Funciones de flecha o arrow function

En esta sección, vamos a identificar diferentes aspectos relacionados con las funciones anónimas y las funciones de flecha en JavaScript, también veremos como simplificar una función.

Función anónima

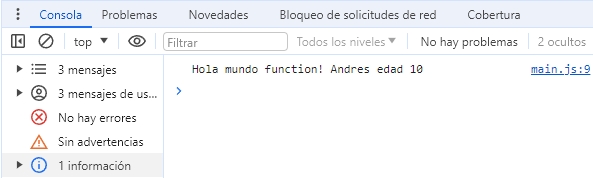
Una función anónima es una función sin nombre y se define generalmente en una constante utilizando la palabra clave **const**. Esto se debe a que, una vez definida, no se puede reasignar la función a otra variable.

Observe el siguiente ejemplo para definir una función anónima con parámetros:

const sayHello = function (name = "Pepe", age = 0) {
  const greeting = `Hola mundo function! ${name} edad ${age}`;

  // console.log("Hola mundo function!");
  return greeting;
};

const result = sayHello("Andres", 10);
console.log(result);



Función de flecha

La función de flecha, también conocida como expresión lambda, es una forma más concisa de definir funciones en JavaScript. Su principal ventaja es la simplificación del código.

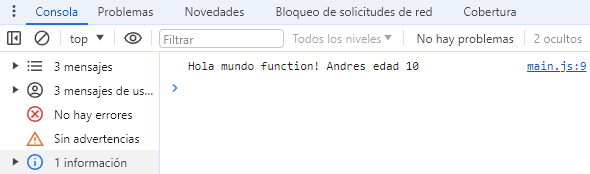
Se define utilizando una función anónima, seguidamente se elimina la palabra clave **function** y se coloca un operador flecha (**=>**) entre los paréntesis y las llaves de la función.

Observe el siguiente ejemplo para definir una función de flecha con parámetros:

const sayHello = (name = "Pepe", age = 0) => {
  const greeting = `Hola mundo function! ${name} edad ${age}`;

  // console.log("Hola mundo function!");
  return greeting;
};

const result = sayHello("Andres", 10);
console.log(result);



Simplificar la función

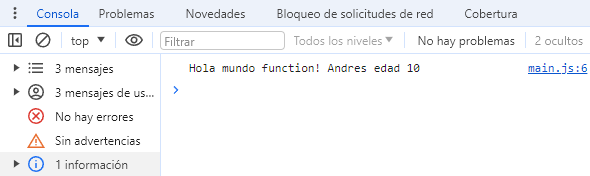
Si una función devuelve directamente un valor sin realizar otro procesamiento, se puede eliminar la definición de la constante contenida en el cuerpo de la función.

const sayHello = (name = "Pepe", age = 0) => {
  return `Hola mundo function! ${name} edad ${age}`;
};

const result = sayHello("Andres", 10);
console.log(result);

const sayHello = (name = "Pepe", age = 0) => {
  return `Hola mundo function! ${name} edad ${age}`;
};

const result = sayHello("Andres", 10);
console.log(result);

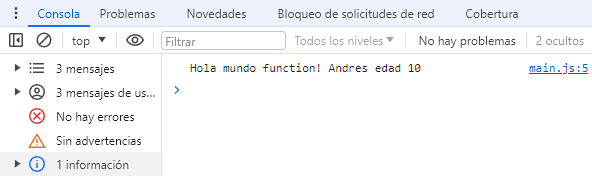


Omitir las llaves y el return

Cuando una función de flecha tiene una única línea que retorna un valor, es posible omitir las llaves y la palabra **return**, de lo contrario se tendría que utilizar las llaves y el **return** (si devuelve algo).

const sayHello = (name = "Pepe", age = 0) =>
  `Hola mundo function! ${name} edad ${age}`;

const result = sayHello("Andres", 10);
console.log(result);

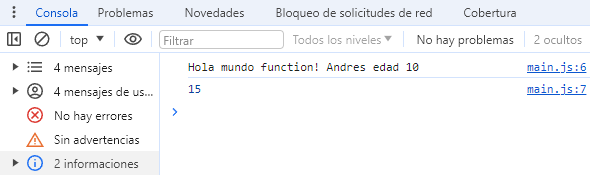


Operaciones dentro de funciones

También es posible realizar operaciones dentro de las funciones de flecha de manera más concisa, especialmente cuando la función realiza una única operación con los parámetros.

const sayHello = (name = "Pepe", age = 0) =>
  `Hola mundo function! ${name} edad ${age}`;
const add = (a = 0, b = 0) => a + b;

const result = sayHello("Andres", 10);
console.log(result);
console.log(add(10, 5));

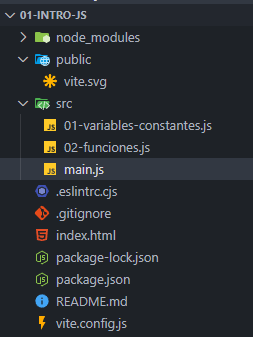
Antes de pasar al siguiente tema

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 **02-funciones.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.