dice la posicion en
binario
console.log("z".codePointAt().toString(16)); //dice la posicion en
hexadecimal
console.log("z".codePointAt().toString(10)); //dice la posicion en
decimal

console.log(String.fromCharCode(65)); //te cambia de codigo ascii al
caracter que represente ese numero ascii
console.log(String.fromCharCode(0x7a));

```

EJERCICIOS STRINGS

```

// Escribe una funcion que devuelva el string con el primer caracter en
mayuscula

function mayus (cadena){
  let primera=cadena[0];
  let mayuscula=primera.toUpperCase();

  let resto= cadena.slice(1);
  let resultado1= `${mayus} ${resto} `;

  return resultado1;
}

// Escribe una funcion que devuelva true si contiene una cadena de texto
determinada o sino false ( 2 parametros que cadena es y que quieres
buscar)
function encontrar(cadena,buscar){
  resultado2= cadena.includes(buscar);
  return resultado2;
}

console.log(encontrar("que tal","al"));
console.log(encontrar("que tal","ol"));

```

```
// Otra función que extraiga el valor numerico de un string y lo devuelva;

function valorNumerico (cadena){
    primero = cadena [0];
    resto=cadena.slice(1);
    return resto;
}
console.log(valorNumerico("@120"));

// Crea una función que reciba una cadena y un caracter y devuelva el indice de la primera aparicion de ese caracter

function indice (cadena,caracter){
    let result = cadena.indexOf(caracter);
    return result;
}
let solucion = indice("hola",'a');
console.log(solucion);
```

OBJETOS NAVEGADOR

BOM WINDOW

```
// CARACTERISTICAS
// - Representa una ventana abierta del navegador
// - No todos los métodos son compatibles con todos los navegadores
// - Si una ventana tiene etiquetas de tipo <iframe>, el navegador crea un objeto window para cada una de ellas

// PROPIEDADES DE WINDOW

let mensaje = "";
// name
window.name = "ventana1";
mensaje += "<br> Nombre: " + window.name;

// Tamaño de la ventana completa con scrollbar y toolbar
mensaje += "<br> Ancho externo: " + window.outerWidth;
mensaje += "<br> Alto externo: " + window.outerHeight;

// Tamaño de la ventana sin scrollbar y toolbar (para saber que tamaño de documento tenemos disponible. si doy zoom cambia el espacio disponible)
mensaje += "<br> Ancho interno: " + window.innerWidth;
mensaje += "<br> Alto interno: " + window.innerHeight;

// Scroll horizontal y vertical
// window.pageXOffset devuelve la posición inicial del scroll
```

```

mensaje += "<br> Scroll horizontal: " + window.pageXOffset;
mensaje += "<br> Scroll vertical: " + window.pageYOffset;

//Distancia desde la esquina superior izquierda
mensaje += "<br> Distancia desde la izquierda: " + window.screenX;
mensaje += "<br> Distancia desde arriba: " + window.screenY +
"<br><br>";

// Propiedades de iframes
// - frames
// - frameElement
// - length

// Propiedades con otras ventanas
// - closed
// - opener
// - parent, te devuelve quien es el padre de la ventana en la que
estemos en ese momento
// - self, te devuelve la ventana en la que tu estas en ese momento
document.write(mensaje);

// METODOS DE WINDOW MÁS IMPORTANTES
// open(URL, name, specs): crear ventanas
// ventana es una variable GLOBAL
let ventana = "";
function crearVentana() {
    // la variable ventana coge la referencia para luego acceder a
cerrarla
    // ventana= window.open("https://www.wikipedia.es","", "");
    ventana = window.open(
        "../ObjetosPredefinidos/Date/moment.js",
        "",
        "height=300,width=200"
    );
}
function cerrarVentana() {
    ventana.close();
}
function redimensionarVentana() {
    ventana.resizeBy(10, 10);
}
// Redimensionar nuestra ventana un numero de pixeles respecto a su
tamaño actual
// resizeBy(ancho,alto)
function redimensionarVentana() {
    ventana.resizeBy(10, 10);
}
// Redimensionar nuestra ventana a un numero de pixeles
// resizeTo(ancho,alto)

```

```

function resizeToVentana() {
    ventana.resizeTo(600, 400);
}
// Mueve nuestra ventana un numero de pixeles respecto a su tamaño
actual
// moveBy(ancho,alto)
function moveByVentana() {
    ventana.moveBy(10, 10);
}
// Mueve nuestra ventana a un numero de pixeles
// resizeTo(ancho,alto)
function moveToVentana() {
    ventana.moveTo(600, 400);
}

// scrollBy(ancho,largo): mueve las barras de scroll un numero de
pixeles desde la posicion actual
// scrollBy(ancho,alto)
function scrollByVentana() {
    ventana.scrollBy(10, 10);
}
// scrollTo(ancho,largo): mueve las barras de scroll a una posicion
function scrollToVentana() {
    ventana.scrollTo(600, 400);
}
// con focus para poner el foco en la ventana
function focusVentana() {
    ventana.focus();
}
// con blur para quitar el foto de la ventana
function blurVentana() {
    ventana.blur();
}
// con print te imprime donde tienes el foco
function printFoco() {
    print();
}
// Esta te imprime la ventana aunque no tengas el foco
function printVentana() {
    ventana.print();
}

// EVENTOS DE TIEMPO

// setTimeout(funcion,milisegundos)

// Primera forma
let temporizador = setTimeout(saludo, 5000);

```



```

function saludo() {
    console.log("Hola");
}

// Segunda forma, con una funcion flecha(arrow)

let temporizador2 = setTimeout(() => {
    console.log("adios");
}, 10000);

// setInterval(funcion,milisegundos)
// Primera forma
let interval1 = setInterval(saludo, 2000);

