LECTURE-24
WEB DEVELOPMENT
(CLASS 5)

26th Sept 2021
(Sunday)

JAVASCRIPT

To Do List:

- first experience with timer
- first experience with downloading data from web
- first experience with writing data in excel file
- first experience at writing pdfs

[हर नये folder mai minimist install करना होगा]

① Create a big file

filename: (CreateBigfile.js) [running statement: // node CreateBigdata.js --dest = Big.data.]

```
let minimist = require("minimist");
let fs = require("fs");
let args = minimist(process.argv);

let arr = [];
```

```
for(let i=0; i< 50000000 ; i++){
    arr.push(i);
}

let str= arr.join("\n");
fs.WritefileSync(args.dest, str, "Utf-8");
fs.appendfileSync(args.dest, str, "Utf-8");
fs.appendfileSync(args.dest, str, "Utf-8");
fs.appendfileSync(args.dest, str, "Utf-8");
```

//around 1.7 GB file is Created

[1.7 GB की बड़ी file बना लीजिए] → ताकी Callbacks को अच्छी तरह समझ पाऐ |

Terminal:

```
node CreateBigfile.js --dest=big.data
```

Note:

Minimist को Globally install करने के लिए Type करो:

```
npm install minimist -g
```

-g → global installation.

⊙ first experience with lack of callback

[filename: firstlack of callback.js]

Previously, done in Lecture-23 (25th Sept 2021)

★ जो कुछ SIR ने नया बताया होगा वो यहा Mention होगा ★

task1 → to read the big.data file
task 2 → to print the prime numbers.

→ task 2 will be done after task 1, which is not good.

TERMINAL: node firstLack of Callback.js --source=abig.data --n=7000

```
let minimist = require ("minimist");
let fs = require ("fs");

let args = minimist (process.argv);
// task 1 begins
let t1 = Date.now()
console.log("Task1 started at" + t1 % 100000);
fs.readFileSync (args.source         );
let t2 = Date.now ();
console.log ("Task1 finished at" + t2 % 100000);
console.log(t2 - t1);
// task1 ends.
```

→ // big file hai uske liye utf-8 nahi Lagana.

[around 4 sec lagenge task1 ko complete hone mai]

```
// task 2 area starts(begins)
{ code for
  prime numbers }

console.log ( t4 - t3);
// task 2 area ends

console.log (t4 - t1);
```

→ total duration of both the tasks.

[Task2 जो है, वो Task1 के बाद ही Execute ho रहा है। बहुत ज्यादा समय लगता है।]