### Funcionamiento de Async Await

async y await son dos características importantes de JavaScript que facilitan el manejo de operaciones asíncronas. Aquí te explico cómo funcionan de manera simple:

**Async**

async es una palabra reservada que transforma una función normal en una función asíncrona. Una función asíncrona siempre devuelve una promesa.

***Ejemplo:***

**async function miFuncionAsincrona**() {

// Código asíncrono aquí

}

### Await

await se utiliza dentro de funciones asíncronas para esperar a que una promesa se resuelva antes de continuar con el código.

***Ejemplo:***

**async function miFuncionAsincrona**() {

try {

const resultado = await **algunaOperacionAsincrona**();

console.log(resultado);

} catch (error) {

console.error("Ocurrió un error:", error);

}

}

**Funcionamiento básico**

1. Cuando se llama una función asíncrona (miFuncionAsincrona()), JavaScript inmediatamente devuelve una promesa y continúa ejecutando el código siguiente.
2. Dentro de la función asíncrona, await detiene la ejecución del código hasta que la promesa asociada se resuelva.
3. Una vez que la promesa se resuelve, el valor resultante se asigna a la variable antes de la palabra reservada await.
4. Si hay un error en la promesa, await lo captura y lanza una excepción que puede ser maneada con un bloque catch.

Ventajas

* Simplifica el manejo de operaciones asíncronas.
* Mejora la legibilidad del código comparado con callbacks o promise chaining.
* Facilita el manejo de errores con try/catch.

***Ejemplo práctico simple***

**async function obtenerDatos**() {

const respuesta = await fetch('https://api.example.com/datos');

const datos = await respuesta.json();

console.log(datos);

}

obtenerDatos().then(() => {

console.log("Operación terminada");

});