// Segunda forma
let interval2 = setInterval(() => {
    console.log("adios");
}, 2000);

let intervalo=setInterval(()=>{juego(jugador1, jugador2)}, 3000);

// clearTimeout(variableSetTimeout)
// usada en los botones
// clearInterval(variableSetInterval)
// usada en los botones

// OBJETO NAVIGATOR

// Características
// - Nos proporciona informacion sobre el navegador donde se está
ejecutando el código
// PROPIEDADES DEL OBJETO NAVIGATOR
// appCodeName, devuelve el codeName del navegador
let textoNavegador = "OBJETO NAVIGATOR";
// codeName del Navegador
textoNavegador += "<br> CodeName " + navigator.appCodeName;
// nombre del navegador
textoNavegador += "<br> AppName " + navigator.appName;
// version del navegador
textoNavegador += "<br> AppVersion " + navigator.appVersion;
// motor del navegador
textoNavegador += "<br> Motor del navegador " + navigator.product;
// plataforma del navegador
textoNavegador += "<br> Plataforma del navegador " +
navigator.platform;
// onLine
textoNavegador += "<br> Conectado a internet " + navigator.onLine;
// lenguaje del navegador

```

```

    textoNavegador += "<br> Lenguaje del navegador " +
navigator.language;
    // tiene las cookies activadas
    textoNavegador += "<br> Cookies activas " +
navigator.cookieEnabled;
    // cabecera user-agent devuelta por el navegador al server
    textoNavegador += "<br> User agent " + navigator.userAgent;
    // geolocalización
    textoNavegador += "<br> Geolocalización " + navigator.geolocation;

    // console.log(navigator.geolocation);

    // METODO NAVIGATOR
    textoNavegador += "<br> JavaEnabled? " + navigator.javaEnabled();

    document.write(textoNavegador);

    // FORMA 1 DE GEOPOSICIONAMIENTO
    //
https://developer.mozilla.org/es/docs/Web/API/Geolocation/getCurrentPosition

    // const options = {
    //   enableHighAccuracy: true,
    //   timeout: 5000,
    //   maximumAge: 0,
    // };

    // function success(pos) {
    //   const crd = pos.coords;

    //   console.log("Tu ubicación actual es:");
    //   console.log(`Latitud : ${crd.latitude}`);
    //   console.log(`Longitud: ${crd.longitude}`);
    //   console.log(`Más o menos ${crd.accuracy} metros.`);
    // }

    // function error(err) {
    //   console.warn(`ERROR(${err.code}): ${err.message}`);
    // }
    // // Si el usuario ha dado a que siempre permitir el uso de
    // ubicación siempre se ejecuta el success, Si no va a llamar a la función
    // error
    // navigator.geolocation.getCurrentPosition(success, error,
    options);

    // SCREEN
    // CARACTERÍSTICAS

```

```

// Contiene la informacion de la ventana
// PROPIEDADES DE SCREEN
let screenTexto="";
// Tamaño de pantalla
screenTexto += "<br> Ancho: " + screen.width;
screenTexto += "<br> Alto: " + screen.height+"<br>";

//Tamaño de pantalla sin barras de tareas

screenTexto += "<br> Ancho sin barra: " + screen.availWidth;
screenTexto += "<br> Alto sin barra: " + screen.availHeight+"<br>";

// Profundidad de color de pantalla
screenTexto += "<br> Profundidad de color de pantalla: " +
screen.colorDepth+"<br>";
// Resolucion de color en bits por pixel
screenTexto += "<br> Resolucion color : " + screen.pixelDepth
+"<br>";
document.write(screenTexto);

// HISTORY
// CARACTERISTICAS
// Guarda informacion de las URL dentro de una ventana
// PROPIEDADES
// length
let historyTexto="<br> HISTORY";
historyTexto += "<br> length : " + history.length;

// METODOS
// back
// carga la URL anterior
function atras(){
    history.back();
}
// forward
//carga la URL siguiente
function adelante(){
    history.forward();
}

// go
// se le puede pasar la URL o el numero, va a una pagina en concreto
del historial, en la que estamos es la 0, se pueden poner numeros
negativos para ir hacia atras o positivos hacia delante
function ir(){
    var numero= prompt("Indica un numero para moverte en el
historial")
    history.go(numero);

```

```

    }
    document.write(historyText);
    // LOCATION
    // CARACTERÍSTICAS
    // Guarda informacion de la URL actual
    // PROPIEDADES
    let locationText="<br> LOCATION";
    // URL de la pagina
    locationText += "<br> href : " + location.href + "<br>";
    // hostname
    locationText += "<br> hostname : " + location.hostname + "<br>";
    // pathname de la pagina
    locationText += "<br> pathname : " + location.pathname + "<br>";
    // protocolo de la pagina
    locationText += "<br> protocolo : " + location.protocol + "<br>";
    // hash o ancla de la pagina
    locationText += "<br> hash : " + location.hash + "<br>";
    // host y puerto de la pagina
    locationText += "<br> host y puerto : " + location.host + "<br>";
    // host , puerto y protocolo de la pagina
    locationText += "<br> host,puerto y protocolo : " + location.origin
    + "<br>";
    //search de la pagina ?user=ada
    locationText += "<br> search : " + location.search + "<br>";

    // METODOS
    // assign(): asignar un nuevo documento a la pagina en la que
    estamos
    function nuevoDocumento(){
        location.assign("http://www.google.com");
    };
    // reload(): recarga la pagina
    function recargar(){
        location.reload();
    };
    // replace(): sustituye la pagina, hace lo mismo que el assign pero
    pierde el historial (la flecha para atras desaparece)
    function sustituye(){
        location.replace("http://www.google.com");
    };
    document.write(locationText);
</script>

