We will start @ 2:10

↳ console.log

↳ process.stdout.write

4 in event queue → we receive call backs

Disadvantages of Callbacks

1) Callback hell

2) Inversion of control → If there is some part of code whose control we are passing to someone else, and we don't know how that part will be executed this problem is called inversion of control.

Promises

↳ Promises are special JS objects that are also considered readability enhancers. They get immediately returned from a function setup to return a promise.

↳ They act as placeholders for the data we hope to get back from some future task.

→ We also attach the functionality we want to defer until the feature task is done. And promise automatically handle execution of this functionality.

→ So promises do two things, one inside JS & one outside JS.

1) It signs up the process required to run in the runtime & gives a placeholder in JS, which has a value property.

(1) How to create a promise??

(2) How to Consume a promise??

```
6   function fetchCustom(url, fn) { // we are mimicing the function
7       // download content of the url
8       // this downloading can take sometime,
9       // we will download the content from url, and then whatver is t
10      console.log("Starting downlaoding from", url);
11      setTimeout(function process() {
12          console.log("Download completed");
13          let response = "Dummy data";
14          fn(response);
15      }, 3000);
16  }
```

fetchCustom("www——")

```
18  function writeFile(data, fn) {
19      // this function writes data in a new file
20      console.log("Started writing data", data);
21      setTimeout(function process() {
22          console.log("Writing completed");
23          let filename = "output.text";
24          fn(filename);
25      }, 4000);
26  }
```

```
function uploadFile(filename, newurl, fn) {
    console.log("Upload started");
    setTimeout(function process() {
        console.log("File", filename,  "uploaded successfully on", newurl);
        let uploadResponse = "SUCCESS";
        fn(uploadResponse);
    }, 2000);
}
```

```
38  fetchCustom("www.google.com", function downloadCallback(response) {
39      console.log("Downloaded response is", response);
40      writeFile(response, function writeCallback(filenameResponse) {
41          console.log("new file written is", filenameResponse);
42          uploadFile(filenameResponse, "www.drive.google.com", function uploadCallback(uploadResponse) {
43              console.log("Successfully uploaded", uploadResponse);
44          })
45      })
46  });
```