**NODE.JS**

Node Fundamentals.  
Express.js

MangoDB, Mongoose

Application

Deployment

**What is Node.js?**

Environment to run js Outside Browser.

Built on Chrome’s V8 JS Engines.

Big Community

Full-stack

**Requisites**

HTML, CSS, JS, ES6  
Callbacks, Promises, Async-Await

Youtube- Coding Addict

Playlist JS Nuggets

|  |  |
| --- | --- |
| **Browser** | **Node.js** |
| DOM | No DOM |
| Window | No Window |
| Interactive App | Server Side App |
| No FileSystem | FileSystem |
| Fragmentation | Versions |
| ES6 Module | CommonJs |

To Access REPL we has to type **node** in cmd prompt and then press Enter.

We are in REPL environment.

**Common.js**

Every file is module by default.

Modules: Encapsulated code (only share minimum)

Built-in module:

1. Operating systems
2. Path
3. File systems(fs)
4. HTTP

Modules Informations are written in Code files in the page.

NPM Fullform is node package manager.

Package.json is the principal place to configure and define how to interact with and manage your application.

**Nodemon-------------------------------------------------------------------------------**

**devDependencies ----** those packages in the package.json file that you need only for project development purposes.

Nodemon is a tool that helps develop Node.js based applications by automatically restarting the node application when file changes in the directory are detected.

Nodemon does **not** require *any* additional changes to your code or method of development. Nodemon is a replacement wrapper for node. To use nodemon, replace the word node on the command line when executing your script.

Installing Nodemon as Dev-dependencies-------------------------------------------

**npm I nodemon –save-dev**

**OR**

**npm I nodemon -D**

the above both works the same

**Watching the Files--------------------------------------------------------------------------**

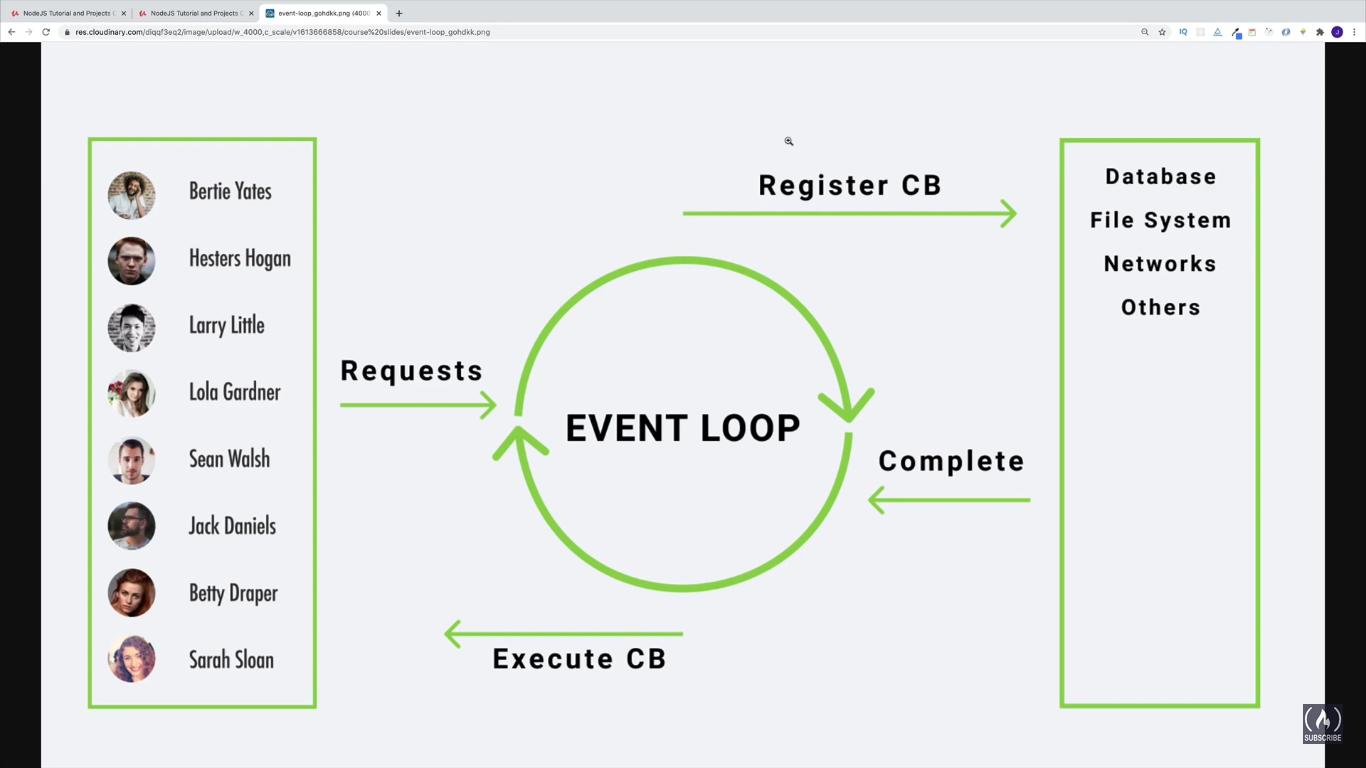
We have to make changes in the package.json file on “scripts” key’s value and change the object key to “start” and value as “nodemon filename”. Filename is the file where we are working and in cmdprompt when we use npm start the value which we give to the object key start gets executed and using nodemon we don’t always have to run the command in the prompt to run the file or a code.

Installing the nodemon globally.

**Package-lock.json----------------------------------------**

All the packages that were already install using npm install and all the extra packages we installed by our own information is situated in this file along with its version thought which we can install packages while working on different persons code.

**EVENT LOOPING-------------------------------**



To practice some of the API’s :- [**www.course-api.com**](http://www.course-api.com)

**CallBack Errors**

Asynchrounous timeout are the actions that we initiate now and they finish later

Eg. setTimeOut Synchronous action are the actions that initiate and finish one by one.

**CallBack Functions**

A callback function are the functions passed into another function as an argument, which is then invoked inside the outer function to complete an action.

Here is the example of a callback function:-

function loadScript(src, callback){   //We are Using Callback function as a input parameter so that we could call that function in another function this is callback functions.

    let script = document.createElement("script");// Here we are creating the elements

    script.src = src;  // Here we are assigning the src value to script.src to src attribute in the script tag.

    script.onload = ()=>{   // Here this event can apply to launch a particular function when the page is fully displayed. as the script tag created onload function occurs to it.

        callback();

        document.body.appendChild(script)  //Here the script tag is getting appended in the body

    }

}

function hello(){                       // this is one function, the function that we want to callback

    console.log("HEllo People!!!!");

}

function goodmorning(){                   // this is second function, the function that we want to callback

    console.log("GoodMorning People!!!!!")

}

loadScript("https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css", hello); // here we are passing and arguments and invoking the function.

This is called callback-based style of async programming.

    script.onerror = () => {

        console.log(`The site ${src} is incorrect.`);

    }

Onerror is used when the website or link we gave is incorrect.

**CallBack Hell // Pyramid Of Doom;**

When we have callbacks inside the callbacks, the code gets difficult to manage,



This is Pyramid of doom or call back hell. Callback hell is the essentially nested callbacks stacked below one another forming a pyramid like structure.

The pyramid of these calls grows towards the right with every asynchronous action. Soon it spreads out of control. So, This way of codding isn’t very good.