Asynchronous and Synchronous

* Synchronous means → tasks will perform one time

→ means that at one time task will perform

→ Users — funtion

→ If user function work then another task perform

Product → task perform second

→ An Asynchronous means → Second task can not wait from one task.

One

for example Synchronous

```
console.log("Start")
Console.log("Node")
Console.log("Programming")
```

→ this one time Perform →

→ If task assign then second task Perform

→ then third task Perform

and output will be

→ Start

→ Node

→ Programming

then Asynchronous Programming means

Second task can not wait for first task perform.

→ for example

```
→ Consde.log('start')
  Settime(()=>{
      console.log("Node is")
  },2000)

  consde.log('Java script')
```

→ first task comple
→ then 2nd task take time so that why then 3rd task will perform then 2nd task perform

this is the example of Asynchronos

→ basic example meas that at one time task will perform 2nd task wait form one 1st to comple

→ Second task can not wait for first time this is called aschrony tion.

Impedant Point

How node js work internally

- what is call stack?
- what is node API?
- what is call Back Queue
- Example

① Architecture

① call back → defenetly work

② Node API

③ call back Que

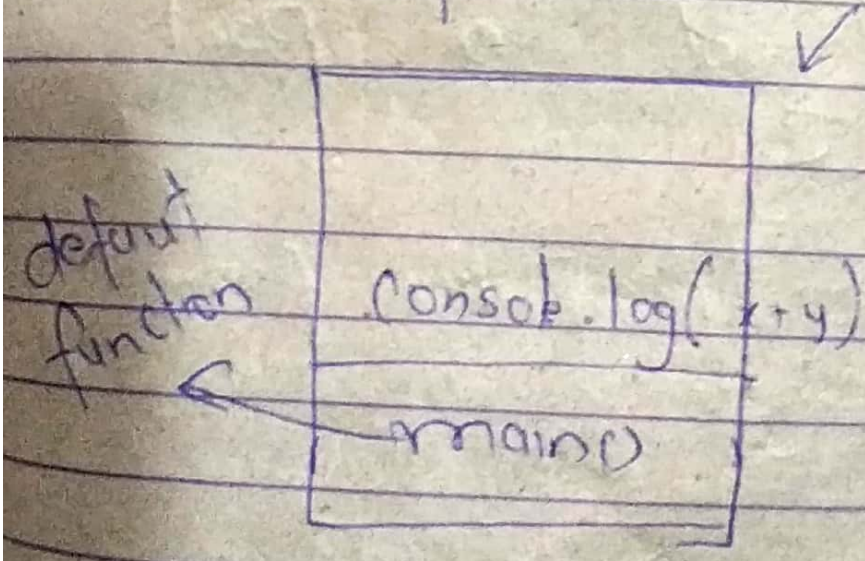
these all Process is called Event loop

for example if we execute one front program basic

```
let x=1
let y=2
console.log(x+2)
```

for example

→ call stack
→ store all the function registery
→ which function will execute first



Out - Sum is 3

then, main()

for example of Node API

Code

```
Consde.log('Hello')

Settimeout (() => {
    (console.log('Hi')}}. 2000)

Set time out ( () => {
    console.log('Hello') }. 0)

Cons ole dog('Js')
```

→ Set timeout is not a part of Js, node Js

→ it inherit these properties from C then Why

→ it login for Node API

→ digaram

PAGE NO.:
DATE: / /

console.log() console.log() main()	Set time out(2 sec) Set time out(1)

callback Kque

→ then

→ Set time out (one) in callback

first

→ then Set time out (2) callback in queue

→ Express js is frame work for node js

→ Jus like Angular gs is frame work

→ and Spring from Java frame that same as Express Js It a frame work of node Js.

→ By using we can get something and request something and response data etc.

→ for Express Js we use Package install Package of express Js

npm install express

→ this is the first step to connect with express js module

Import module

PAGE NO.:
DATE: / /

① const express = require('express')

→ Cont app = express() → executable module

```
app.get('', (req, resp) => {
    resp.send('')
    console.log('first page')
}).listen(2000)

app.get('', (req, resp) => {
    resp.send('first page')
})
app.listen(5000);
```

→ another second example of node js and express JS

① First import module

```
Const express = require('express')
```

② second convert into executable mode

```
const app = express();
```

③ then call method get

```
app.get('/hello', (res, resp) => {
    resp.send('Second page')
})

app.listen(5000);
```

Now task is connect with DB like mongo DB to connect with node js

first know that what is mongoDB

→ mongoDB is document data Base

→ no sql not only sql data base

→ Data will be store in the from of document

→ first we have to create data base in Terminalike

Command

① mongo

② mongod

the by typing

→ usedbs → by using this command to show how many data bas

then we create collection

→ command
db. Create collection ('You')

PAGE NO.:
DATE: / /

Db.you.InsertOne({ Json form })

→ this command is used to insert element in data base

→ Most important thing is to connect with nodes is as mongo DB

→ First Install Package modeule

→ npm i mongodb

then Import module,

→ const { mongo client } = require ('mongodb')

And then store url of mongoDB Second step

const URL = 'mongodb://localhost:2701'

then store url into client

const client = new mongo client (url

→ then create function

→ let result await client.connect()

→ let db =result .db('mys')

→ let then create collection

→ let collection = db.Collection('you')

and it connect all data base

→ full example

```
const {mongo client} = require ('mongodb')
Const url = 'mongodB://localhost:27017'
Const Client = New mongo client (url)
async function getData() {
                await
  let result = client.connect()
```

```
let db = result.DB('mys')
let collection = db.collection('ya')
let data = await collection.find({}).toArray()
console.log(data)
}
getData()
```