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

(A) ← Normal function in which we have called callback

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

// Now Let's Call the function fun

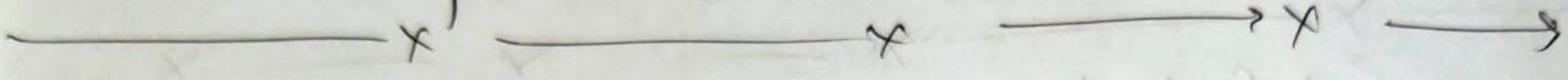
(B) ←

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

Note → The PROBLEM of inversion of Control is we don't Whether the Callback, cb() (i) is Called or no by the function fun., cons also we don't know whether it is calling the Callback function two times or three times or if it is not called so we don't know how the function fun in handelling we don't know

Scanned by TapScanner

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 handelling it we don't know, and this is the problem of IOC.



* 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 objection 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 downloded 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.

Scanned by TapScanner