Note :- When we call only require ('./path');
why we dont able to get the methods & variable from that path.

→ Bcoz, all the code of the module is wrapped inside a function. its their scope is only the function.

→ that function is called IIFE

IIFE → Immediately Invoked Function Expression.

So, IIFE is a function that got immediately invoked.

i.e.
```
(function () {

    })(); → created anonymous function & immediately called it.
```

So,
```
( function () {
        // all code of the module runs inside here.
    }) ();
```

→ IIFE should be wrapped inside ( ). so that it becomes expression.

→ why we need it ?

① It immediately invoke the code, & we need that immediate ~~invok~~ invocation during code loading

② It keeps variable & function safe.

Ⓠ How are variable & functions are private in different module?...

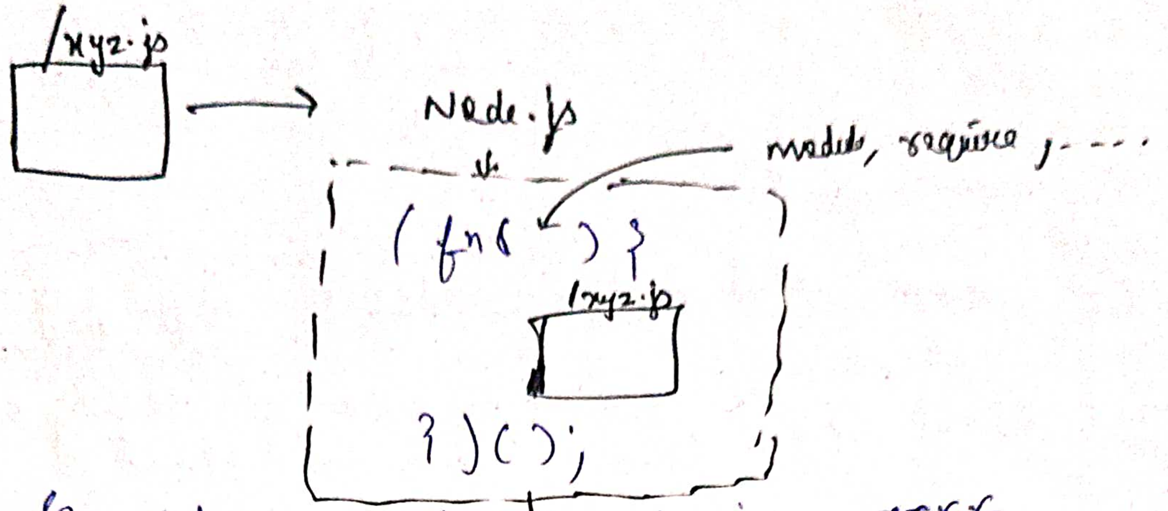
→ IIFE & require (statement)
↑
(wrapping code)

Ⓠ How do you get access to module.exports ?

→ Actually, Module is a parameter in ~~IIF~~ IIFE.
So, Node js is providing us access to module.
& it is passing module as parameter

Q) How are we getting access to require?

→ Node also provide require as parameter in IIFE



So, Node.js wrapped that code in a IIFE

↓

V8 engine

new V8 engine execute that code

Five steps ~~to~~ mechanism of require('./path').

1) Resolving the module.

- it checks ~~withe~~ whether the ~~path is~~ data is coming from,

→ ./localpath → node: module

→ ./json

then it resolves that module

2) Loading the Module :-

↳ file content is loaded according to file type

3) wraps inside IIFE (compile step)

4) Evaluation → module.exports happens/returned out to get data

5) Caching → so, next time all ~~w~~ loading not occur again.