<button onclick="crearVentana()">Crear ventana</button>
<button onclick="cerrarVentana()">Cerrar ventana</button>
<button onclick="redimensionarVentana()">Redimensionar
ventana</button>

```

```

<button onclick="resizeToVentana()">Resize to ventana</button>
<button onclick="moveByVentana()">Move by ventana</button>
<button onclick="moveToVentana()">Move to ventana</button>
<button onclick="scrollByVentana()">Scroll by ventana</button>
<button onclick="scrollToVentana()">Scroll to ventana</button>
<button onclick="focusVentana()">Focus ventana</button>
<button onclick="blurVentana()">Blur ventana</button>
<button onclick="printFoco()">Print</button>
<button onclick="printVentana()">Print ventana</button>

<button onclick="clearTimeout(temporizador)">
  Quitar setTimeout "hola"
</button>
<button onclick="clearTimeout(temporizador2)">
  Quitar setTimeout "adios"
</button>
<button onclick="clearInterval(interval1)">
  Quitar setInterval "hola"
</button>
<button onclick="clearInterval(interval2)">
  Quitar setInterval "adios"
</button>
<br>
<br>

<button onclick="atras()">Atras</button>
<button onclick="adelante()">Adelante</button>
<button onclick="ir()">Ir a una página</button>
<br>
<br>

<button onclick="nuevoDocumento()">Assign</button>
<button onclick="recargar()">Reload</button>
<button onclick="sustituye()">Replace</button>
</body>

```

GEOPOSICIONAMIENTO

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>navigator.geolocation.getCurrentPosition</title>
</head>
<body>

  <div>

```

```

<h1>WINDOW.NAVIGATOR.GEOLOCATION.GETCURRENTPOSITION</h1>
<p>Tu posición es: <span id="Pos">Desconocida</span></p>
<p id="PosLink"><a></a></p>
<button id="boton">Localiza las coordenadas</button>
</div>

<script>

    // FORMA 1 DE GEOPOSICIONAMIENTO

    /*const options = {
        enableHighAccuracy: true,
        timeout: 5000,
        maximumAge: 0,
    };

    function success(pos) {
        const crd = pos.coords;

        console.log("Tu ubicación actual es:");
        console.log(`Latitud : ${crd.latitude}`);
        console.log(`Longitud: ${crd.longitude}`);
        console.log(`Más o menos ${crd.accuracy} metros.`);
    }

    function error(err) {
        console.warn(`ERROR(${err.code}): ${err.message}`);
    }

    navigator.geolocation.getCurrentPosition(success, error,
options);*/

    // FORMA 2 DE GEOPOSICIONAMIENTO

    let coordenadas;
    let linkCoordenadas;

    let botonPulsar = document.getElementById("boton");

    botonPulsar.addEventListener("click", ()=>{
        coordenadas = document.getElementById("Pos");
        linkCoordenadas = document.querySelector('#PosLink > a');

        if(navigator.geolocation){
            navigator.geolocation.getCurrentPosition(success, error);
        }
    });

    function success (posicion) {

```

```

        let lat = posicion.coords.latitude;
        let long = posicion.coords.longitude;

        coordenadas.textContent = `${lat}, ${long}`;
        linkCoordenadas.href =
`https://maps.google.com/?q=${lat},${long}`;
        linkCoordenadas.textContent = `Mostrar tu posición en google
maps`;

    };

    function error (err) {
        console.warn(err.message);
        let msg;
        switch (err.code) {
            case err.PERMISSION_DENIED:
                msg = "No nos has dado permiso para obtener tu
posición"
                break;
            case err.POSITION_UNAVAILABLE:
                msg = "Tu posición actual no está disponible"
                break;
            case err.TIMEOUT:
                msg = "No se ha podido obtener tu posición en un
tiempo prudencial"
                break;
            default:
                msg = "Error desconocido";
                break;
        }
        coordenadas.textContent = msg;
    };
</script>
</body>
</html>

```

OBJETOS

```

// DEFINICION
// Existen 3 maneras
// 1. Defino y creo un objeto simple utilizando un literal o
variable
let persona1 = {
    nombre: "Oscar",
    apellido: "Delgado",
    anio: 2000,
};

```

```

//2. Con la palabra new
let persona2 = new Object();
persona2.nombre = "Isaac";
persona2.apellido = "Perez";
persona2.anio = 2000;

// 3. Definir un constructor de un objeto. Cuando se definan
objetos ponemos mayuscula
function Persona(nom, ape, fecha) {
    this.nombre = nom;
    this.apellido = ape;
    this.anio = fecha;
}
let persona3 = new Persona("David", "Lopez", 1980);
let persona4 = new Persona("Sara", "Gonzalez", 1990);

// PROPIEDADES
// nombreObjeto.propiedad
// nombreObjeto["propiedad"]
// nombreObjeto[expresion]

alert(persona3.nombre);
alert(persona3["apellido"]);
let expresion = "anio";
alert(persona3[expresion]);

// Recorrer las propiedades del objeto
// for (nombre_variable in nombre_objeto) {
//     nomrbe_objeto[nombre_variable];
// }

for (propiedad in persona3) {
    alert(propiedad + " : " + persona3[propiedad]);
}
// Añadir propiedad
persona3["nacimiento"] = "Soria";
alert(persona3["nacimiento"]);

persona3.telefono = 975258642;
alert(persona3.telefono);

console.log(persona3);
// Borrar propiedad
delete persona3.telefono;
alert(persona3.telefono);
// Para ver la estructura del objeto con console.log
console.log(persona3);

// MÉTODOS

