

3 calidad del código

```
ALIDAD DEL CODIGO 3
index.html
main.js
index.html
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Prácticas del Curso de JavaScript</title>
7 <style>
8 {
9   font-family: Arial, sans-serif;
10  margin: 0;
11  background-color: #ffe0e0; /* Fondo rosa claro */
12  color: #333;
13 }
14 header {
15   background-color: #ffb3b3; /* Encabezado rosa más oscuro */
16   padding: 20px 0;
17   text-align: center;
18   color: #fff;
19   box-shadow: 0 2px 4px #000, 0 0 0 1px;
20 }
21 h1 {
22   margin: 0;
23   font-size: 2.5em;
24   color: #ff3366; /* Rosa fuerte para el título */
25 }
26 .container {
27   width: 80%;
28   max-width: 900px;
29   margin: 20px auto;
30   padding: 0 20px;
31 }
32 .info-bar {
33   background-color: #ffe0e0; /* Barra de información rosa muy claro */
34   padding: 10px 20px;
35   margin-bottom: 20px;
36   border-radius: 5px;
37   color: #555;
38 }
```

```
main.js
1 // TEMA: 3.1 Depuración en el navegador
2 // Este bloque demuestra cómo usar las herramientas de depuración del navegador,
3 // específicamente `console.log()` para mostrar mensajes y cómo usar `debugger`
4 // para pausar la ejecución del código.
5 // Aprendí que la consola es fundamental para entender el flujo del programa
6 function calcularSuma(a, b) {
7   let resultado = a + b;
8   console.log('Depuración 3.1: Calculando suma de ${a} y ${b}.'); // Mensaje informativo
9   // debugger; // Descomentar para pausar la ejecución aquí y abrir las Developer Tools
10  return resultado;
11 }
12
13 let num1 = 5;
14 let num2 = 10;
15 let sumaTotal = calcularSuma(num1, num2);
16 console.log('Depuración 3.1: La suma total es: ${sumaTotal}'); // Mostrar el resultado final
17
18 // Mostramos el resultado en el DOM
19 document.getElementById('output-3-1').innerHTML = `
20   <p>Revisa la consola (F12) para ver los mensajes de depuración.</p>
21   <p>Suma de ${num1} y ${num2} es: ${sumaTotal}</p>
22 `;
23
24 // TEMA: 3.2 Estilo de codificación
25 // Este bloque ilustra la importancia de un buen estilo de codificación:
26 // Indentación consistente, nombres de variables descriptivos, espacios alrededor
27 // de operadores y uso de llaves. Un buen estilo mejora la legibilidad y
28 // mantenibilidad del código.
29 // Aprendí que no es solo cuestión de estética, sino de eficiencia y colaboración.
30
31
```

```
main.js
1 // TEMA: 3.5 Pruebas automatizadas con Mocha - 20/10/2024
2 // Este bloque explica la importancia de las pruebas automatizadas para
3 // asegurar que el código funcione como se espera y evitar regresiones.
4 // Mocha es un framework popular para pruebas en JavaScript. Aquí, solo simulamos
5 // la lógica de prueba para un ejemplo simple.
6 // Aprendí que las pruebas son cruciales para un desarrollo robusto y sin errores.
7
8 // Función a ser probada
9 function esPar(numero) {
10   return numero % 2 === 0;
11 }
12
13 // Simulando la estructura de una prueba con Mocha (esto no ejecuta pruebas reales aquí)
14 // En un entorno de pruebas real, tendríamos archivos de prueba separados.
15 const output35 = document.getElementById('output-3-5');
16 output35.innerHTML = `<h3>Simulación de pruebas unitarias para `esPar(numero)`</h3>`;
17
18 function runTest(description, testFunction) {
19   try {
20     testFunction();
21     output35.innerHTML += `<p style="color: green;">✓ ${description}</p>`;
22   } catch (e) {
23     output35.innerHTML += `<p style="color: red;">✗ ${description} - ${e.message}</p>`;
24   }
25 }
26
27 runTest('esPar(2) debería ser true', () => {
28   if (!esPar(2)) throw new Error('Falló para 2');
29 });
30
```

```
index.html
1 <head>
2 </head>
3 <body>
4 <div class="container">
5   <div class="info-bar">
6     &lt;F12> para ver la mayoría de los resultados.
7   </div>
8
9   <!-- 3.1 Depuración en el navegador -->
10  <section id="tema-3-1">
11    <h2>3.1 Depuración en el navegador</h2>
12    <div class="content">
13      <p>Aquí se mostrarán mensajes en la consola del navegador para depuración.</p>
14      <div class="output" id="output-3-1"></div>
15    </div>
16  </section>
17
18  <!-- 3.2 Estilo de codificación -->
19  <section id="tema-3-2">
20    <h2>3.2 Estilo de codificación</h2>
21    <div class="content">
22      <p>Ejemplo de código con buen estilo y formato.</p>
23      <pre>code id="code-3-2"</code></pre>
24      <div class="output" id="output-3-2"></div>
25    </div>
26  </section>
27
28  <!-- 3.3 Comentarios -->
29  <section id="tema-3-3">
30    <h2>3.3 Comentarios</h2>
31    <div class="content">
32
```

```
main.js
1 // TEMA: 3.3 Comentarios - 20/10/2024
2 // Este es un ejemplo de un comentario de varias líneas, explicando la finalidad
3 // del bloque de código o de una función compleja.
4 // Los comentarios son esenciales para explicar "por qué" se hizo algo, no solo "qué" hace.
5 // Aprendí a usar diferentes tipos de comentarios para diferentes propósitos:
6 // - Comentario de una línea para explicaciones rápidas.
7 // - Comentario de múltiples líneas para detalles más extensos, como la descripción de una función.
8 /**
9  * Comentario JSDoc para documentar funciones,
10  * parámetros y valores de retorno.
11  * @param {number} a - El primer número.
12  * @param {number} b - El segundo número.
13  * @returns {number} La suma de los dos números.
14  */
15 function sumarNumeros(a, b) {
16   // Esta es una suma simple.
17   return a + b;
18 }
19
20 const sumando1 = 8;
21 const sumando2 = 12;
22 const resultadoSuma = sumarNumeros(sumando1, sumando2);
23 document.getElementById('output-3-3').innerHTML += `<p>La suma de ${sumando1} y ${sumando2} es
24
25 // TEMA: 3.4 Código ninja - 20/10/2024
26 // El "código ninja" se refiere a código extremadamente conciso pero difícil
27 // de leer y mantener. Aunque a veces parece inteligente, prioriza la brevedad
28 // sobre la claridad. Es mejor evitarlo en proyectos colaborativos.
29 // Aprendí que la legibilidad es más importante que la concisión extrema.
30
```

calidad del Curso

<F12> para ver la mayoría de los resultados.

3.1 Depuración en el navegador

Aquí se mostrarán mensajes en la consola del navegador para depuración.

3.2 Estilo de codificación

Ejemplo de código con buen estilo y formato.

3.3 Comentarios

Este tema se explica a través de los comentarios en el archivo 'main.js'.

Revisa el archivo main.js para ver los ejemplos de comentarios.