NODE JS

NodeJs is an open **source** and **cross plateform** Javascript runtime environment .

Usage of NodeJs can be found in video calling apps , and chat apps.

Node.js runs the V8 JavaScript engine , the core of Google Chrome, outside of the browser.

A Node.js app runs in a single process ,**without creating a new thread for every request.**

It works on **asynchronous , non blocking I/O Model.**

Asynchronous operations allow Node.js to serve multiple requests efficiently.

“ Blocking is when the execution of additional javaScript in the Node.js process must wait util a non Javascript operation completes. This happens because the event loop is unable to continue running JavaScript while a blocking operation is occurring. “

When Node.js performs an I/O operations, like reading from the network ,accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

NPM

NPM is a packet manager which manages the packages of the node.js . It facilates us to install packages just like pip in python.

> PIP is a package manager for python packages.

Dev Dependencies are dependencies which we want to use during web development like nodemon. Such dependencies are written under Dev Dependencies.

In package json , it has the dependencies that we have taken.

> In **Package Lock Json** , the entire dependencies tree is there.

Node.js has two module systems : CommonJs modules and ECMAScript modules. By default , Node.js will treat the following as CommonJS modules - > files with .cjs extension , or files with a .js extension when the nearest parent package .json file contains a top-level field “type” with a value of “commonjs”.

There are multiple modules in Node.js : -
OS , path , url , file , HTTP, event , and many more.

IMPORTING A MODULE IN NODE.JS

File : index.js

```
const lovish = require(“./second”);
console.log(“Hello World”,lovish);
```

File : second.js

```
Harry = {
        name : “Harry”,
        favNum : 36,
        Developer: true
}
module.exports = harry;         ( This will export your object )
```

IN ES MODULE (extension .mjs)

```
// import { simple , simple2} from “./modulesecond.mjs”
import * as a2 from  “./modulesecond.mjs”

//const simple2 = require(“./modulesecond.mjs”);   This will give error!!


//simple23()

console.log(a2.simple())
```

File : modulesecond.mjs

```
export function simple(){
        Console.log(“Simple is complex”);
        Return 45;
}

Export default function simple2(){
        Console.log(“Simple2 is complex”)
}

//module.exports = simple;    ( This will certainly give error!! )
```

OS MODULE IN NODE.JS

EX :-

```
        const os = require(‘os’);

        Console.log(os.freemem());
        Console.log(os.homedir());
        Console.log(os.hostname());
        Console.log(os.platform());
        Console.log(os.release());
        Console.log(os.type());
```

PATH MODULE IN NODE.JS

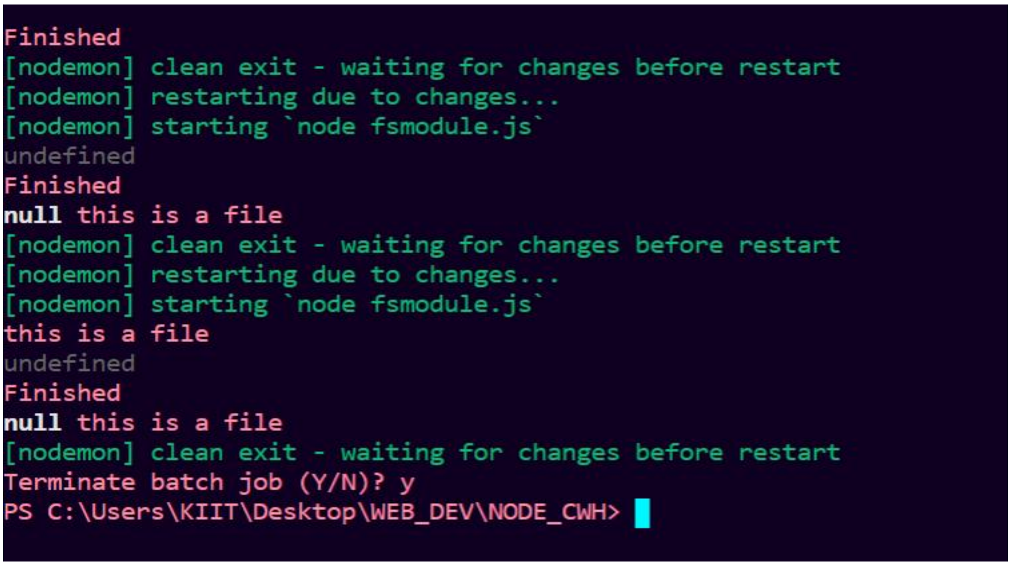
```
Const path = require(‘path’);

Const a1 = path.basename(‘C:\\\temp\\myfile.html’);
Const a2 = path.dirname(‘C:\\temp\\myfile.html’);

Console.log(“Printing a1 constant variable”,a1);       [ will give myfile.html output ]

Console.log(“Printing a2 constant variable”,a2);       [ will give the C:\temp as output]


Const a3 = path.extname(__filename);          // will return the extension of the file.

Console.log(__filename,a3)    // __filename give the entire file address
```