```



```
let persona5 = {
  nombre: "Javier",
  apellido: "Gutierrez",
  anio: 1995,
  telefono: 676985214,
  informacion: function () {
    return (
      " Hola mi nombre es : " +
      this.nombre +
      " , " +
      this.apellido +
      " , he nacido en " +
      this.anio +
      " y mi telefono es " +
      this.telefono
    );
  },
};

alert(persona5.informacion());

//  PROTOTIPOS
function Profesor(nom,ape,fecha) {
  this.nombre= nom,
  this.apellido= ape,
  this.anio= fecha,

  this.informacion= function () {
    return
      " Hola mi nombre es : " +
      this.nombre +
      " , " +
      this.apellido +
      " , he nacido en " +
      this.anio +
      " y mi telefono es " +
      this.telefono;

  }
};

let joel = new Profesor ("joel","callejo",2000);
let profe2=new Profesor ("jacinto","puerro",1900);
joel.nacionalidad= "española";

alert(joel.nacionalidad);
alert(profe2.nacionalidad);
```

```

        // Añadir una propiedad a un prototipo, tenemos dos formas:
        // 1. Añadiendo directamente en el prototipo
        // 2. Mediante la sintaxis:
nombre_objeto.prototype.propiedad/funcion (Profesor.prototype.coche =
true)

Profesor.prototype.coche = true;
alert(joel.coche);
console.log(joel);
console.log(profe2);

// Método -> .hasOwnProperty(propiedad) te devuelve true si
existe la propiedad para ese objeto y false si no existe.

```

EJERCICIOS OBJETOS

```

<!-- 1. Crea un objeto persona con las propiedades nombre, edad y ciudad
e imprimir el objeto entero y cada propiedad en la consola -->
<!-- 2. Crea un objeto libro con las propiedades titulo, autor y año
de edicion, y modificar las propiedades del libro y mostrar el objeto
actualizado en la consola -->
<!-- 3. Crea un objeto vehiculo con las prop marca y modelo y agregar
una propiedad color al objeto y se le asigna un valor -->
<!-- 4. Crea un objeto usuario con las propiedades nombre, email y
edad y elimina cualquiera de ellas y muestra el objeto resultante -->
<!-- 5. Crear un objeto calculadora y cada metodo debe tomar dos
numeros por parametro y devolver el resultado -->
<!-- 6. Crea un objeto persona con las prop nombre, apellido, edad,
localidad y una propiedad que sea saludar que imprima todos los datos de
la persona -->
<!-- 7. Crea un objeto animal con nombre, raza, edad y color y una
funcion que cuente y devuelva el numero de propiedades del objeto -->
<!-- 8. Teniendo un objeto persona con las propiedades nombre y edad
escribir una funcion que tome el objeto y modifique la edad -->
<!-- 9. Crear un objeto rectangulo con las prop largo y ancho y
agregar una propiedad area que calcule el area -->
<!-- 10. Crear un objeto empleado con las propiedades que se
quieran y una funcion que elimine todas las prop del objeto -->
<!-- 11. Escribir una funcion que tome dos parametros clave y valor y
devuelva ese objeto con la clave y el valor -->

<script>
// 1. Crea un objeto persona con las propiedades nombre, edad y
ciudad e imprimir el objeto entero y cada propiedad en la consola

let persona1 = {
  nombre: "Elena",
  edad: 20,
  ciudad: "Soria",

```

```
};  
console.log(persona1);  
  
for (propiedad in persona1) {  
    console.log(propiedad + " : " + persona1[propiedad]);  
}
```

// 2. Crea un objeto libro con las propiedades titulo, autor y año de edicion, y modificar las propiedades del libro y mostrar el objeto actualizado en la consola

```
let libro = new Object();  
libro.titulo = "El mar";  
libro.autor = "Maria Fernandez";  
libro.edicion = 2000;  
  
libro["titulo"] = "las olas";  
libro["autor"] = "Juana";  
libro["edicion"] = 2004;  
  
console.log(libro);
```

// 3. Crea un objero vehiculo con las prop marca y modelo y agregar una propiedad color al objeto y se le asigna un valor

```
let vehiculo = {  
    marca: "Seat",  
    modelo: "Panda",  
};  
  
vehiculo.color = "negro";  
console.log(vehiculo);
```

// 4. Crea un objeto usuario con las propiedades nombre, email y edad y elimina cualquiera de ellas y muestra el objeto resultante

```
let usuario = new Object();  
usuario.nombre = "Paula";  
usuario.email = "paula@gmail.com";  
usuario.edad = 25;  
  
if (usuario.hasOwnProperty("nombre")) {  
    delete usuario.nombre;  
} else {  
    usuario.nombre = "Paula";  
}  
  
console.log(usuario);
```

// 5. Crear un objeto calculadora y cada metodo debe tomar dos numeros por parametro y devolver el resultado

```
let calculadora = {  
  sumar: function (num1, num2) {  
    return num1 + num2;  
  },  
  restar: function (num1, num2) {  
    return num1 - num2;  
  },  
  multiplicar: function (num1, num2) {  
    return num1 * num2;  
  },  
};  
  
console.log(calculadora.sumar(2, 3));
```

// 6. Crea un objeto persona con las prop nombre, apellido, edad, localidad y una propiedad que sea saludar que imprima todos los datos de la persona

```
let persona2 = {  
  nombre: "Lorena",  
  apellido: "Pérez",  
  edad: 20,  
  localidad: "Burgos",  
  saludar: function () {  
    return (  
      " Hola mi nombre es : " +  
      this.nombre +  
      " " +  
      this.apellido +  
      " , tengo " +  
      this.edad +  
      " y soy de " +  
      this.localidad  
    );  
  },  
};  
  
console.log(persona2.saludar());
```

