5/08/24

Lec-5

Diving Deep into NodeJs Github Repo

What is IIFE

When we do require('./path').
All the code of this module is wrapped inside a function (IIFE)

IIFE → Immediately Invoked Function Expression.

```
(function () {
    // All Code of Module Runs inside this !
})()
```

↑ These Brackets are important

Before Execution All the requiries are Converted into IFFE or wrapped inside IFFE

It keeps Variables & Functions Private and Safe. The Code inside IIFE will not interfere with the Files Code.

Q How are Variables & functions Private in Different Module ?

lets suppose we have a file :

```
function x() {
    Const x = 10;
    function b() {
        Console.log('b');
    }
}
```

Console.log(a) ← this a is not accessible here.

Whenever you create a function it Creates its own space / scope.

Modules works the same way

Whenever you create a module all the Code that you write in a module is wrapped inside a function and then executed

This is why you cannot access the data of the Module without expo

* Whenever you create a module and Require it into your file what happens is nodejs takes the code from that file or module and wraps it into a function & then executes it

Why Cant we access variables/functions directly ?

* All the code of a module is wrapped inside a function, when you call Require

The function is a special function known as (IIFE)

IIFE → Immediately invoked Function Expression.

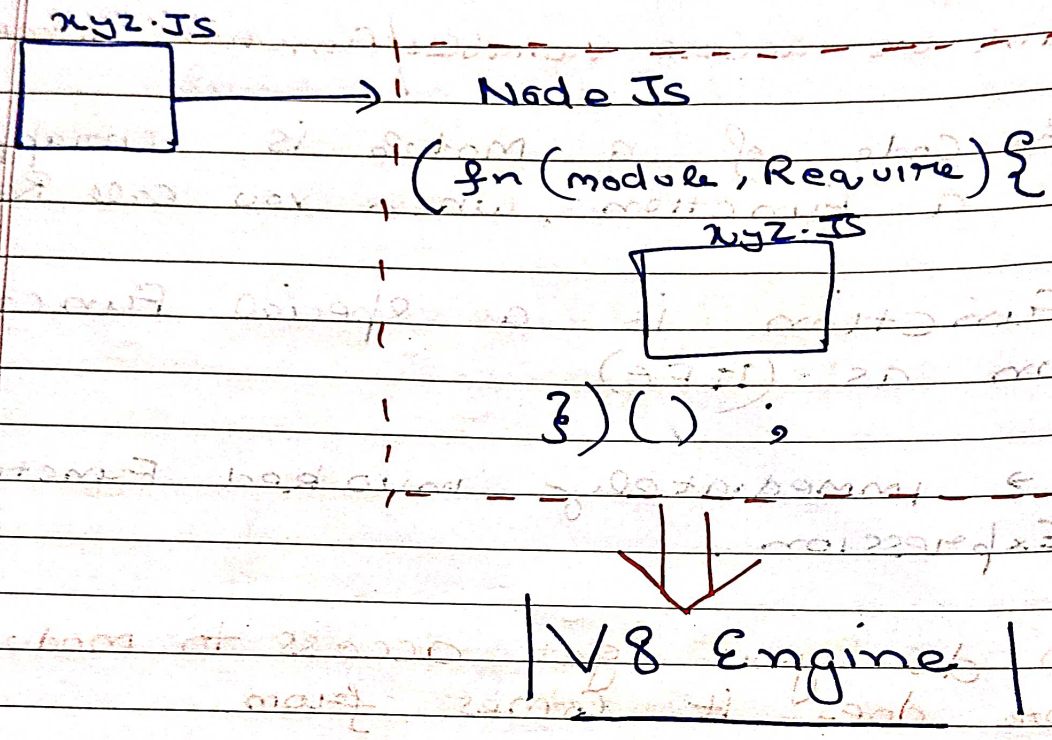
How do you get access to module.exports where does it comes from.

At the end of the day all the JS code is wrapped inside a function & gets executed (IIFE)

```
(function(module, Require) {
    require('/Path');
    module.exports = {}
})();
```

Given by Node.

Node Js Passes Modules as a parameter to IIFE in which the Code is wrapped



Node Js takes your Code wraps it inside IIFE & Sends it to V8 For execution. V8 Engine knows how to execute IIFE

This was all about the IIFE & its Working

5 Step Machinasm of Reavuire ('path')

① Resolving The Module

→ ./ local Path

→ . JSON

→ node : module

It sees from where the Data is comming And accordingly it resolve the module

② Loading The module

↳ File Content is loaded Ato File type

③ Wraps Inside IIFE

④ Evaluation → Code is executed & Returns

↳ module. exports

⑤ Caching