**Node JS**

**From Piyush Garg**

**(**[**https://www.youtube.com/playlist?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo**](https://www.youtube.com/playlist?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo)**)**

**Lecture 1 : What is NodeJS?**

**(**[**https://youtu.be/ohIAiuHMKMI?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo**](https://youtu.be/ohIAiuHMKMI?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo)**)**

* Node.js is a runtime environment for JavaScript. You should have a good understanding of JavaScript before using Node.js.
* **Node.js is a JavaScript runtime environment for creating web servers and APIs.**
* Node.js is not a framework or library, but a runtime environment.
* It allows JavaScript to be executed outside of the browser.

**✦JavaScript is a language of the browser that allows interactivity.**

* JavaScript can only be executed inside a browser because every browser has a JavaScript engine.
* Executing JavaScript outside the browser was not possible in the past.

**✦V8 engine is the most popular JavaScript engine**

* Different browsers have their own JavaScript engines
* NodeJS uses the V8 engine for executing JavaScript outside the browser

**✦NodeJS allows executing JavaScript outside the browser**

* NodeJS uses V8 engine extracted from Chrome
* V8 engine embedded with C++ allows JavaScript to interact with native machine

**✦JavaScript can now talk to native machines and perform tasks like file handling**

* JavaScript internally uses C++ through the V8 engine
* Node.js is a runtime environment for JavaScript to create web servers

**✦Node.js has a runtime environment for executing JavaScript code.**

* The V8 engine is used to execute JavaScript in the browser.

**✦Node.js is a runtime environment for JavaScript**

* Node.js allows JavaScript code to be executed outside of a browser, such as in a terminal
* Node.js is an open source cross-platform JavaScript runtime environment

**✦Node.js is a runtime environment for JavaScript.**

* Node.js is built on JavaScript, so a good understanding of JavaScript is necessary.
* A JavaScript tutorial series is recommended for those who are unfamiliar with JavaScript.

Browser with v8 Engine = Can Execute JavaScript within(INSIDE) the browser.

**BUT**

Browser with v8 engine Embedded(With the help of) C++ = Can Execute JS OUTSIDE the browser.

**Lecture 3 : Hello World NodeJS**

**(**[**https://youtu.be/XhCs5cTYW\_8?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo**](https://youtu.be/XhCs5cTYW_8?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo)**)**

**(**[**https://vscode.dev/github/hirtikmalvi/Web-Development/blob/main/5.%20Node%20JS/Hello%20World**](https://vscode.dev/github/hirtikmalvi/Web-Development/blob/main/5.%20Node%20JS/Hello%20World)**)**

* First Node JS File and how to execute it.

**✦Setting up a new NodeJS project**

Creating a new folder and opening it in a code editor

Organizing code files in separate folders for better tracking

**✦Create and Run Node JS file**

Create a new JS file named **hello.js** with a simple JavaScript code

Run the JS file using the terminal by specifying the file name as **node hello.js** and enter or simply node hello and enter

**✦Issues with window object and console in browser environment**

Window and console are not defined in NodeJS environment

Certain JavaScript functions behave differently in different environments

**✦NodeJS contains core functionalities needed on the server side**

NodeJS is essentially JavaScript with the core functionality for server-side use

Features like cryptography, file handling, and unnecessary client-side functionalities have been added or removed accordingly

