Node Js Tutorial

Using a if loop in Node Js

Variable

```
Var x = 20;
if (x == 20) {
    console.log ("matched")
}
```

Using for loop in Node.js

```
Var x = 20
for (i = 0; i <= 10; i++)
  {
  Console.log (i)
```

Array in node.js

```
var x = '20'
Const arr = [2, 4, 6, 8, 10, 12, 14, 16.]
  console.log (arr[0])
```

* How i import a code of one file to another.

index.js file

```
module.exports = {
    x: 10
    y: 20
```

App.js file

```
Const app = require('./app'
Const arr = [2, 4, 6, 8]
Console.log (app.y)
```

Interview Questions

filte function use → If we want to find a array then we use filter function.

Syntax → arr.filter ((item) ⇒ {
    console.log('item')
})

Another Syntax

```
let result = arr.filter ((item) ⇒ {
    return item = 4
})
console.worn (result)
```

* Core Modules in Nodejs Global Modules Example

Modules → Module are by default file

Core Modules. ⇒ Core Module is a module is which is ultimately defined in our programming Language.

Global Module ⇒ Global module is ultimately defined in our programming language which we can use in our program.

Global Module → The module is a module which is ultimately defined in our programming language.

Non Global Module → The module which we use to import while using them are know as Non global module.

* Syntax To Create a Text file in node JS

```
Cons fs = require('fs')
fs.Writefilesync("Index1.txt",)
```

* To check our directory that on which directory we are working?
To access the physical file system.

```
Cons fs = require("fs")
console.log(".", __mydir)
Console.log("->>" __filename)
```

56 minute

Make basic Server Output on Browser topics.

- How to Make basic Server
- function as Parameter in node
- In terview Questions.
- Arrow function
- Get output on Browser

Server ESM / nodejs ke and on Server Ki Request ko respond karkh

→ Const http = require('http');
→ http.createServer(() ⇒ listen(4500);
→ ~~http.create~~ }).listen 4500

Second Method

```
Const http = require ('Http')

http.createServer ((req, resp) => {
    response.write ('Hello this is Mudit Sharma')
    resp.end ( );
}) .listen (4500)
```

Node.Js is asynchronous.

Node.Js is asynchronous because it does does to wait for one commands if one method is taking time to execute tho it inwhile time it transfer to another and the another command will execute.

*

```
Const http = req ('Http')

Http.createServer ((req, resp) => {

resp.writeHead (200, {'content-Type' : 'application\Json
resp.write (JSON.stingify ({name : 'Anilsidhu', email :
resp.end
}) .listen (6000);
```

* Creating a file System in node.js

```
Const fs = require('fs')
fs.writefilesync ('file.txt', 'This is a file');
```
→ In this we create

```
const fs = require ('fs')
const path = require ('path')
const dirpath = path.join (__dirname, 'files');
for (i=0; i<5; i++)
{
    fs.writefilesync ("Hello" + i + ".txt", "a simple text
                      (          $i
}
```

* Sending the Message to the server by using var fun

```
Var http = require ('http')
var Server = http.create server (function (req, Resp) {
console.log ('request was made: ' + req.url);
resp.writehead (200, 'content-Type': 'application/Json'});

Var myObj = {
        name: 'Rahul',
        Age: '30,
        developer: 'true'
    };
    res.end (JSON.stringify (myobj));
});
server.listen (3000, '127.0.0.1');
Consok.log ('listening to port 3000');
```

Code for Performing CRUD operations.

CRUD operation with file System
(C - create, D - Delete, R - Read, U - updat)

C - Create
R - Read
U - update
D - delete.

```
Const fs = require ('fs');
Const path = require ('Path');
Const dirpath = path.join (__dirname, 'crud');
Const filepath = `${dirPath / apple.Txt;
```

It is used to create file

```
# fs.writeFileSync (filepath, 'This is a system text file');

// fs.readfile (filePath, 'utf8' (err, item) => {
//     Console.log (item); — Read a file
//     })
// fs.appendfile (filepath, 'and file name is apple txt, (err) =>
    if (!err) console.log ("file is updated")
})          // update the file

⇒ fs.rename (filepath, `${dirPath}/fruit.txt`, (err) => {
    if (!err) Consok.log ("file name is updated")

        // Update the file
    fs.unlinkSync (`${dirPath}/fruit.txt`)
```

It deletes the file.

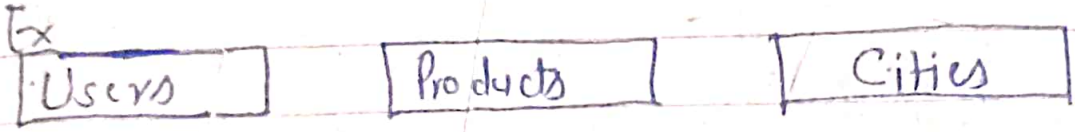
Q. What is a Buffer.?

By Buffer simply means a temporary Memory location which is used to performs the operations is known as Buffer.

Synchronous and Asynchronous.

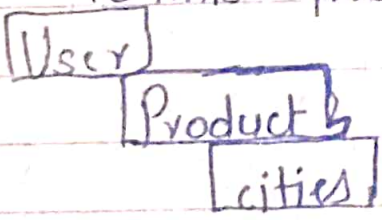
Synchronous → In synchronous operation task are perform one at a time

Ex



Asynchronous → In Asynchronous, Second Task don't wait To finish first task.

