Scripts

To avoid running files seperately and mismatch, like

node ~~npm~~ app.js

we write

```
"scripts" : {
    "start" : "node app.js"
```

and in cmd

npm start

Node Event Emitter

On the backend, nodejs offers us to build a interactive system using the events module. awhich consists of Event Emitter class.

Initialize =>

```
const EventEmitter = require('events');
const event = new EventEmitter();
event.on('start', () => { c.l('started') });
event.emit('start');
```

on → used to add a callback funct when a event is triggered.

emit → to trigger an event

__/__/____

Node.js

free & open source server.

→ A platform which allows us to run JS on a computer / server
→ read, delete and update files.
→ easily communicate with a database

Why node.js ?

→ It uses JS
→ Very fast (runs on V8 engine & uses non-blocking code)
→ huge ecosystem of open source packages (npm)
→ great for real time services (like chats)

(browsers)

V8 engine is written in C++ by google and it compiles JS into machine code.

Node.js is also written in C++. V8 engine is wrapped inside node.js as well as in browsers. which helps JS to run on computers outside browsers too.

JS

Arrow functions

Eg 1> `const greet = (n) =>`

```
{
    console.log(`hello ${n}`);
or  console.log("hello, " + n");
}

greet("Yukta");
greet("Jai");
```

2> `const add = (x, y) => x + y;`

`console.log( add(1, 2) );`

Global Object

Objects that are available in all modules.
Built-in objects part of JS and can be used directly without importing.

Eg > buffer, console, process, global, setImmediate(), setInterval(), clearTimeout(), URL, etc.

To get directory name : `console.log(__dirname);`
" " filename : `console.log(__filename);`

(Import & Export)

__/__/____

Modules & Require

Import using Require

```
const xyz = require
        ("./people");

console.log(xyz);

// prints
   empty
   object {}.
```

```
const people = [

    "yukta",
    "jai",
    "tripti",

]

console.log( people);
        // prints the
           object.
```

Exporting using modules

```
module.exports = people;
```

Then,

```
console.log (xyz);   // will print the object.
```

Exporting multiple objects

```
module.exports = {
        people, ages
};
```

To import seperately,

```
console.log(xyz.people, xyz.ages);
```

Importing using same name

```
const {people} = require("./people");
console.log(people);
```

~~console~~ const {people, ages}

```
c.l(people, ages);
```

Function expressions

The function keyword is used to define a function inside an expression.

Eg:

```
const area = function(length, width)
{
    return length * width;
}

console.log(area(3,4));
```

// without expression

```
function area(a,b)
{
    return a*b;
}
console.log(area(3,4));
```

File System (create, read, delete)

core node
This is a module which allows us to read, create, delete files on our computer using Javascript.

To read file,

```
const fs = require("fs");

fs.readFile("./doc/text.txt", (err, data)
=> { if (err) {
        console.log(err);
   }
   console.log(data.toString());
});
```

pathname

callback function to print data

prints the data of this file

```
console.log("second line");
```

→ this line will be printed first.

readFile → it is asynchronous and takes some time to process the code.

__/__/____

Writing files and creating

```
fs.writeFile('./doc/text.txt', "Writing file", () => {});
fs.writeFile("./doc/text2.txt", "Creating file", () => {});
```

Create folder/directories

```
fs.mkdir("./doc2", (err) => { });

if (!fs.existsSync("./doc2"))
{
    fs.mk
}
else
{
    fs.rmdir("./doc2");
}
```

remove folder

delete files

```
fs.unlink("./doc/text2.txt", () => {});
```