

¿Qué es JavaScript?

JavaScript es un lenguaje de programación de alto nivel y orientado a objetos que se utiliza principalmente para desarrollar aplicaciones web interactivas. Fue creado en 1995 por Brendan Eich mientras trabajaba en Netscape Communications Corporation. JavaScript se ejecuta en el lado del cliente (navegador web) y del lado del servidor (Node.js) y permite a los desarrolladores crear aplicaciones web dinámicas e interactivas que pueden responder a la interacción del usuario en tiempo real.



JavaScript es un lenguaje de programación muy popular y se utiliza en casi todas las páginas web. Algunas de las cosas que se pueden hacer con JavaScript incluyen validar formularios, crear efectos visuales como animaciones y transiciones, manipular elementos HTML y CSS, interactuar con API de terceros, crear juegos y aplicaciones móviles híbridas, y mucho más.

JavaScript se ha convertido en un lenguaje muy versátil y esencial en el desarrollo web moderno, y su popularidad ha llevado al desarrollo de una amplia variedad de frameworks y bibliotecas como React, Angular, Vue.js, jQuery, entre otros.

Usos

JavaScript se utiliza en una amplia variedad de aplicaciones y contextos en el desarrollo web. Aquí te muestro algunos de los usos más comunes de JavaScript:

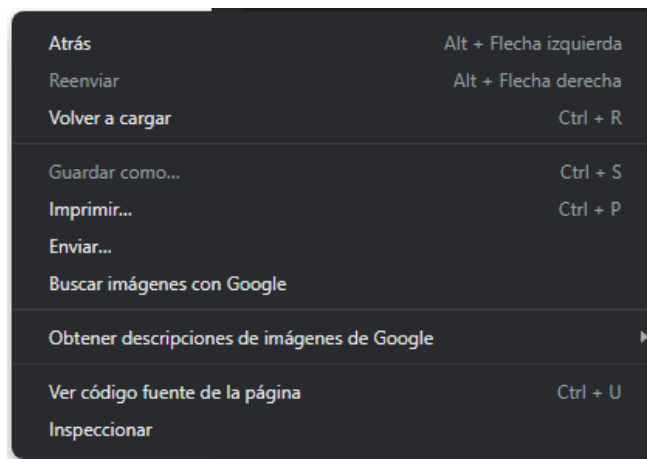
- Creación de aplicaciones web dinámicas: JavaScript se utiliza para crear aplicaciones web interactivas y dinámicas que pueden actualizar y cambiar el contenido de una página web sin necesidad de recargarla.
- Validación de formularios: JavaScript se utiliza para validar los datos ingresados en los formularios de una página web antes de enviarlos al servidor.

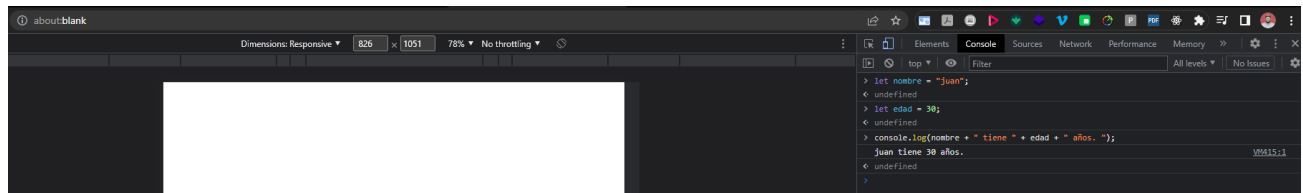
```
15
16 const LOCALE = globalThis.navigator.language
17
18 const div = document.body.appendChild(document.createElement('div'))
19 const list = div.appendChild(document.createElement('ol'))
20
21 const dayNames = new Map()
22
23 for (let i = 0; i < 7; ++i) {
24   const d = Temporal.PlainDate.from({
25     year: Temporal.Now.plainDateISO().year,
26     month: 1,
27     day: i + 1,
28   })
29   dayNames.set(d.dayOfWeek, d.toLocaleString(LOCALE, { weekday: 'long' }))
30 }
31
32
33 for (const num of [...dayNames.keys()].sort((a, b) => a - b)) {
34   list.appendChild(Object.assign(
35     document.createElement('li'),
36     { textContent: dayNames.get(num) },
37   ))
38 }
39
```

