What is Node js

- This is Server envronment

- the different between node js and Plain Js is node js is connected with data base

- But js not connected with data base there is difference

- because node js run on Server side that why it connected on data base

- Node js is free we not required any single Paisa to used that why it free

- Node js use Chrome's v8 engine to execute code.

why do we used node js

- we can used node js for API

- and for data base Purpose.

- with the help of node js and js we can become full stack developer

- written in C, C++, Javascript

- client and server.

- client side means - JS, Html, css

- Server side - node JS

- V8 engine written in c++

- fundamental of node JS

- var → Keyword for declaring variable

- for example var a=10
  console. log (a)

  output → 10

  If const a=10
  then aggain we assign value
  like this

  a=10

  output
  error getting

- Printing Sum Program
  in node JS

  let x=10
  let y=20
  console.log(x+y)

  output will be

  → 30

* another Program from check content value is Same or not

```
* var x = 10
var y = 10

if (x===y) {
        console.log('match')
}
```

output → match

```
var x = '10'
var y = '10'
if (x===y)
        console.log('match')
```

* array in node js

```
let arr = [10,20,30,40]
Console.log(arr[0]).
```

Output will be

→ 10

* another array Program

```
let arr = [10, 20
```

output If will check value as well as Same type ; that why out put not Showing

* Loop → node js

```
for (var i=0; i<10; i++){
        Console.log(i)
}
```

* In module Part If we use export for example

→ App.Js file

```
export let a=10
export let b=20.
```

then

→ index. Js file there
If we import these file then

```
Import {x,y} from app.Js
```

then output will be show
cannot use import statement outside a module.

object

then we used module.exports = {x:10, y:20}

for example App.Js file there

function

```
const app = require('APP Js')
```

then output will be show.

core module in Node js

- * What is the core modules
- * what are global module
- * Global model example
- * non-global module example

→ Build in function are available node js give this feature this is called core module (fs Buffe, HTTP)

→ global modules means we not ~~im~~port explicity ~~are~~ like conde - log function

→ fs → is non global function

→ fs. write file syn

→ first we have to export

const fs - require (' fs');

request and response http Server handle·

① const http= require('http');

```
http. createServer ((req, resp) => {
        resp. write ("<h1> Hello this is Hello
                world" <h1>);
    resp. end ();
  }). listen (4500);
```

Or second way

```
              def
  function (req, resp) {
        resp. write ("Hello world");
        resp. end();

      }
   http. createServer (def). listen (4500);
```

→ all about Package. Json

→ what is package. Json → marge Package

→ It Store detail Code detail ~~[illegible]~~e

→ npm init → node Package manage →
marge all the Packe

first file run entry file

Keywood = (step by step)

In Package.JS → we can See Which Package we use →

→ Nodemon module

→ Nodemon module we can use time Perpose

→ for example

we we run console.warn("TryNodemon") then → run output will show

then again if we changes then again we run the code again and again that why time waste

→ that why we use nodemon module

→ first we download nodemon -g

-g → global export we can use at any Project

→ node mon index.js → run

→ node mon example

→ console.warm("code step by step")
→ console.log (100+200)

→ we can run once only and change many tim

for http resquest response server

```
① const http = require ('HTTP)

   http. crete server (req, resp) } ⇒ {
                 response
   } ) .listen(6000)

① const http= require (' HTTP'),
   http. create Server ((res, reqs) ⇒ {
       reqs. writHead (200, {'conten':
           'application\json }) ;
       resp. write('{
       resp.write(JSon. Stringify({
       name: 'Asad', email :'ahes'})});
```