{



Drawback

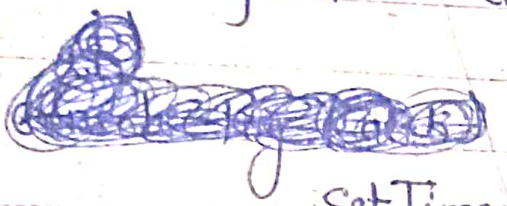
Ex

```
console.log ('start.exe...")          let a=10;
setTimeout (() => {                    let b= 0;
        Console.log ("logic exe...")
        }, 2000)                       b=20
    Console.log ("Complete exe...")

    Console.log (a+b)
```

We have use Promises and Async await in handling the Asynchronous drawbacks.

```
let a = 20
b = 0;
setTimeout (() => {
    b=30;
}, 2000)
let waitingData = newPromise ((resolve, reject) => {
```

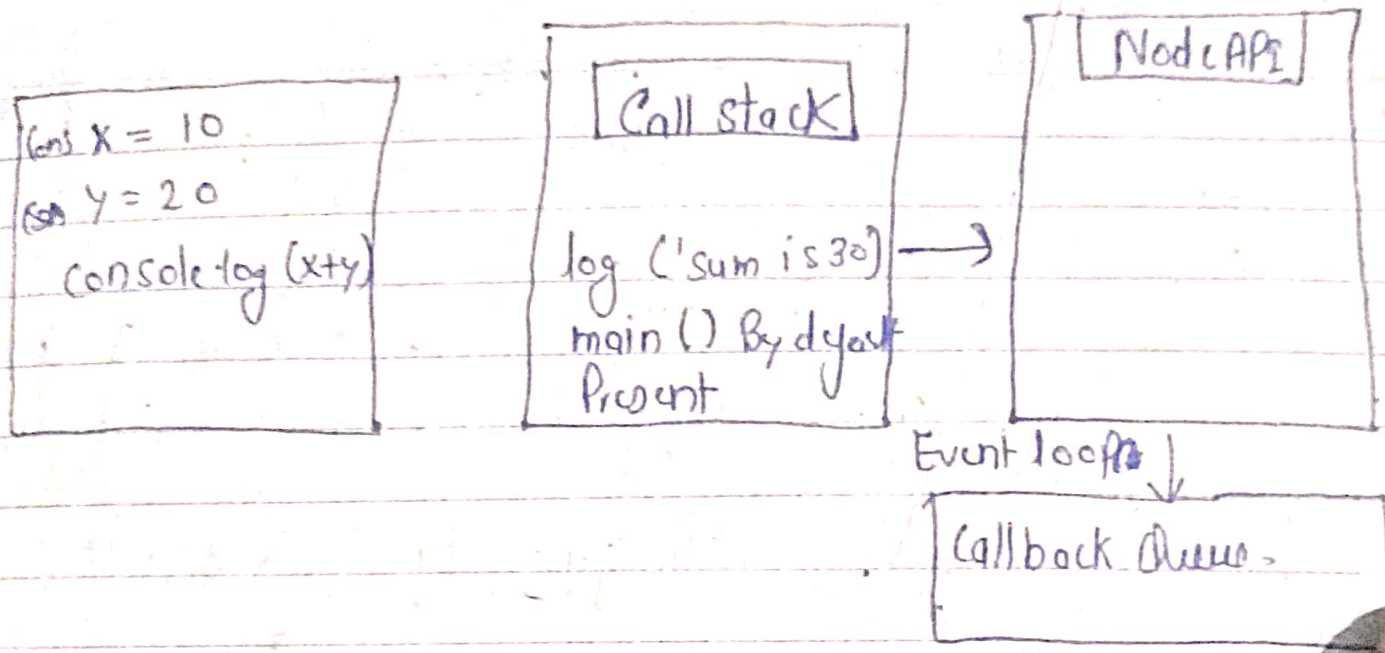


```
    setTimeout (() => {
        resolve (30)
    }, 2000
})
waitingData.then ((data) => {
    b=data;
    console.log (a+b)
```

How Node JS works.

1. call stack 2. Node API 3. call back Queue
4. Example

Architecture of Node JS.




```
Console.log ("starting up")
  setTimeout (() => {
      Console.log ("2 second log")
  }, 2000
Console.log ("finishing up")
```

Repeat

*

* Express Js

Q.1 What is Express Js ?

Ans:- Express Js is basically a framework of node Js

1. We can make api easily
2. We can get routes easily
3. api? we can make in 2 to 3 lines instead of 15 lines in Node J.S.

Q. How to install Express

run Command ~~Express~~

npm install express --Save

Code for making API By using Express.

```
Const express = require ('Express');
Const app = express ();
app.get ('', (req, resp) => {
  resp.send ('Hello, this the Homepage');

app.get ('/About', (req, resp)
  resp.send ('Welcome, This is About page')
```

Template Engine

Q. What is Template Engine?

A Template engine is basically used for making the dynamic pages. and we have to install.

Q. What are Dynamic Page?

A When we getting all the information that we are requesting are coming from the database then it is known as the Dynamic page.

Q. What are static Page?

A The data that we want to show by writing some code as it is shown to us is known as static page.

Q. Why we use EJx only?

A Install Command → npm install ejs.