Async operations in javascript

We had a look at four different approaches:

-**Callbacks** with the danger of entering callback hell

-**Promises** to escape callback hell

-**Observables** to handle streams of data and apply operator

-**async/ await** to write "synchronous" code with promises

Which approach should you use?

Callbacks

Use **Callbacks** if you got no other choice or only handle one async operation.

The code will then still be perfectly manageable and understandable.

there exist better alternatives in many cases for the Callback functions.

So just avoid using them - because once a case has multiple chained (or dependent) asynchronous operations. You quickly enter callback hell when trying to use callbacks in such a situation.

Promise

Promises are a great tool to handle your operations in a structured and predictable way.

They are native object in es6.

RxJS - Observable

There's a strong argument to be made for observables once you handle data streams, i.e. operations where you're not just getting back one single value.

Promises can't easily handle such situations.

observable are native as eventEmitter in node.js

async/ await

async/ await is a great tool for cases where you don't really want or need to use observables but still want to use promises.

Async/ await gives you a powerful way for working with promises. (It doesn't work with observables though).

You can write "synchronous" code with async/ await and handle your promise chains even easier.

new feature added by ES8