

CALLBACK

VS

PROMISE

VS

ASYNC AWAIT



codeinjs



**Callback, promise, and
async await** are used to
achieve **asynchronous**
programming in
JavaScript.

Let's check each of them in next slides



CALLBACK

The callback is the **oldest** way to work with asynchronous operations in JavaScript.

The callback is **a function passed to another function** as an argument.



CALLBACK

The callback suffers from the **callback hell** problem when working with **dependent** asynchronous calls.

It offers **poor** error handling.



PROMISE

The promise is the **new way** of dealing with asynchronous operations and it is introduced in **ES6**.

It overcomes the problems in callback.



PROMISE

The promise offers different **set of methods** to work with **dependent** and **parallel** asynchronous calls.

It provides **better** error handling than the callback.



ASYNC AWAIT

Async and await keywords are introduced in **ES8** and they are always used together.

The **async** keyword is used in front of a **function**. While **await** is used in front of the **asynchronous call**.



ASYNC AWAIT

The `async await` makes code more **readable** to developers.

It provides **better** error handling with traditional **try catch** blocks.