// 7. Crea un objeto animal con nombre, raza, edad y color y una funcion que cuente y devuelva el numero de propiedades del objeto

```
let animal = new Object();  
animal.nombre = "bigotes";  
animal.raza = "felino";  
animal.edad = 2;
```

```

    animal.color = "marron";

    function numPropiedad(Object) {
        let contador = 0;
        for (propiedad in Object) {
            contador++;
        }
        return contador;
    }
    console.log("el animal tiene " + numPropiedad(animal) + "
propiedades");

    // 8. Tenienmdo un objeto persona con las propiedades nombre y edad
    escribir una funcion que tome el objeto y modifique la edad

    let persona3 = {
        nombre: "Juan",
        edad: 20,
    };

    function modificarEdad(Objeto, edad) {
        Objeto["edad"] = edad;
    }

    modificarEdad(persona3, 18);
    console.log(persona3);

    // 9. Crear un objeto rectangulo con las prop largo y ancho y
    agregar una propiedad area que calcule el area

    let rectangulo = new Object();
    rectangulo.largo = 20;
    rectangulo.ancho = 10;
    rectangulo.area = function () {
        let area = rectangulo.largo * rectangulo.ancho;
        return area;
    };

    console.log(rectangulo.area());

    // 10. Crear un objeto empleado con las propiedades que se
    quieran y una funcion que elimine todas las prop del objeto

    let empleado = {
        nombre: "Julio",
        antiguedad: 8,
        puesto: "calidad",
        eliminar: function () {
            for (propiedad in empleado) {

```

```

        delete empleado[propiedad];
    }
},
};

console.log(empleado);

empleado.eliminar();

console.log(empleado);

// 11. Escribir una funcion que tome dos parametros clave y valor y
devuelva ese objeto con la clave y el valor

let alumno = {};

function añadirClaveValor(clave, valor) {
    alumno[clave] = valor;
}

añadirClaveValor("nombre", "Pepe");

console.log(alumno);

```

CLASES

```

class Empleado {
    #_nombre;
    #_apellido;
    #_salario;
    constructor(nombre, apellido, salario) {
        this.#_nombre = nombre;
        this.#_apellido = apellido;
        this.#_salario = salario;
    }

    get nombre() {
        return this.#_nombre;
    }
    set nombreNuevo(nuevoNombre) {
        this.#_nombre = nuevoNombre;
    }
    get apellido() {
        return this.#_apellido;
    }
    set apellidoNuevo(nuevoApellido) {
        this.#_apellido = nuevoApellido;
    }
}

```

```

    get salario() {
        return this.#_salario;
    }
    set salarioNuevo(nuevoSalario) {
        this.#_salario = nuevoSalario;
    }
    verDatos() {
        return (
            "Hola soy " +
            this.#_nombre +
            " " +
            this.#_apellido +
            " y cobro " +
            this.#_salario +
            "€"
        );
    }
}

let empleado = new Empleado("Alejandro", "Gil", 0);
console.log(empleado.verDatos());
console.log(empleado.nombre);

empleado.nombreNuevo = "David";

console.log(empleado.nombre);

empleado.apellido = "Perez";
console.log(empleado.apellido);

console.log(empleado.salario);
empleado.salarioNuevo = 200;
console.log(empleado.salario);

console.log(empleado.verDatos());

class Delegado extends Empleado {
    #puesto;
    constructor(nombre, apellido, salario, puesto){
        super(nombre, apellido, salario);
        this.#puesto=puesto;
    };
    get puesto() {
        return this.#_puesto;
    };
    set puestoNuevo(nuevoPuesto) {
        this.#_puesto = nuevoPuesto;
    };
};

```

```

        verDatos(){
            return (
                super.verDatos()+" y mi puesto es "+this.#_puesto
            );
        }
    };

let delegado = new Delegado("Oscar","Iglesia",150,"Presidente");

console.log(delegado.nombre);
console.log(delegado.verDatos());

```

ARRAYS

```

// formas para crear un array
// se pueden rellenar con lo que queramos
let numeros = [1, 2, 3, 4, 5];
// Esta forma no la vamos a utilizar
// let numeros2=new Array (5);
console.log(numeros);
// PROPIEDADES
// length
console.log(numeros.length);

// Acceso a los valores de un array

console.log(numeros[0]);
console.log(numeros[1]);
console.log(numeros[2]);
console.log(numeros[3]);
console.log(numeros[4]);

// Acceso a la ultima posicion

console.log(numeros[numeros.length - 1]);
// Podemos utilizar numeros negativos para acceder a los ultimos
elementos
// metodo AT()
console.log(numeros.at(-1));
console.log(numeros.at(-2));
console.log(numeros.at(3));

// Multidimensional
let numerosMultidimensional = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],

```



```

];
console.log(numerosMultidimensional);
console.log(numerosMultidimensional[0][0]);
console.log(numerosMultidimensional[0]);
console.log(numerosMultidimensional[1]);
console.log(numerosMultidimensional[2]);
console.log(numerosMultidimensional[1][0]);

// Se recorren con bucles anidados

// Recorrido de los arrays
// Recorrerlo de adelante hacia atrás
for (let i = 0; i < numeros.length; i++) {
  console.log(numeros[i]);
}
// Recorrerlo de atrás hacia adelante
for (let i = numeros.length - 1; i >= 0; i--) {
  console.log(numeros[i]);
}
// Recorrer arrays multidimensional
for (let i = 0; i < numerosMultidimensional.length; i++) {
  for (let j = 0; j < numerosMultidimensional[0].length; j++) {
    console.log(numerosMultidimensional[i][j]);
  }
}

function mostrarDOM(elemento, array) {
  document.getElementById(elemento).innerHTML = "";
  for (let i = 0; i < array.length; i++) {
    document.getElementById(elemento).innerHTML +=
`<div>${array[i]}</div>`;
  }
}

let cervezas = ["Mahou", "Estrella", "Alhambra", "Coronita"];
// MUTABILIDAD
// Si ejecutamos un método de una variable y este método es inmutable la
variable por su misma no cambia el contenido/atributos/propiedades, lo
que hace es crear una copia y devolverla.
// METODOS MUTABLES
// PUSH: añade un elemento al final
console.log(numeros.push(8));
// te devuelve la LONGITUD del array
console.log(numeros.push(7));
// UNSHIFT: añade un elemento al principio
// con el console log devuelve la longitud, por eso aquí da 8, pero ha
apadido uno al inicio
console.log(numeros.unshift(9));
// POP: elimina el último elemento del array

