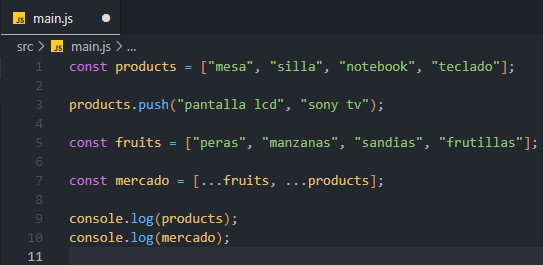
Arreglos y el operador Spread

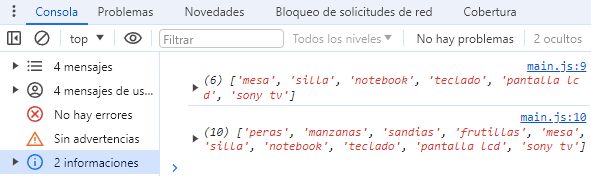
En esta sección vamos a ver como el operador Spread sirve para unir o concatenar arreglos, también como crear un nuevo arreglo con combinaciones de arreglos y elementos con el método **concat**.

Operador Spread en arreglos

El operador Spread (**...**) en JavaScript también se utiliza para expandir o concatenar arreglos, permitiendo crear un nuevo arreglo a partir de otros. Es una herramienta fundamental para crear arreglos combinados sin modificar los originales, asegurando la inmutabilidad de los datos.

Para crear un nuevo arreglo a partir de dos arreglos existentes, se puede usar el operador Spread. Al hacerlo, se preserva la integridad de los arreglos originales, ya que no se alteran.



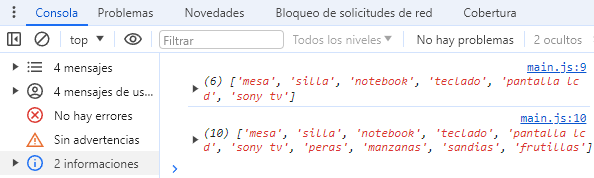


En el ejemplo mostrado, se tiene en cuenta que las constantes que definen arreglos se nombran en plural para mantener la consistencia en el código.

El orden importa

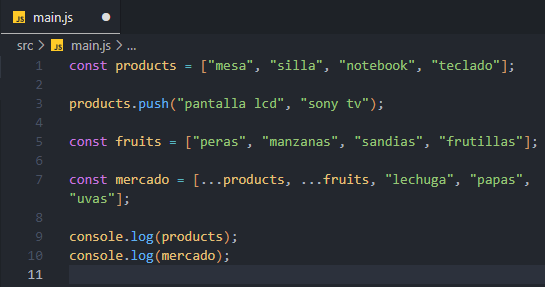
Si alteramos el orden de los arreglos que se unen, el orden de los elementos en el nuevo arreglo también cambiará.

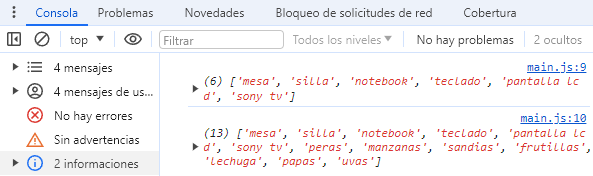




Añadir nuevos elementos al arreglo resultante

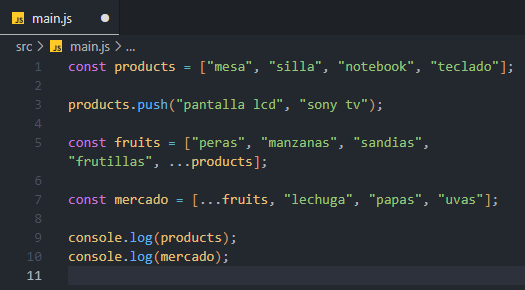
También es posible agregar nuevos elementos al arreglo resultante de la unión de otros arreglos.

