JS - Asynchronous Programming

* Promises

Promise is a technique in JavaScript for handling asynchronous execution.

A promise is a special JavaScript object that links the "producing code" and the "consuming code" together.

The "producing code" takes whatever time it needs to produce the promised result, and the "promise" makes that result available to all the subscribed code when it's ready.

Syntax :-

```
                                      executor
let promise = new Promise(function(resolve, reject){
    //executor (the producing code, "singer")
});
```

When ┌`new Promise`┐ is created, the executor runs automatically.

Its arguments resolve and reject are callbacks provided by JavaScript itself.

A promise object can be in one of three states:-

i) Pending
ii) Fulfilled (with a value)
iii) Rejected (with a reason)

When the executor obtains the result, be it soon or late, doesn't matter, it should call one of these callbacks:-

- resolve(value) — if the job is finished successfully, with result value

- reject(error) — if an error has occurred, with the error object.

The promise object returned by the ┌`new Promise`┐ constructor has these internal properties:-

- state — initially "pending", then changes to either "fulfilled" when resolve is called or "rejected" when reject is called.

- result — initially undefined, then changes to value when resolve(value) called or error when reject(error) is called.
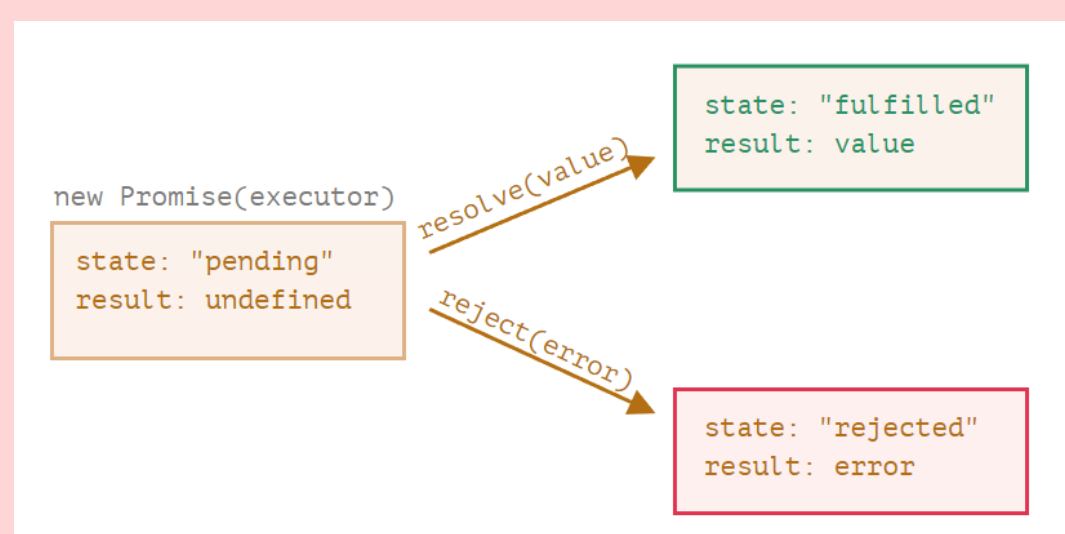
* Async/await

Introduced in ES 2017, it is a syntactic sugar for promises.

It is a special syntax to work with promises in a more comfortable fashion.

↳ Async functions

The async keyword can be placed before a function, like this:-

```
async function f(){
    return 1;        → returns a resolved promise with
}                       the result [1].
```

The word "async" before a function means one simple thing: a function that always returns a promise. Other values are wrapped in a resolved promise automatically.

↳ Await

```
let value = await promise;
```

Await works only inside async functions.

The keyword await makes JavaScript wait until that promise settles and return its result.