```

```

// Devuelve el valor eliminando
console.log(numeros.pop());
// SHIFT: elimina el primer elemento del array
// Devuelve el numero que elimina
console.log(numeros.shift());
// SPLICE: agrega en cualquier posicion
// Primer elemento indica la posicion (posicion 2), el segundo las
posiciones que quieres borrar(no borrar ninguno) y el 3 lo que quieres
insertar (insertar 99)
console.log(numeros.splice(2, 0, 99));
// En la posicion 4, eliminar 1 y poner 25
console.log(numeros.splice(4, 1, 25));
// FILL: rellenar valores desde una posicion a otra posicion
let coches = ["BMW", "Audi", "SEAT", "FORD", "VOLVO"];
// 1 parametro: que nos va a rellenar
// 2 parametro: desde que posicion
// 3 parametro: hasta que posicion
// EN CASO DE QUE EL 3º PARAMETRO SEA MAYOR QUE LA LONGITUD DEL ARRAY
SOLO RELLENA EL ARRAY NO CREA MAS
coches.fill("DACIA", 0, 20000);
// SPREAD
// Tiene 3 usos:
// 1er uso:
let numeros3 = [1, 2, 3, 4];
function sumar(a, b, c, d) {
    return a + b + c + d;
}
console.log(sumar(1, 2, 3, 4));
// los ... con esto trata cada elemento del array en particular, por lo
que la suma da 10
console.log(sumar(...numeros3));
// 2º uso:
// esto concatena los dos arrays en el orden que le pongamos y podemos
añadir los datos que queremos por el medio
let spreadArray = [...numeros, 150, 200, ...numeros3];
// 3er uso:
// Crear copias
// Estamos creando otro objeto con la misma referencia por lo que si
modificas 1 modificaras el otro(estan apuntando al mismo objeto)

// let numerosCopia=spreadArray;

// spreadArray[0] =5000;
// numerosCopia[1]=7000;

// Para crear copias que al modificar 1 no se modifique el otro esta es
la manera
let numerosCopia = [...spreadArray];
spreadArray[0] = 5000;

```

```

numerosCopia[1] = 7000;

// METODOS INMUTABLES
// CONCAT: concatenar varios arrays
let almacen = numeros.concat(cervezas);

// console.log(almacen.push(35));
// INDEXOF: busca un elemento y si lo encuentra devuelve el indice y si
no devuelve -1
console.log(almacen.indexOf("Mahou"));
console.log(almacen.indexOf(99));
// Si no esta te devuelve un -1, si esta te devuelve el indice(la
posicion)
console.log(almacen.indexOf(270));
// Este método utiliza ===, por lo que compara valor y tipo, por eso con
"25" string da -1, porque es de otro tipo (number)
console.log(almacen.indexOf("25"));
// SLICE: genera un nuevo array a partir de un grupo de valores, hay que
pasarle dos valores el primero es la posicion en la que va a empezar y el
segundo la posicion en la que termina

// let almacen1=almacen.slice(2,5);
// En este ejemplo quitará solo el último
let almacen1 = almacen.slice(0, -1);

// ARRAYS DE OBJETOS

let lenguajesProg = [
  {
    nombre: "CSS",
    horas: 100,
    precio: 50,
  },
  {
    nombre: "HTML",
    horas: 75,
    precio: 25,
  },
  {
    nombre: "js",
    horas: 200,
    precio: 175,
  },
  {
    nombre: "PYTHON",
    horas: 175,
    precio: 150,
  },
];

```

```

// Cuando te permite llamar a la funcion antes de declararla se llama
hoisting
// mostrarObjetosDOM("array1",lenguajesProg);

// function mostrarObjetosDOM(elemento,array){
//   document.getElementById(elemento).innerHTML="";
//   for(let i=0;i<array.length;i++){
//     document.getElementById(elemento).innerHTML
//     +=`<div>${array[i].nombre} ${array[i].horas}
//     ${array[i].precio}</div>`;
//   }
// };

// Funciones de primer orden VS funcioens de orden superior
// Primer orden: son funciones que pueden ser mandadas por parámetro
let fn = function () {
  console.log("Funcion de primer orden");
};
fn();

// Orden superior: son funciones que pueden recibir funciones como
parámetros
function ordenSuperior(funcion) {
  console.log("Algo antes");
  funcion();
  console.log("Algo después");
}
ordenSuperior(fn);

let fnArrow = () => console.log("funcion flecha");

ordenSuperior(fnArrow);

// FUNCIONES RECORRER
// for...of

// mostrarObjetosDOM("array2", lenguajesProg);

function mostrarObjetosDOM(elemento, array) {
  document.getElementById(elemento).innerHTML = "";
  for (let e of array) {
    document.getElementById(
      elemento
    ).innerHTML += `<div>${e.nombre} ${e.horas} ${e.precio}</div>`;
  }
}

// forEach