URL MODULE IN NODE.JS

```
Const myURL = new URL(‘https://example.org:8000’);

myURL.pathname = ‘a/b/c’;
myURL.search = ‘?d=e’;
myURL.hash = ‘#fgh’;

Console.log(myURL);
Console.log(myURL.href);
```

FS MODULE IN NODE.JS

```
Const fs = require(‘fs’);

//Here the file reading is asynchronous

Fs.readFile(‘file.txt’,’utf8’,(err,data)=>{
        Console.log(err,data);
});



//Now the file reading is synchronous

Const a = fs.readFileSync(‘file2.txt’);

Console.log(a.toString());




Fs.writeFile(‘file2.txt’,”This is a data”, ()=>{
        Console.log(“Written to the file”);
});


B = fs.writeFileSync(“file2.txt’,”This is a data2”);

Console.log(b);

Console.log(“Finished”);
```



```
Finished
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node fsmodule.js`
undefined
Finished
null this is a file
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node fsmodule.js`
this is a file
undefined
Finished
null this is a file
[nodemon] clean exit - waiting for changes before restart
Terminate batch job (Y/N)? y
PS C:\Users\KIIT\Desktop\WEB_DEV\NODE_CWH>
```

CREATING A SERVER USING NODE.JS

```
Const { Console } = require(‘console’);

Const http = require(‘http’);

Const port = process.env.PORT || 3000;

Const server = http.createServer((req,res)=>{
        Console.log(req.url);
        Res.statusCode = 200;
        Res.setHeader(‘Content-Type’,’text/html’);
        Res.end(‘<h1> This is Code with harry </h1> <p> Hey this is the introductory page of our website.</p>’);

});


Server.listen(port, ()=>{
        Console.log(`Server is listening on port ${port}`);
});
```

The above is a simple demonstration of creating a server .

Next we will create website using nodeJS with different request url to go : like /cwh and /about .

```
});
```

CREATING A DEMO WEBSITE USING NODE.JS

```
Const fs = require('fs');
Const http = require('http');

Const port = process.env.PORT || 3000;


Const server = http.createServer((req,res)=>{
        Res.setHeader('Content-Type','text/html');
        Console.log(req.url)

        If(req.url == '/'){
                Res.statusCode = 200;
                Const data = fs.readFileSync('index.html');
                Res.end(data.toString());
        }


        Else if(req.url == '/cwh'){
                Res.statusCode = 200;
                Res.end('<h1> This is Code with Harry </h1> <p> hey this is the
introductory page of our website </p>');
        }

        Else if(req.url == '/about'){
                Res.statusCode = 200;
                Res.end('<h1> About Code With Harry </h1> <p> Hey this the about
page of our website</p>');
        }


        Else {
                // res.harry();  This function does'nt exist so when you will run the
                //server and the server goes to this part of the code ,
                //Your 'SERVER WILL CRASH'

                Res.statusCode = 404;
                Res.end('<h1> Not Found </h1> <p> Hey this page is not found </p>');
        }
});
```

EXPRESS JS

> Express is a node js web application framework that provides broad features for building web and mobile applications .

> To install the express you need to write the command : -
npm install express

Example : -

```
Const express = require(‘express’);
Const app = express();
Const port = 3000;

App.get(‘/’,(req,res)=>{
        Res.send(‘Hello World2!’);
});

App.get(‘/about’,(req,res)=>{
        Res.send(‘This is an about page!’);
});

App.listen(port,() =>{
        Console.log(`Example app listening at http://localhost:${port}`);
});
```