⚡ ByteNova

JavaScript para principiantes

Guía práctica desde cero

Aprenderás:
variables, tipos de datos, condiciones, ciclos, funciones, arreglos, objetos, DOM, eventos y un proyecto práctico.

Nivel: principiante

Creado por ByteNova · 2026

Cómo utilizar esta guía

Lee cada tema en orden, escribe los ejemplos por tu cuenta y modifica los valores para comprobar que comprendes el resultado. La práctica constante es más importante que memorizar instrucciones.

> Recomendación
> Crea una carpeta llamada practica-javascript con los archivos index.html, style.css y script.js.

Contenido

1. ¿Qué es JavaScript?

JavaScript es un lenguaje de programación que permite agregar comportamiento e interactividad a páginas web.

HTML organiza el contenido, CSS controla la apariencia y JavaScript responde a acciones como clics, escritura en formularios o cambios en la página.

```
console.log("Hola desde JavaScript");
```

Esta instrucción muestra un mensaje en la consola del navegador.

Usos frecuentes

- Abrir y cerrar menús móviles.
- Validar formularios.
- Crear modo oscuro.
- Actualizar textos e imágenes.
- Mostrar mensajes y ventanas.
- Construir aplicaciones web interactivas.

> Idea clave
> JavaScript no reemplaza a HTML ni a CSS; trabaja junto a ellos.

2. Conectar JavaScript con HTML

Guarda el código en un archivo separado llamado script.js y enlázalo antes de cerrar el elemento body.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mi proyecto</title>
</head>
<body>
    <h1>Mi primera página interactiva</h1>

    <script src="script.js"></script>
</body>
</html>
```

Dentro de script.js:

```
console.log("El archivo JavaScript funciona");
```

Problema común
Si no aparece el mensaje, revisa la ruta del archivo y guarda los cambios antes de recargar.

3. Consola del navegador

La consola permite comprobar resultados y encontrar errores. Normalmente se abre con F12 y luego seleccionando la pestaña Console o Consola.

```
console.log("Mensaje normal");
console.warn("Advertencia");
console.error("Mensaje de error");
```

Los mensajes de error suelen indicar el archivo y la línea aproximada donde ocurrió el problema.

4. Variables y tipos de datos

Una variable guarda información. Utiliza const cuando no necesitas reasignar el valor y let cuando el valor cambiará.

```
const nombre = "Ana";
let puntos = 0;

puntos = puntos + 10;
console.log(nombre, puntos);
```

Tipos de datos básicos

Tipo	Ejemplo	Descripción
String	"ByteNova"	Texto
Number	25	Números enteros o decimales
Boolean	true	Verdadero o falso
Null	null	Ausencia intencional de valor
Array	["HTML", "CSS"]	Colección ordenada
Object	{ nombre: "Ana" }	Datos agrupados por propiedades

```
const edad = 20;
console.log(typeof edad); // number
```

5. Operadores

Los operadores permiten realizar cálculos, comparar valores y combinar condiciones.

```
const suma = 10 + 5;
const resta = 10 - 5;
const multiplicacion = 10 * 5;
const division = 10 / 5;
```

Comparaciones frecuentes:

- === igualdad estricta
- !== diferente
- > mayor que
- < menor que
- >= mayor o igual
- <= menor o igual

```
const edad = 20;
console.log(edad >= 18); // true
```

No confundas = con ===
Un signo igual asigna un valor. Tres signos iguales comparan valor y tipo de dato.

6. Condiciones

Las condiciones ejecutan instrucciones diferentes según una expresión sea verdadera o falsa.

```
const edad = 20;

if (edad >= 18) {
    console.log("Eres mayor de edad");
} else {
    console.log("Eres menor de edad");
}
```

También puedes comprobar varias posibilidades con else if.

```
const nota = 85;

if (nota >= 90) {
    console.log("Excelente");
} else if (nota >= 70) {
    console.log("Aprobado");
} else {
    console.log("Debes mejorar");
}
```

7. Ciclos

Los ciclos repiten instrucciones. Utiliza un ciclo cuando la misma tarea debe ejecutarse varias veces.

Ciclo for

```
for (let numero = 1; numero <= 5; numero++) {
    console.log(numero);
}
```

Recorrer una colección

```
const tecnologias = ["HTML", "CSS", "JavaScript"];

for (const tecnologia of tecnologias) {
    console.log(tecnologia);
}
```

> Precaución
> Una condición que nunca cambia puede crear un ciclo infinito.

8. Funciones

Una función agrupa instrucciones reutilizables. Puede recibir datos mediante parámetros y devolver un resultado.

```
function saludar(nombre) {
    return `Hola, ${nombre}`;
}

const mensaje = saludar("Ana");
console.log(mensaje);
```

Funciones flecha

```
const sumar = (numeroUno, numeroDos) => {
    return numeroUno + numeroDos;
};

console.log(sumar(5, 7));
```

Buena práctica
Crea funciones pequeñas, con una tarea clara y nombres descriptivos.

9. Arreglos y objetos

Un arreglo guarda varios valores en orden. Las posiciones comienzan en cero.