```

```

lenguajesProg.forEach((lenguaje, indice) => {
  console.log(lenguaje);
  console.log(
    `El lenguaje es ${lenguaje.nombre} y está en el indice ${indice}`
  );
});

// SORT
let palabrasOrdenar = ["BMW", "Audi", "SEAT", "FORD", "VOLVO"];

console.log(palabrasOrdenar.sort());
let numerosDesordenados = [88, 24, 2, 99, 8, 15];
// Con numeros sort simplemente, lo compara segun orden alfabético, por
// eso hay que pasarle dos parametros y que los vaya ordenando de dos en
// dos, de esta manera te lo ordena.
console.log(numerosDesordenados.sort((a, b) => a - b));
// descendente
console.log(numerosDesordenados.sort((a, b) => b - a));
// ORDENAR DOS OBJETOS
// no funciona este ejemplo pq lenguajes prog esta definida despues
console.log(lenguajesProg.sort((a, b) => a.precio - b.precio));
lenguajesProg.sort((a,b)=>{
  if(a.nombre>b.nombre){
    return 1;
  }
  if (a.nombre<b.nombre){
    return -1;
  }
  return 0;
});
console.log(lenguajesProg);

// find: nos permite encontrar el PRIMER ELEMENTO que cumple la
// condicion. SOLO DEVUELVE EL PRIMER ELEMENTO ENCONTRADO
let encontrado=lenguajesProg.find(elemento=>elemento.nombre === "HTML");
console.log(encontrado);
// devuelve solo el primero que encuentra
// let precioMayor=lenguajesProg.find(elemento=>elemento.precio >100);
// console.log(precioMayor);

// FILTER, devuelve un vector con el resultado
let precioMayor=lenguajesProg.filter(elemento=>elemento.precio >100);
console.log(precioMayor);

// EVERY: RECORRE TODOS LOS ELEMENTOS DEL ARRAY SI TODOS CUMPLEN LA
// CONDICION DEVUELVE UN TRUE SI HAY ALGUN ELEMENTO QUE NO CUMPLE LA
// CONDICION DEVOLVERA UN FALSE

```

```

// en este caso devuelve true porque todos los elementos tienen mas de 70
horas
let funcionEvery=e=>e.horas>70;
console.log(lenguajesProg.every(funcionEvery));
// Lo mismo que lo anterior escrito de otra manera
console.log("HOLA "+ lenguajesProg.every(e=>e.horas<1));

// SOME: RECORRE TODOS LOS ELEMENTOS DEL ARRAY CON QUE UN ELEMETO CUMPLA
LA CONDICION YA DEVUELVE TRUE
let funcionSome=e=>e.horas>199;
console.log(lenguajesProg.some(funcionSome));

// REVERSE: metodo mutable (devuelve el mismo array invertido), revierte
los elementos de un array

console.log(numerosDesordenados.reverse());

//map:(INMUTABLE)

let lenguajesProg2 =lenguajesProg.map(elemento=>{
  return {
    nombre: elemento.nombre,
    horas: elemento.horas,
    precio: elemento.precio / 2
  }
})
console.log(lenguajesProg2);
// Asi te devuelve un array
let lenguajesProg3 =lenguajesProg.map(elemento=>{
  return elemento.nombre;
})
console.log(lenguajesProg3);

// Te devuelve objetos con el nombre (por poner las llaves)
let lenguajesProg4 =lenguajesProg.map(elemento=>{
  return {
    nombre:elemento.nombre
  }
});
console.log(lenguajesProg4);

// eliminar duplicados: Objeto SET -> es una coleccion especial donde no
se permiten duplicados

let repetidos = ["Madrid", "Soria", "Burgos",
"Zamora","Madrid","Soria","Burgos"];

```

```

console.log(repetidos);
let sinRepetidos= new Set(repetidos);
console.log(sinRepetidos);
// tamaño. no se usa length por no ser un array , es un objeto set
console.log(sinRepetidos.size);
// Al querer añadir un elemento repetido no te va a dar ningun fallo pero
no te lo va a añadir
console.log(sinRepetidos.add("Madrid"));
console.log(sinRepetidos);
console.log(sinRepetidos.add("Valencia"));
console.log(sinRepetidos);
console.log(sinRepetidos.delete("Valencia"));
console.log(sinRepetidos);
// dos formas de convertir SET a un array
// 1 forma con from
let arrayDesdeSet = Array.from(sinRepetidos);
console.log(arrayDesdeSet);
// 2 forma con spread

let arrayDesdeSet1 = [...sinRepetidos];
console.log(arrayDesdeSet1);

//IMPORTANTE reduce , si quieres sumar todos p e. unicamente te devuelve
un valor
// tiene 3 parámetros
// primer parámetro: la variable donde se va a acumular el resultado
// segundo parametro: valor actual del elemento que va recorriendo
// tercer parametro: la inicializacion (si es un asuma a 0). en caso de
un string " "

let sumaHoras=lenguajesProg.reduce((acumulador,valorActual)=>acumulador +
valorActual.horas,0);
console.log(sumaHoras);
// tambien sirve para concatenar
let
nombreCursos=lenguajesProg.reduce((acumulador,valorActual)=>acumulador +
" "+valorActual.nombre,"El nombre de los cursos son: ");
console.log(nombreCursos);

// ARRAY DE FUNCIONES

let arrayFunciones=[
  ()=>console.log("Primera función"),
  ()=>console.log("Segunda función"),
  ()=>console.log("Tercera función"),
  ()=>console.log("Cuarta función")
];

```

```
arrayFunciones.forEach(funcion=>funcion());  
// mostrarDOM("array1", spreadArray);  
// mostrarDOM("array2", numerosCopia);  
// mostrarDOM("array3", almacen1);
```

