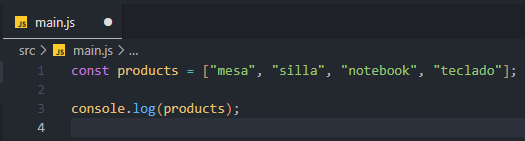
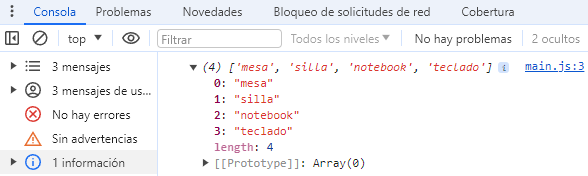
Arreglos en JavaScript

En esta sección vamos a definir un arreglo en JavaScript, aplicar los métodos **push** y **forEach** en arreglos, las distintas formas de cómo iterar con los elementos de un arreglo e imprimir un elemento de un arreglo por su índice.

Definición de un arreglo

Un arreglo es una estructura de datos que se define utilizando corchetes **[ ]**. A diferencia de los objetos, que usan pares clave-valor para organizar datos, un arreglo contiene una colección ordenada de elementos. Cada elemento en el arreglo tiene un índice numérico que comienza desde **0** y se utiliza para acceder a ese elemento específico en el arreglo.

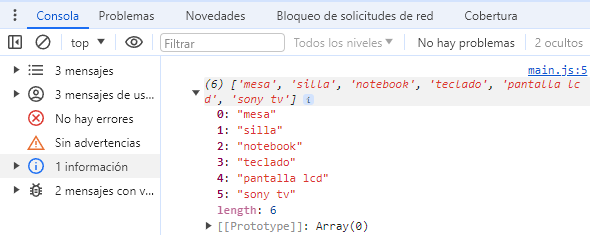




Agregar elementos en un arreglo

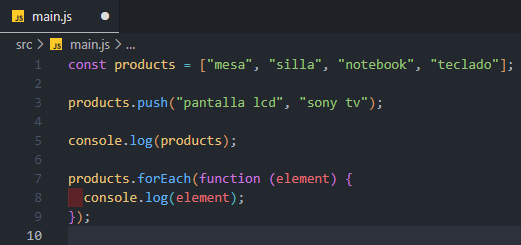
Para agregar varios elementos a la vez, se utilizar el método **push** con múltiples argumentos. Esto permite agregar elementos al final del arreglo.

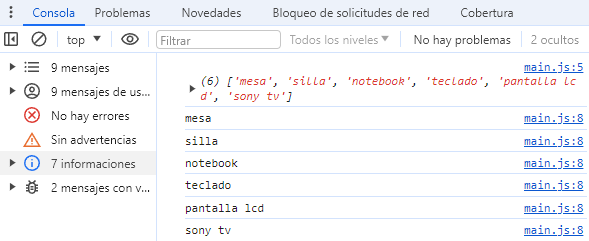




Iterar con los elementos de un arreglo

Para iterar a través de los elementos de un arreglo, se pude utilizar el método **forEach**. Este método recibe una función de callback (puede ser una función anónima o una función de flecha) que se ejecuta una vez por cada elemento en el arreglo.



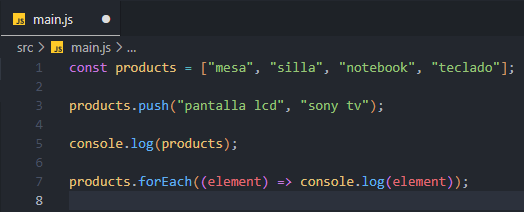


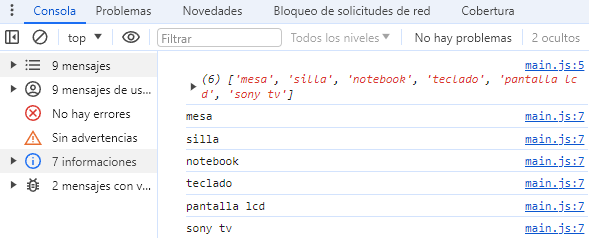
También existen otros métodos útiles como **map**, **filter** y **find**, que permiten manipular los elementos del arreglo sin modificar el original, creando una nueva instancia con los elementos modificados.

Es importante tener en cuenta que un callback en programación se refiere a una función que se pasa como argumento a otra función. Esta función de callback se ejecuta después de que cierto proceso o tarea se haya completado, lo que permite que el código continúe ejecutándose de manera asíncrona.

Método forEach con función de flecha

Como ya se había mencionado antes, el método **forEach** también se puede usar con funciones de flecha en lugar de funciones anónimas tradicionales, para hacer el código más conciso y legible.

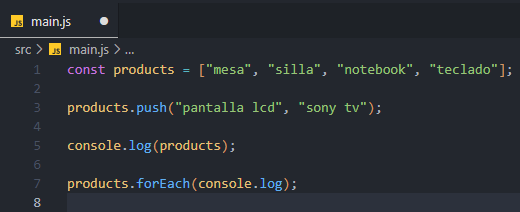




Recordar que, si la función de flecha tiene un solo parámetro, tal y como se ha mostrado en el ejemplo, se pueden omitir los paréntesis **( )** alrededor del parámetro. Si hay más de un parámetro, es necesario incluir los paréntesis. Además, si la función tiene una sola línea de código, se pueden omitir las llaves y el **return**.

