Episode-05 → Diving Into NodeJS GitHub Repo

Module? → It is a collection of code which is private in itself and it can decide whether which functions and variable should be allowed to Access by any other module which imports it.

#How require( ) Works?

Whenever we require a module, the nodeJS takes the code from that file and wraps it into a function and then executes it.

- This is the reason we cannot Access the function and variables as inside a function they became privately scoped that is can only be Accessed inside the function.
- All the code of module (which is imported using require) is run inside the function cause code is wrapped inside a function (IIFE).
- This function is an IIFE (Immediately invoked function expression)

IIFE → `( function (){`

`})();` → Immediately Invoked function expression

- Hence NodeJS wraps the code of imported module in a IIFE function and gives it to V8 engine
- IIFE keeps variables and functions safe/ private.

Query How are variables and functions private in different Module?

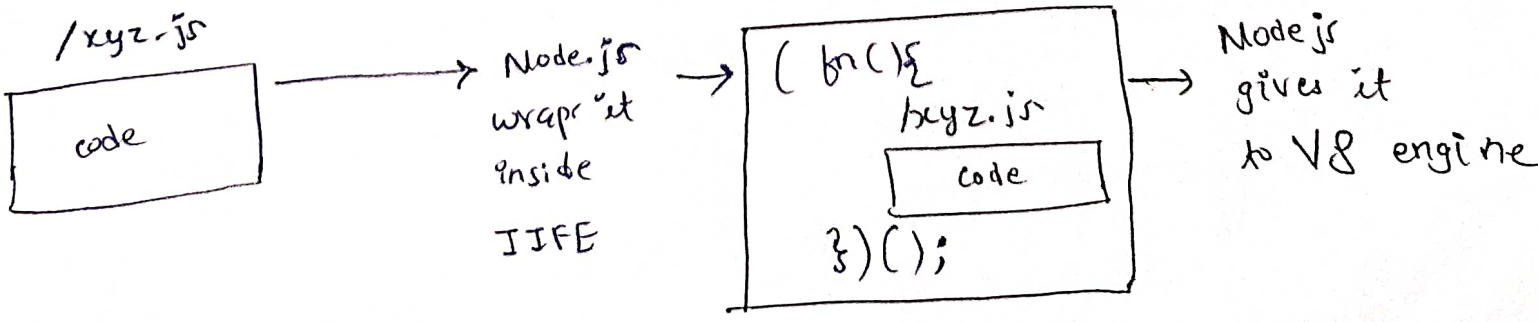
→ Because of IIFE and require statement

→ require statements wraps the code in IIFE.

Query from where does module.exports and require comes from?

→ Node gives these as parameters to the function in which the code of imported module is wrapped in.

→ NodeJS passes module.exports and require as a parameters to the IIFE.



#5 Step Mechanism of require( )

5 steps that happen to get the module and execute the code :→

① Resolving The Module : checks whether module is ./localpath , .json or node-module

② Loading The Module : file content is loaded according to file Type

③ Wraps inside an IIFE (Compile step)

④ Code Evaluation : module.exports is returned in this step.

⑤ Caching : Module is Cached, i.e. if the same module is required again, caching helps very much as it won't run this module again and again. Caching makes it efficient to run the code. resulting in faster execution.