- Efectos visuales y animaciones: JavaScript se utiliza para crear efectos visuales y animaciones en una página web, lo que puede mejorar la experiencia del usuario y hacer que la página sea más atractiva.
- Manipulación de elementos HTML y CSS: JavaScript se utiliza para manipular y cambiar elementos HTML y CSS en una página web en tiempo real.
- Interacción con API de terceros: JavaScript se utiliza para interactuar con las API de terceros, lo que permite a los desarrolladores integrar la funcionalidad de otros servicios en sus propias aplicaciones.
- Creación de juegos: JavaScript se utiliza para crear juegos en línea y aplicaciones de entretenimiento en la web.
- Aplicaciones móviles híbridas: JavaScript se utiliza para crear aplicaciones móviles híbridas, que son aplicaciones móviles que se ejecutan en un navegador web integrado en una aplicación nativa.
- Desarrollo de extensiones de navegador: JavaScript se utiliza para desarrollar extensiones de navegador que agregan funcionalidad adicional a los navegadores web.
- Análisis de datos y visualización: JavaScript se utiliza para analizar datos y crear visualizaciones interactivas en una página web.
- Programación del lado del servidor: JavaScript se utiliza para programar del lado del servidor mediante el uso de plataformas como Node.js, que permite a los desarrolladores crear aplicaciones web escalables y de alto rendimiento.

Fundamentos JS

Para estos ejercicios prácticos utiliza el navegador web abre una página en blanco **about:blank** y utiliza la consola de las herramientas de desarrollador, clic derecho inspeccionar.





1. Variables y tipos de datos:

Las variables son contenedores que almacenan valores. JavaScript tiene varios tipos de datos, como cadenas de texto, números, booleanos, arrays, entre otros.

Ejercicio práctico: Declara una variable llamada "nombre" y asígnale tu nombre. Luego, declara otra variable llamada "edad" y asígnale tu edad. Finalmente, imprime en la consola la concatenación de ambas variables.

```
let nombre = "Juan";
let edad = 30;
console.log(nombre + " tiene " + edad + " años.");
```

2. Operadores:

Los operadores son símbolos que se utilizan para realizar operaciones matemáticas y de comparación en JavaScript.

Ejercicio práctico: Declara dos variables numéricas y realiza varias operaciones con ellas, como suma, resta, multiplicación y división.

```
let num1 = 10;
let num2 = 5;
console.log(num1 + num2); // suma
console.log(num1 - num2); // resta
console.log(num1 * num2); // multiplicación
console.log(num1 / num2); // división
```

3. Condicionales:

Los condicionales se utilizan para tomar decisiones en función de una determinada condición.

Ejercicio práctico: Declara una variable numérica y utiliza un condicional para imprimir en la consola si el número es mayor o menor que 10.

```
let num = 15;
if (num > 10) {
  console.log("El número es mayor que 10");
} else {
  console.log("El número es menor o igual a 10");
}
```

4. Bucles:

Los bucles se utilizan para repetir una acción varias veces.

Ejercicio práctico: Utiliza un bucle "for" para imprimir en la consola los números del 1 al 10.

```
for (let i = 1; i <= 10; i++) {
  console.log(i);
}
```

5. Funciones:

Las funciones son bloques de código reutilizables que se utilizan para realizar una tarea específica.

Ejercicio práctico: Crea una función que reciba dos números como parámetros y devuelva la suma de ambos.

```
function sumar(num1, num2) {
  return num1 + num2;
}
console.log(sumar(5, 7)); // 12
```

6. Arrays:

Los arrays son estructuras de datos que almacenan varios valores en una sola variable.

Ejercicio práctico: Crea un array de números y utiliza un bucle "for" para imprimir en la consola cada uno de ellos.

```
let numeros = [1, 2, 3, 4, 5];  
for (let i = 0; i < numeros.length; i++) {  
  console.log(numeros[i]);  
}
```

7. Objetos:

Los objetos son estructuras de datos que almacenan valores relacionados entre sí en propiedades y métodos.

Ejercicio práctico: Crea un objeto que represente una persona y tenga propiedades como nombre, edad y género. Luego, imprime en la consola el nombre y la edad de la persona.

```
let persona = {  
  nombre: "María",  
  edad: 25,  
  genero
```

8. Métodos de arrays:

Los arrays tienen varios métodos integrados que se utilizan para manipular y recorrer sus elementos.

