

Promises

- * Promises Enhance Readability
- * Promises Can solve the problem of Inversion of Control
- * In J.S, Promises are special type of objects that get returned immediately when we call them.

Inversion of Control

function fun(x, cb) {
 for (let i=0; i<x; i++) {
 //
 }
 ① cb() → here we are calling the Call back.
}

① Normal function in which we have called callback

// Now Let's call the function fun

fun(10, function exec() {
 console.log("done")
});

② ←

Note → The PROBLEM of inversion of control is we don't know whether the callback, cb() ① is called or no by the function fun., and also we don't know whether it is calling the callback function two times or three times or it is not called so we don't know how the function fun is handling ~~we don't know~~

Suppose the function `fun()` (A) is not implemented by me it is implemented by Someone else, so the mistake we are doing here is we are giving our `exec()` (B) function control to Someone else and how they are handling it we don't know, and this is the problem of IOC.

_____ x _____ x _____> x _____>

* Promises acts as a placeholder for the data we hope to get back somehow in future.

Suppose for example, we are downloading some file

`x = fetch("http://www.xyz.com")`

Suppose we are downloading this
assume `fetch` is written using Promises. then it will immediately return a Promise object which will act as a **placeholder**.

* In these Promise objects we can attach the functionality we want to execute once the future task is done.

Suppose for example, we click on a button where we downloaded the movies and after downloading the movies we will get the names of movie downloaded, so once future task is done, promises will automatically execute the attached functionality.