Imprimir todos los argumentos que se recibe

Al utilizar **console.log** como callback en el método **forEach**, cada elemento del arreglo se imprimirá junto con su índice y el arreglo completo.

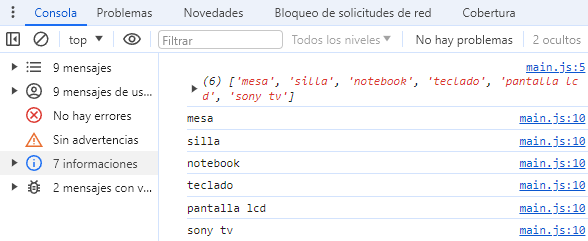




Uso de for …of

El **for …of** es una sintaxis moderna de JavaScript que permite iterar sobre objetos que son iterables, como los arreglos. A diferencia del **for** clásico, el **for...of** itera directamente sobre los valores del iterable, sin necesidad de acceder a los índices.

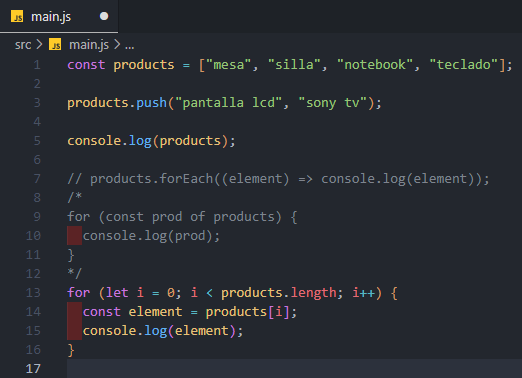


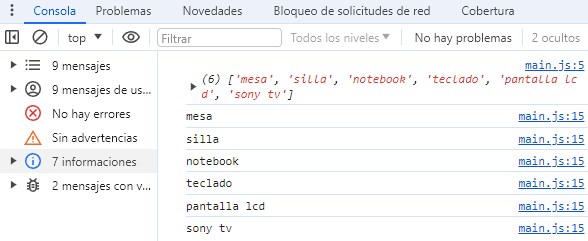


En el ejemplo mostrado, dentro del cuerpo del ciclo **for ...of**, se utiliza la constante **prod** para acceder al elemento actual del arreglo en cada iteración.

Uso de for clásico

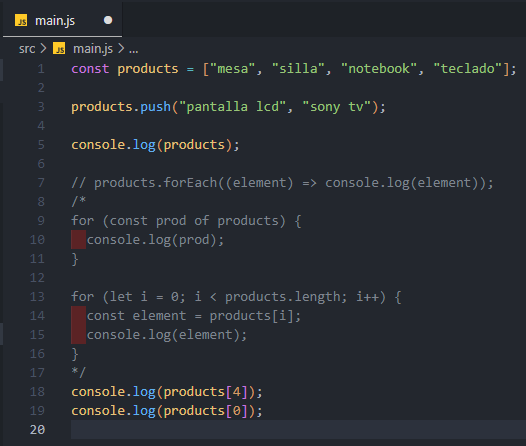
El **for** clásico es una estructura de control de flujo que permite iterar sobre una lista de elementos utilizando un índice. Además, se puede acceder tanto al índice como al valor del elemento en cada iteración.

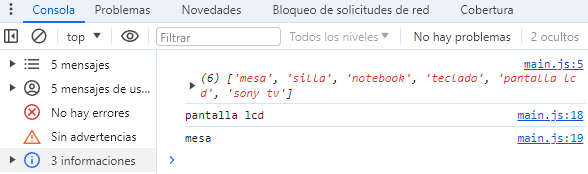




Imprimir elementos de un arreglo por su índice

Para acceder e imprimir elementos específicos de un arreglo utilizando su índice, se puede hacerlo directamente accediendo a la posición del elemento en el arreglo.





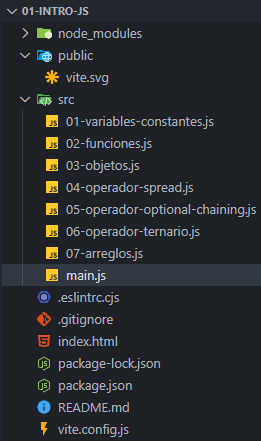
Antes de pasar al siguiente tema

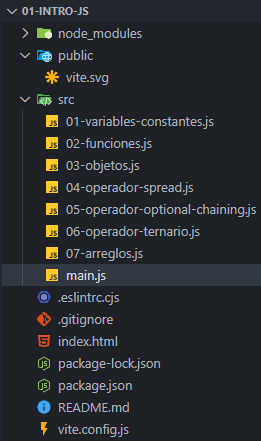
Realiza estas acciones en Visual Studio Code:

1. Abre el panel izquierdo y busca el archivo **main.js**.
2. Haz clic derecho en el archivo y selecciona **Cambiar nombre**.
3. Cambia el nombre del archivo a **07-arreglos.js** y presiona **Enter**.
4. Ahora, crea un nuevo archivo, dentro la carpeta raíz **src**, haciendo clic en el botón **Nuevo archivo** en la parte superior del panel izquierdo.



1. Nombre el nuevo archivo como **main.js** y presiona **Enter**.





A partir de este momento, se realizarán cambios en el archivo **main.js** que has creado.