



# Async/await

## In Javascript

# What is Asynchronous Programming?

**Asynchronous programming ensures that the execution does not stop if a function is performing other operations.**

**Instead, the execution continues normally until the function is called back again. To facilitate this, concepts like Callbacks, Promises, and Async/await are used.**



**Async/Await is used to work with promises in asynchronous functions. It is basically syntactic sugar for promises. It is just a wrapper to restyle code and make promises easier to read and use. It makes asynchronous code look more like synchronous/procedural code, which is easier to understand.**

```
const helperPromise = function () {
  const promise = new Promise(function (resolve, reject) {
    const x = "geeksforgeeks";
    const y = "geeksforgeeks";
    if (x === y) {
      resolve("Strings are same");
    } else {
      reject("Strings are not same");
    }
  });

  return promise;
};

async function demoPromise() {
  try {
    let message = await helperPromise();
    console.log(message);
  } catch (error) {
    console.log("Error: " + error);
  }
}

demoPromise();
```

# Error Handling in Async/Await:

For a successfully resolved promise, we use `try` and for rejected promise, we use `catch`. To run a code after the promise has been handled using `try` or `catch`, we can `.finally()` method. The code inside `.finally()` method runs once regardless of the state of the promise.

