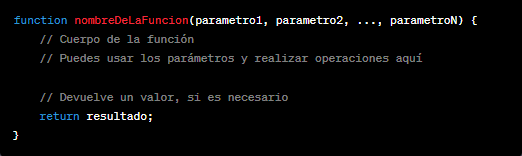
Funciones

En JavaScript, **las funciones son bloques de código reutilizables que pueden ser llamados con un nombre específico**. Pueden aceptar datos, conocidos como argumentos, y pueden devolver un valor como resultado. Las funciones en JavaScript son objetos de primera clase, lo que significa que pueden ser pasadas como argumentos a otras funciones, devueltas como valores desde otras funciones y asignadas a variables.

Aquí hay una estructura básica de una función en JavaScript:



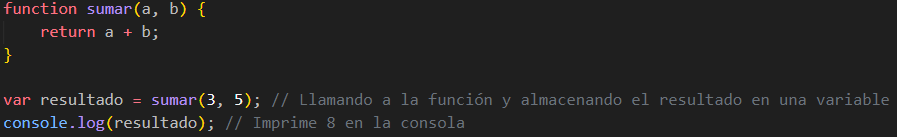
**nombreDeLaFuncion:** Este es el nombre que le das a tu función. Debe seguir las reglas para nombrar variables en JavaScript.

**parametro1, parametro2, ..., parametroN:** Estos son los valores que la función espera recibir cuando es llamada. Los parámetros son opcionales; una función puede no tener ningún parámetro o puede tener varios.

**Cuerpo de la función:** Aquí es donde escribes las instrucciones que quieres que la función ejecute cuando es llamada.

**return:** La palabra clave return **se utiliza para devolver un valor desde la función**. Una función puede o no tener una declaración de retorno. Si una función no tiene un return o tiene un return sin un valor, la función devuelve undefined por defecto.

Aquí hay un ejemplo de una función simple que suma dos números:



En este ejemplo, la función **sumar** **toma dos parámetros**, **a** y **b**, y **devuelve la suma de estos dos valores**. Cuando llamamos a la función con **sumar(3, 5)**, devuelve **8**, que luego se almacena en la variable resultado y se imprime en la consola.

Además de la forma tradicional de declarar funciones, a partir de ECMAScript 6 (ES6), se introdujo una nueva sintaxis llamada **funciones de flecha** (arrow functions), que proporcionan una forma más concisa de escribir funciones en JavaScript. Aquí hay un ejemplo de una función de flecha que hace lo mismo que la función sumar del ejemplo anterior:

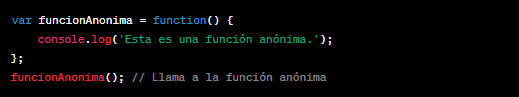


Ambos ejemplos hacen lo mismo: toman dos números como argumentos y devuelven su suma. La función de flecha es más corta y a menudo se utiliza en casos simples como este.

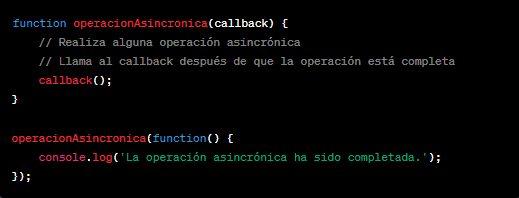
Tipos de Funciones

hay varios tipos de funciones en JavaScript, cada uno con sus características y usos específicos. Además de las funciones estándar y las funciones de flecha, hay otras categorías de funciones que son importantes conocer:

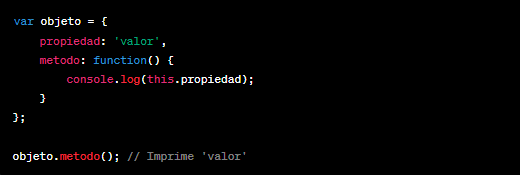
❶ Funciones Anónimas: Las funciones anónimas son funciones sin nombre. Pueden ser asignadas a variables y también pueden ser pasadas como argumentos a otras funciones. Son útiles en situaciones donde necesitas una función temporalmente y no necesitas reutilizarla. Aquí hay un ejemplo:



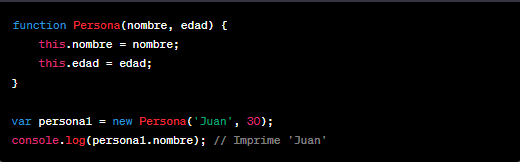
❷ Funciones de Callback: Las funciones de callback son funciones que se pasan como argumentos a otras funciones y se ejecutan después de que cierta operación ha sido completada. Son comúnmente utilizadas en operaciones asincrónicas, como las operaciones de lectura de archivos o las solicitudes a servidores.



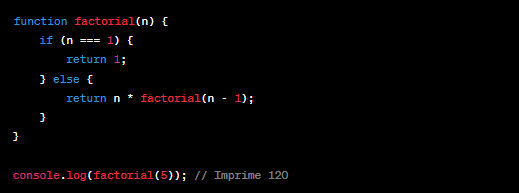
❸ Funciones de Métodos: Las funciones pueden ser definidas como métodos de objetos. Cuando una función es un método de un objeto, **this** se refiere al objeto que posee el método. Esto es fundamental en la programación orientada a objetos en JavaScript.



❹ Funciones Constructoras: Las funciones constructoras son utilizadas para crear objetos. Se utilizan junto con la palabra clave **new**. Los objetos creados con funciones constructoras pueden tener propiedades y métodos.



❺ Funciones Recursivas: Una función recursiva es una función que se llama a sí misma dentro de su definición. Esto es útil para resolver problemas que se pueden dividir en subproblemas del mismo tipo.



***Estos son solo algunos ejemplos de los tipos de funciones en JavaScript. Cada tipo tiene su propio propósito y uso específico en diferentes situaciones.***