Lecture 25 [JS-DOM-IV] Priyanshu Patel

21/02/23

* Asynchronous JS

→ Javascript program can start long-running tasks, and continue running other tasks in paralell.

→ Asynchronous programming is solved using promises instead.

* Promise

→ JS Promise object can be

(i) Pending (ii) Fulfilled (success full execution) (iii) Rejected

→ Promise object support two properties →

(i) state (ii) result

→ Syntax –

let myPromise = new Promise(function(resolve, reject) { (contain call back func)

(parameter must)

– –

}

Promise —state→ fulfilled

Promise —state→ Rejected

Promise —apply method→

(I) then():- Return value by promise used by it. [after promise execute something]

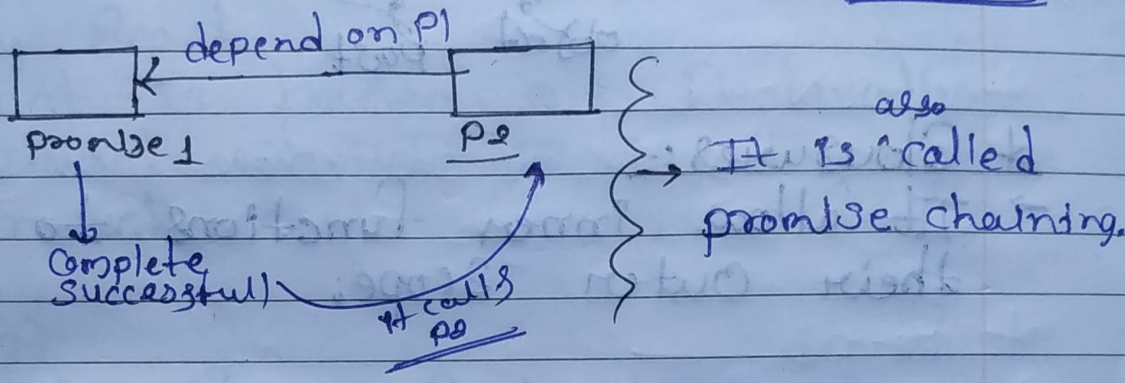
(II) Catch():- error comes in promise. use by catch().

Date: !!
P. No:

→ Promise runs in background.

→ On the successfully completion of promise, we have to do something then use then method. (then())

Ex



* Async - await

→ Special syntax used to work with promises.

→ Async always return promise.
→ Example of Async function
↳ network call.

→ await is used for wait for promise.
→ await keyword can only be used inside an async function

* Fetch - API

→ fetch() method starts the process of fetching a resource from a server.

→ fetch() method return a Promise that resolve to a Response Object.

for security purpose

Date: 11

P. No:

Brijansh (K) Patil

→ fetch ('url', key);

object

→ fetch('url';) → get

→ fectc ('url', options)

object post

* Clousures:-

→ It allow inner functions to access their outer scope.