**✦Node Package Manager is used to manage packages and dependencies in NodeJS projects.**

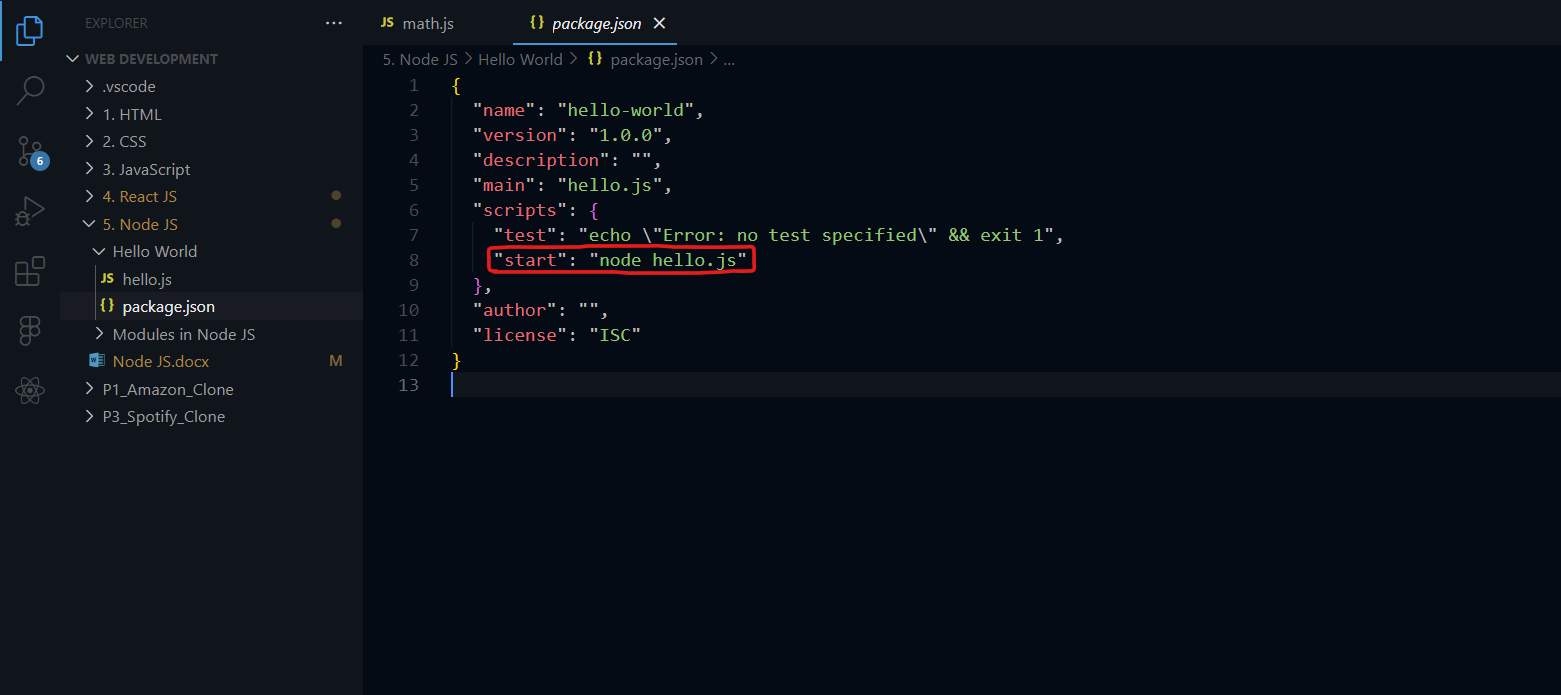
Using 'npm init' command we can initialize a new project and create a template.

During initialization, it creates a file for the project and asks for package name and other details.

**✦You can create your own scripts in NodeJS**

You can name the script whatever you want as follows

When you command as ‘npm start’ it will automatically converted into ‘node hello.js’ and will execute the file



**✦NodeJS provides a convenient way to create and manage scripts and dependencies.**

NodeJS allows for easy installation of dependencies and running scripts.

NodeJS includes a pre-generated package.json file for easy configuration and management of scripts and dependencies.

**Lecture 4 : Modules in NodeJS**

**(**[**https://youtu.be/FSRo41TaHFU?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo**](https://youtu.be/FSRo41TaHFU?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo)**)**

**(**[**https://vscode.dev/github/hirtikmalvi/Web-Development/blob/main/5.%20Node%20JS/Modules%20in%20Node%20JS**](https://vscode.dev/github/hirtikmalvi/Web-Development/blob/main/5.%20Node%20JS/Modules%20in%20Node%20JS)**)**

**✦Understanding modular programming in NodeJS**

Modular programming involves dividing the code base into small modules for better organization

This allows for creating different files for different functionalities and dividing the code accordingly. For example, we have different JS file that contains math function and we use it in different file.