EJERCICIOS ARRAYS

```
// EJERCICIOS CON ARRAYS NOELIA  
  
// 1. **Imprimir cada elemento**: Crea un array con nombres de  
ciudades para imprimir cada ciudad.  
  
let ciudades = ["Madrid", "Soria", "Burgos", "Zamora"];  
for (let i = 0; i < ciudades.length - 1; i++) {  
  console.log(ciudades[i]);  
}  
  
// 2. **Filtrar números pares**: Crea un array de números del 1 al 20.  
Crear un nuevo array que solo contenga números pares.  
  
let numeros = [  
  1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
];  
  
let soloPares = numeros.filter((numero) => numero % 2 == 0);  
console.log(soloPares);  
  
// 3. **Multiplicar por dos**: Dado un array de números `[1, 2, 3, 4,  
5]`, crea un nuevo array donde cada número esté multiplicado por dos.  
  
let numeros1 = [1, 2, 3, 4, 5];  
let dobles = [];  
for (let i = 0; i < numeros1.length; i++) {  
  let multi = numeros1[i] * 2;  
  dobles.push(multi);  
}  
  
console.log(dobles);  
  
// 4. **Suma total**: Dado un array de precios `[5, 10, 15, 20]`,  
calcular la suma total.  
  
let precios = [5, 10, 15, 20];  
  
let suma = precios.reduce(  
  (acumulador, valorActual) => acumulador + valorActual,  
  0  
);
```



```
console.log(suma);

// 5. **Buscar un nombre específico**: Crea un array de nombres.
Encontrar el primer nombre que empieza con "J".

let nombres = ["Paula", "Jenni", "Paco", "Maria", "Javier", "Lola",
  "Janna"];

let nombreJ = nombres.find((nombre) => nombre.at(0) === "J");
console.log(nombreJ);

// 6. **Combinar arrays**: Dado dos arrays `[1, 2, 3]` y `[4, 5, 6]`,
combinarlos en uno solo.

let primeros = [1, 2, 3];
let segundos = [4, 5, 6];
let combinados = [...primeros, ...segundos];
console.log(combinados);

// 7. **Sacar el primer elemento**: Dado un array de colores `["rojo",
"verde", "azul"]`, eliminar y mostrar el primer elemento.

let colores = ["rojo", "verde", "azul"];
console.log(colores.shift());

// 8. **Agregar elementos al principio**: Dado un array de frutas
`["manzana", "banana"]`, agregar "naranja" al principio del array.

let frutas = ["manzana", "banana"];

frutas.unshift("naranja");

console.log(frutas[0]);

// 9. **Revertir un array**: Dado un array de letras `["a", "b", "c",
"d"]`, invertir el orden de los elementos.
let letras = ["a", "b", "c", "d"];
letras.reverse();
console.log(letras);

// 10. **Ordenar alfabéticamente**: Dado un array de nombres `["Carlos",
"Ana", "Pedro"]`, ordenarlos alfabéticamente.

let ordenar = ["Carlos", "Ana", "Pedro"];
console.log(ordenar.sort());

// 11. **Quitar duplicados**: Dado un array `[1, 2, 2, 3, 4, 4, 5]`,
crear un nuevo array sin duplicados.
```

```
let duplicados = [1, 2, 2, 3, 4, 4, 5];
let sinDuplica = new Set(duplicados);
let arraySinDuplica = [...sinDuplica];
console.log(arraySinDuplica);

// 12. **Encontrar un elemento**: Dado un array `["perro", "gato", "pez"]`, verificar si "gato" está en el array.

let animales = ["perro", "gato", "pez"];

function buscar(array, elemento) {
  if (array.indexOf(elemento) !== -1) {
    console.log(`${elemento} está en el array`);
  } else {
    console.log(`${elemento} no está en el array`);
  }
}

buscar(animales, "gato");

// 13. **Agregar elementos al final**: Dado un array `["león", "tigre"]`, agregar "elefante".

let salvajes = ["leon", "tigre"];
salvajes.push("elefante");
console.log(salvajes);

// 14. **Eliminar el último elemento**: Dado un array de colores `["amarillo", "verde", "azul"]`, eliminar el último elemento.

let color = ["amarillo", "verde", "azul"];
color.pop();
console.log(color);

// 15. **Unir elementos en una cadena**: Dado un array `["Hola", "mundo"]`, convertirlo en una cadena separada por espacios.

let palabras = ["Hola", "mundo"];

let frase = `${palabras[0]} ${palabras[1]}`;

console.log(frase);

// 16. **Encontrar el índice de un elemento**: Dado un array `["a", "b", "c", "d"]`, encontrar el índice del elemento "c".

let encontrar = ["a", "b", "c", "d"];
```

```
console.log(encontrar.indexOf("c"));

// 17. **Crear una subarray**: Dado un array `["lunes", "martes", "miércoles", "jueves", "viernes"]`, crear un nuevo array con solo los primeros tres días.

let dias = ["lunes", "martes", "miercoles", "jueves", "viernes"];
let primer = [];
for (let i = 0; i < 3; i++) {
  primer.push(dias[i]);
}
console.log(primer);

// 18. **Modificar un array**: Dado un array `[1, 2, 3, 4, 5]`, eliminar el tercer elemento.

let eliminar3 = [1, 2, 3, 4, 5];
eliminar3.splice(2, 1);
console.log(eliminar3);

// 19. **Multiplicación acumulada**: Dado un array `[1, 2, 3, 4]`, multiplicar todos los números.

let multi = [1, 2, 3, 4];

let resultado=multi.reduce((acumulador,valorActual)=> acumulador * valorActual,1);
console.log(resultado);

// 20. **Encontrar elementos mayores que un número**: Dado un array `[3, 7, 12, 5, 8]`, crear un array solo con los números mayores que 6.

let num = [3, 7, 12, 5, 8];

let mayoresDe6=num.filter(numero=>numero>6);
console.log(mayoresDe6)
```