Ejercicio práctico: Crea un array de números y utiliza el método "forEach" para imprimir en la consola cada uno de ellos.

```
let numeros = [1, 2, 3, 4, 5];
numeros.forEach(function(numero) {
  console.log(numero);
});
```

9. Métodos de strings:

Las cadenas de texto tienen varios métodos integrados que se utilizan para manipular y recorrer sus caracteres.

Ejercicio práctico: Crea una cadena de texto y utiliza el método "toUpperCase" para convertir todos sus caracteres a mayúsculas.

```
let texto = "Hola mundo";
console.log(texto.toUpperCase()); // "HOLA MUNDO"
```

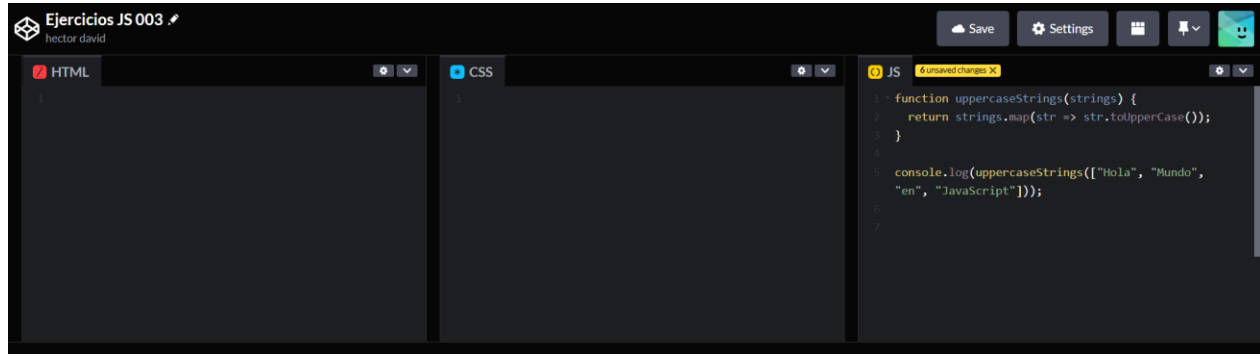
10. Eventos:

Los eventos se utilizan para detectar acciones del usuario, como hacer clic en un botón.

Ejercicio práctico: Crea un botón en HTML y utiliza JavaScript para detectar cuando se hace clic en él y mostrar un mensaje en la consola.

```
let boton = document.getElementById("miBoton");
boton.addEventListener("click", function() {
  console.log("Se hizo clic en el botón");
});
```

Ejercicios Prácticos



Para esta parte realizaremos los ejercicios en la plataforma [codepen](https://codepen.io), se registran en la plataforma y desarrollarán cada uno de los ejercicios en ella, compartirán el enlace de su perfil con los ejercicios realizados y los subirán a plataforma para revisarlos.

Ejercicios

1. Crea una función que tome dos números como argumentos y devuelva su suma.
2. Crea una función que tome una cadena como argumento y devuelva el número de caracteres que tiene.
3. Crea una función que tome un array de números como argumento y devuelva el número más grande.
4. Crea una función que tome una cadena como argumento y devuelva la misma cadena pero con todas las letras en mayúsculas.
5. Crea una función que tome un objeto como argumento y devuelva un array con todas las claves del objeto.
6. Crea una función que tome un array de números como argumento y devuelva un nuevo array con los números ordenados de menor a mayor.
7. Crea una función que tome dos fechas como argumentos y devuelva el número de días que hay entre ellas.
8. Crea una función que tome un array de strings como argumento y devuelva un nuevo array con la longitud de cada string.
9. Crea una función que tome un objeto como argumento y devuelva un nuevo objeto con las mismas claves pero con los valores invertidos.
10. Crea una función que tome un número como argumento y devuelva un array con todos los números enteros desde 1 hasta ese número.

11. Crea una función que tome un array de strings como argumento y devuelva un nuevo array con todas las strings en mayúsculas.
12. Crea una función que tome dos strings como argumentos y devuelva true si son iguales (sin importar mayúsculas o minúsculas) o false si son diferentes.
13. Crea una función que tome un array de objetos y un string como argumentos y devuelva un nuevo array con todos los objetos que contienen una propiedad con ese string como clave.
14. Crea una función que tome un array de números como argumento y devuelva true si todos los números son pares o false si alguno es impar.