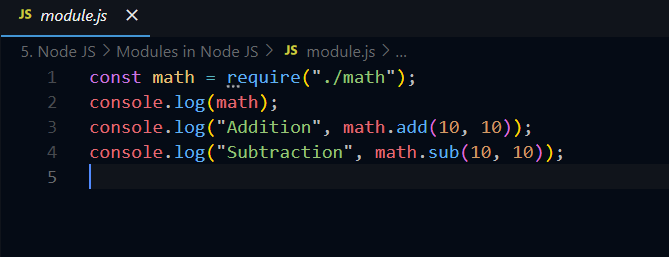
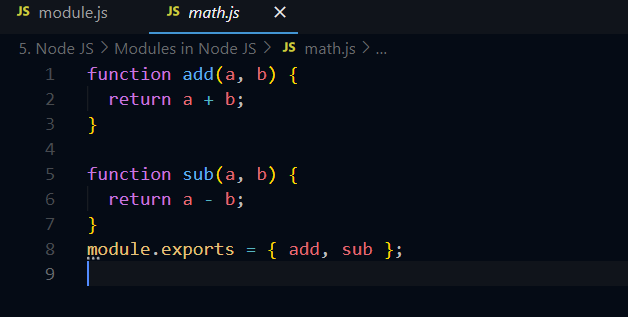
**✦Importing modules in NodeJS**

Use the require() function to import modules

Specify the module name or file path to import

**✦Exporting functions in NodeJS modules**

Functions can be exported from a NodeJS module using module.exports

Exported functions can be accessed and used in other files

**✦Exporting and importing functions and objects in NodeJS**

Functions and objects can be exported using the 'export' keyword

They can be imported using the 'required' function and utilized in other modules

Moreover, Node JS has built in packages which is also need to be imported as require(‘fs’). It is for file handling.

**Lecture 5 : File Handling in NodeJS**

**(**[**https://youtu.be/YazJFb\_i4A0?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo**](https://youtu.be/YazJFb_i4A0?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo)**)**

**(**[**https://vscode.dev/github/hirtikmalvi/Web-Development/blob/main/5.%20Node%20JS/File%20Handling**](https://vscode.dev/github/hirtikmalvi/Web-Development/blob/main/5.%20Node%20JS/File%20Handling)**)**

**✦Understanding file system in NodeJS and different types of tasks**

Explaining asynchronous and synchronous tasks in file handling

Differentiating between blocking and non-blocking tasks

**✦File handling in NodeJS involves reading and decoding different types of files.**

Files can be in different formats such as text or binary.

Decoding files requires understanding the file type and using appropriate methods.

**✦File handling in NodeJS involves synchronous and asynchronous operations with file reading and writing.**

Synchronous file handling returns the result directly, while asynchronous file handling uses callback functions to handle the result or error.

Understanding the concepts of synchronous and asynchronous operations in file handling is important for back end developers and anyone creating their own applications or startups.

**✦File handling in NodeJS allows creating, copying, and deleting files**

File handling includes read file, write file, append file, copy file, and delete file operations

Additional options such as unlink, file status, and sync can be used for more control over file handling



**Lecture 5 : How NodeJS Works?**

**(**[**https://youtu.be/y0aTs56DJWk?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo**](https://youtu.be/y0aTs56DJWk?list=PLinedj3B30sDby4Al-i13hQJGQoRQDfPo)**)**

**(…)**

**✦NodeJS architecture and request handling**

NodeJS architecture starts from client request to server

Incoming requests are handled within the event loop

✦**NodeJS uses event loop to handle requests efficiently**

Requests are picked up from a queue based on FIFO principle

Event loop differentiates between blocking and non-blocking operations to process requests

✦**NodeJS handles non-blocking and blocking operations differently**

Non-blocking operations are processed and the response is sent to the user

Blocking operations are handled by the thread pool, where a worker is assigned to fulfill the operation

✦**NodeJS architecture and event loop processing**

The event loop picks up requests and delegates non-blocking operations or assigns workers for blocking operations.

Thread pool with workers handles blocking operations, and limited thread availability can lead to processing delays.

✦**NodeJS has scalability issues with blocking operations.**

When the server uses blocking operations, it can cause excessive waiting time for users.

The difference between blocking and non-blocking requests in NodeJS is crucial for understanding its operation.

✦**Understanding the sequencing and execution in NodeJS**

In NodeJS, the execution starts from the top and blocks the thread, then continues once the file is read.

Converting code to non-blocking by removing synchronous and adding a callback function.

✦**NodeJS uses non-blocking requests to handle asynchronous operations.**

Non-blocking requests allow the program to continue executing other tasks while waiting for a response.

NodeJS event loop efficiently handles non-blocking operations and can be configured to increase worker threads.

✦**NodeJS architecture and thread management**

Maximum thread capacity is dependent on the machine and CPU cores

Operating System provides information about the computer and manages thread allocation