8/12/25

Modules.export & require

always available
require(". /xyz.js") ⇒ ye phele run hoga
↳ to add another code file

// code ⇒ ye uske baad

// module
// modules protects their variables and functions from leaking. (by default)

⇒ To do this:
→ module.exports = calculateSum;

↳ const calculateSum = require(".\sum.js");

more than one:
~~module.exports~~ module.exports = {
x: x,
calculateSum: calculateSum,
};

new way
{ x,
calculateSum,
};

const obj = require(".\sum.js");
⇓
obj.calculateSum(a, b);

bethr
↳ const { x, calculateSum } = require(".\sum.js")
↳ destructuring.

To use package.json

↓

{"type": "module"}

↑

__/__/____

CommonJs modules (CJS)	ES Modules (mjs)
→ module.exports	→ import
⇒ require	export
→ By default used in Node	→ by default used in React, angular
→ Older way	→ Newer way
→ synchronous	→ Async option is available
→ Non-strict mode	→ strict mode

• "ES modules"

⇒ To export:

export function .. { ... }

To import

import {calculateSum} from "./sum.js"

To make file as module

/calculate → (folder)

multiply.js

index.js

const {multiply} = require("./...

cons { .. } = require("./...")

module.exports = { multiply, ... }

const {calculateSum} = require("./calculate/...")

__/__/__

strict mode :

ismai directly hum "z=10" ye nhi kar sakte hai hume "var z=10" hi karna hoga.

→ commonjs module mai directly kar sakte hai.

console.log (module.exports) ;

⇒ {} ⇒ empty object

↓

~~module.x~~ = module.exports.x = x ;

module.exports.calculateSum = calculateSum ;