**Learn Node Js**

**Node Module System:**

**\*\*Module Wrapper Function:**

In Node Js, whatever module we create, is wrapped in a function before execution. This function is called a module- wrapper function.

This is what a module-wrapper function looks like:

1. (function(exports, require, module, \_\_filename, \_\_dirname) {

2. // Your module code actually lives here

3. });

4.

Here we can see that a module wrapper function has 5 parameters. **Remember, if we try to print any of the parameter values, the names need to be exactly the same that’s in the example.**

**Node Package Manager (NPM):**

**\*\*What is NPM:**

Node package manager or NPM is the default package manager for node js. It allows us to **install packages**, **manage dependencies**, **publish packages** and **run scripts to automate tasks like testing, building and deployment**.

**\*\*package.json File:**

It is a configuration file that contains the metadata about my project. It also contains:

1. Project details like name, author name etc.
2. Lists all dependencies
3. Scripts to automate tasks like starting the project or testing etc.

**\*\*Initializing NPM in project or folder:**

To initialize NPM in a project we go the folder that we want to initialize it in and type **“npm init -y”**. This initializes the npm in that folder and gives us a **“package.json”** file.

**\*\*Installing, updating and deleting packages from NPM:**

To install a package:

**npm install packageName**

We can also install multiple packages at once:

**npm install packageName1 packageName2**

to update a package:

**npm update packageName**

to delete a package:

**npm delete packageName**

**\*\*Adding a new script in package.json file:**

We can add a new script inside the package.json file under the “**scripts**” section. Inside the package.json file everything is a key value pair. For the scripts section, the key determines the action. Which means, what will it do when we run the script and the value determines with what command do we run the script.

Example:

**“start”: “node index.js”**

This means the index.js file should start or should be executed when we write **node index.js** in the terminal.