```
const articulos = ["HTML", "CSS", "JavaScript"];

console.log(articulos[0]);
articulos.push("Python");
console.log(articulos.length);
```

Un objeto agrupa propiedades relacionadas.

```
const estudiante = {
    nombre: "Ana",
    edad: 20,
    activo: true
};

console.log(estudiante.nombre);
estudiante.edad = 21;
```

Recorrer un arreglo con forEach

```
articulos.forEach(function (articulo) {
    console.log(articulo);
});
```

10. Manipular el DOM

El DOM representa los elementos HTML para que JavaScript pueda encontrarlos y modificarlos.

HTML de ejemplo:

```
<h1 id="titulo">Título original</h1>
<p class="mensaje">Contenido inicial</p>
```

JavaScript:

```
const titulo = document.getElementById("titulo");
const mensaje = document.querySelector(".mensaje");

titulo.textContent = "Nuevo título";
mensaje.classList.add("destacado");
```

Métodos útiles

- getElementById(): busca por identificador.
- querySelector(): devuelve la primera coincidencia.
- querySelectorAll(): devuelve varias coincidencias.

- textContent: cambia el texto.
- classList.add/remove/toggle: administra clases CSS.

11. Eventos

Un evento ocurre cuando una persona interactúa con la página.

```
const boton = document.getElementById("boton");

boton.addEventListener("click", function () {
    console.log("Presionaste el botón");
});
```

Eventos frecuentes:

- click: presionar un elemento.
- input: escribir dentro de un campo.
- submit: enviar un formulario.
- keydown: presionar una tecla.
- change: cambiar una opción.

12. Proyecto práctico: contador de clics

Construiremos un contador que aumenta, disminuye y se reinicia.

HTML

```
<main class="contador">
    <h1>Contador de clics</h1>
    <p id="resultado">0</p>

    <button id="aumentar">Aumentar</button>
    <button id="disminuir">Disminuir</button>
    <button id="reiniciar">Reiniciar</button>
</main>

<script src="script.js"></script>
```

JavaScript

```
const resultado = document.getElementById("resultado");
const aumentar = document.getElementById("aumentar");
const disminuir = document.getElementById("disminuir");
const reiniciar = document.getElementById("reiniciar");

let contador = 0;

function actualizarResultado() {
    resultado.textContent = contador;
}

aumentar.addEventListener("click", function () {
    contador++;
    actualizarResultado();
});

disminuir.addEventListener("click", function () {
    contador--;
    actualizarResultado();
});

reiniciar.addEventListener("click", function () {
    contador = 0;
    actualizarResultado();
});
```

CSS opcional

```
.contador {
    max-width: 500px;
    margin: 80px auto;
    padding: 30px;
    text-align: center;
    border: 1px solid #e2e8f0;
    border-radius: 18px;
}

button {
    margin: 5px;
    padding: 10px 16px;
    cursor: pointer;
}
```

Reto adicional
Evita que el contador sea menor que cero o cambia el color del número según sea positivo o negativo.

13. Errores frecuentes

- No guardar los archivos antes de recargar.
- Conectar una ruta incorrecta hacia script.js.
- Escribir un identificador diferente en HTML y JavaScript.
- Utilizar una variable antes de declararla.
- Olvidar cerrar comillas, paréntesis o llaves.
- Seleccionar un elemento que no existe.
- Confundir = con ===.
- No revisar la consola del navegador.

```
const boton = document.getElementById("boton");

if (boton) {
    boton.addEventListener("click", function () {
        console.log("Botón encontrado");
    });
}
```

Aprender a depurar
Corrige un error a la vez y vuelve a probar después de cada cambio.

14. Ejercicios finales

Saludo personalizado

Solicita o define un nombre y muestra un mensaje en la página.

Calculadora básica

Realiza suma, resta, multiplicación y división con dos números.

Cambio de tema

Crea un botón que agregue o quite una clase de modo oscuro.

Lista de tareas

Permite agregar elementos y eliminarlos con un botón.

Validador de formulario

Impide enviar un formulario si un campo obligatorio está vacío.

Filtro de tarjetas

Muestra solamente las tarjetas que coincidan con una palabra escrita.

Plan de estudio
Completa un ejercicio por día. Guarda cada proyecto en una carpeta distinta y escribe un breve README con lo aprendido.

Conclusión

JavaScript permite transformar una página estática en una experiencia interactiva. Dominar variables, condiciones, funciones, DOM y eventos te dará una base sólida para crear proyectos más completos.

Siguiente paso
Continúa con proyectos pequeños, practica con frecuencia y utiliza Git para conservar versiones de tu código.

ByteNova

Aprende tecnología desde cero con artículos, rutas, retos, guías y recursos gratuitos.

https://bytenova.app/