Namaste Node.js

→ Every node.js application has a single entry point.

→ So, if some code (file or folders or modules) required in some other module how do we achieve this ?

↓

it can be solved by using " require (" filename"); " keyword wherever the filename need to be use.

(a global function) used for using one module into another.

↳ if the other module contained some function and we need to use it (we can't use the variables, methods, and functions of one module into another by only using require (" filename"))
↳ (more correctly filepath)

⇒ Modules protects their variables and functions from leaking.
↓ what's the solution, if we want to use it

→ we need to explicitly export those

variable and methods.

e.g. first module	using Module

function calculateSum(a,b){
const sum = a+b;
console.log (sum);
}
	const calculateSum =
	require (" ./first-module.js);
	// importing the
	exported function

module.exports = calculateSum	calculateSum(20,30);

- These can be written in another ways as well (destructing, other methods of exporting and importing) (React ways)

→ These patterns we have used is known as CommonJs modules.

CommonJs Modules (.cjs)	Es modules (Esm) (.mjs)
→ module. exports = { n, calculateSum } → require (" filepath") • by default used in Node JI	→ import export a. package.json is needed for type change module • by default used in React , Angular ,

→ older way	→ newer way
	(may be in coming years this will be the only standard)
→ works in synchronous way	→ works in Asynchronous way (better way)
→ code works in non strict mode e.g. Z = "Hello" ✓ works ✓	→ code works in strict mode. e.g., Z = "Hello" ✗ ✗ Z is not defined (Error)

Q) What is (module) exports ?
A) It is an empty object { }
so, we can also write
e.g., module. exports. calculatesum = calculatesum

Separation of concerns will need to make different modules for different functionalities which makes our app much larger but easier. To understand and navigate.

Q. what do you mean by module?

A) → Module is a collection of code which is private and independent to itself.

→ More like its local.