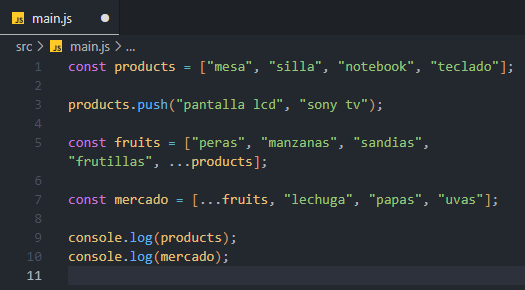


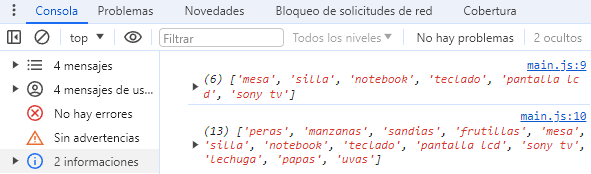


Combinar arreglos expandidos

El operador spread también permite combinar un arreglo que ya incluye otro arreglo expandido con más elementos.



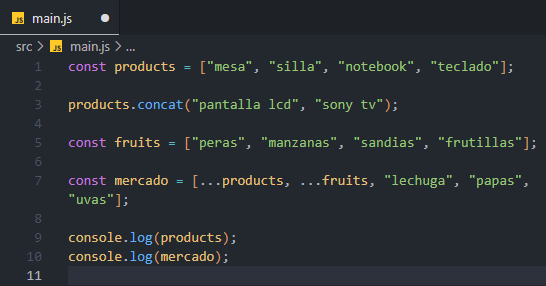


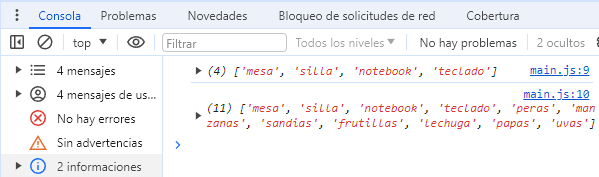


En el ejemplo mostrado, el arreglo **products** se mantiene inmutable, mientras que el arreglo **fruits** se modifica al incluir el operador spread. Al usar el operador spread, se crea una nueva instancia del arreglo.

El método concat en arreglos

También se puede utilizar el método **concat**, que es similar al operador spread, pero es más limpio y ordenado. A diferencia de **push**, **concat** puede aceptar múltiples ítems y devuelve una nueva instancia del arreglo sin modificar el original.

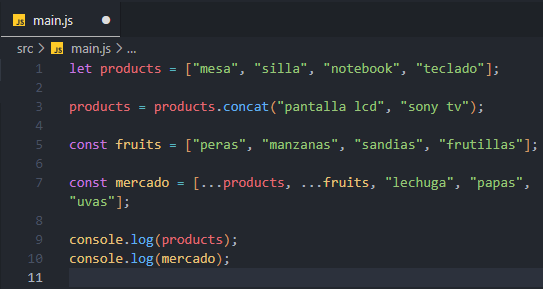


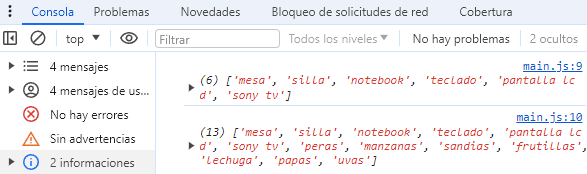


En el ejemplo mostrado, el resultado del método **concat** no se está mostrando directamente en la consola, porque se está imprimiendo una constante, pero los elementos agregados se aplican correctamente al arreglo original.

Definir el arreglo con una variable

Para visualizar los nuevos elementos agregados al arreglo **products** con el método **concat**, se tiene que definir el arreglo con una variable, porque una variable permite modificar su contenido.

