What this chapter is about?

async await >> promise chains >> callback hell

Sync in JS

Synchronous

Synchronous means the code runs in a particular sequence of instructions given in the program.
Each instruction waits for the previous instruction to complete its execution.

Asynchronous

Due to synchronous programming, sometimes imp instructions get
blocked due to some previous instructions, which causes a delay in the UI.
Asynchronous code execution allows to execute next instructions
immediately and doesn't block the flow.

Callbacks

A callback is a function passed as an argument to another function.

Callback Hell

Callback Hell : Nested callbacks stacked below one another forming a pyramid structure.

(Pyramid of Doom)

This style of programming becomes difficult to understand & manage.

Promises

Promise is for “eventual” completion of task. It is an object in JS.

It is a solution to callback hell.

```
let promise = new Promise( (resolve, reject) => { .... } )
```

Function with 2 handlers

*resolve & reject are callbacks provided by JS

Promises

A JavaScript Promise object can be:

- Pending : the result is undefined
- Resolved : the result is a value (fulfilled)
- Rejected : the result is an error object

resolve(result)

reject(error)

*Promise has state (pending, fulfilled) & some result (result for resolve & error for reject).

Promises

.then( ) & .catch( )

```
promise.then( ( res ) => { .... } )
```

```
promise.catch( ( err ) ) => { .... } )
```

Async-Await

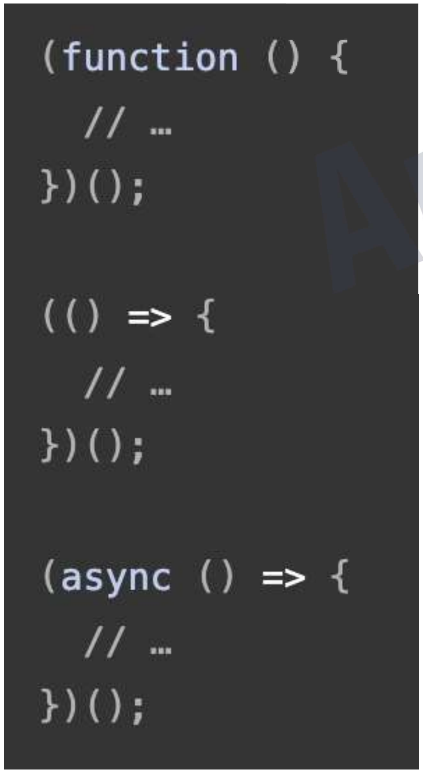
async function always returns a promise.

```
async function myFunc( ) { .... }
```

await pauses the execution of its surrounding async function until the promise is settled.

IIFE : Immediately Invoked Function Expression

IIFE is a function that is called immediately as soon as it is defined.



```
(function () {
  // …
})();

(() => {
  // …
})();

(async () => {
  // …
})();
```