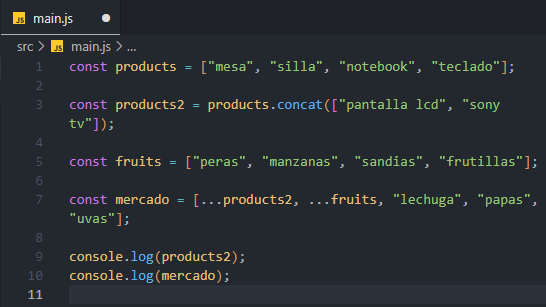


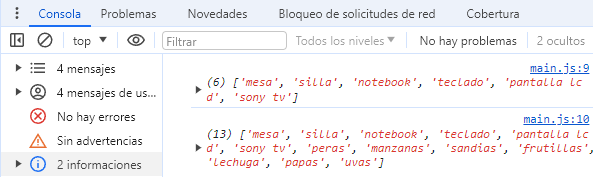


En el ejemplo mostrado, se imprimen los 2 elementos que faltaban, porque se está imprimiendo el valor de la variable **products** luego de aplicar el método **concat**.

Definir el arreglo resultante con una constante

En este caso se está definiendo ambos arreglos (**products** y **products2**) con una constante cada uno. Además, el método **concat** también puede admitir un arreglo definido entre corchetes. Ten en cuenta que en el arreglo resultante se está clonando el arreglo **products2**.

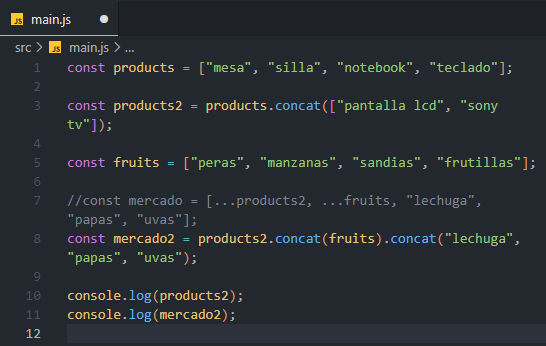


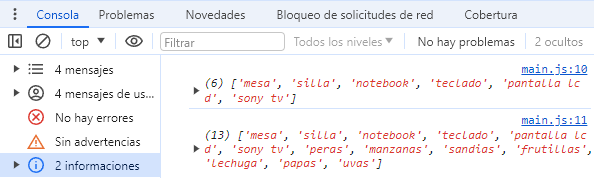


Ten en cuenta que se está imprimiendo en la consola la constante **products2**, el cual concatena el arreglo definido en **products** junto con los elementos: **“pantalla lcd”** y **“sony tv”**.

Concatenar arreglos con el método concat

Como ya se había mencionado antes, el método **concat** sirve para clonar y unir o concatenar varios arreglos. Sin embargo, en el desarrollo moderno y en bibliotecas como React, en la mayoría de casos se prefiere el uso del operador spread en lugar de **concat**.





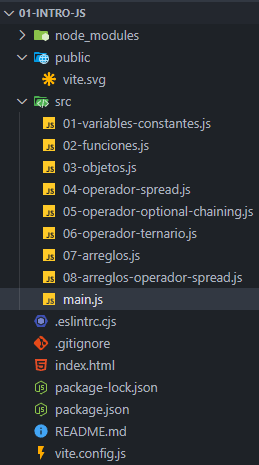
Antes de pasar al siguiente tema

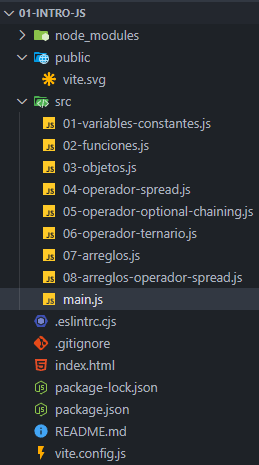
Realiza estas acciones en Visual Studio Code:

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 **08-arreglos-operador-spread.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.