

**Part A (To be referred by students)**

|  |  |
| --- | --- |
| **Name: Arjun Mehta** | **Roll no: K036** |
| **Program/Branch : B. Tech Cybersecurity** | **Batch: K1** |

**Topic covered:** To Study the NodeJs

**Learning Objective: Learner would be able to**

1. To create HTTP request module to connect with server.
2. To Build an HTTP server using the core modules in Node.js
3. To Create modules to organize the server.

**Prerequisites:-**

- Javascript

**Outcomes:-**

- Student will know about web server and HTTP protocols that will allow to connect Nodejs to Mysql Server.

**Aim: To learn and implement server-side scripting using Node.js Prerequisites:-**

- Basic Tags of Javascript

# Theory:-

**Server Side Scripting**

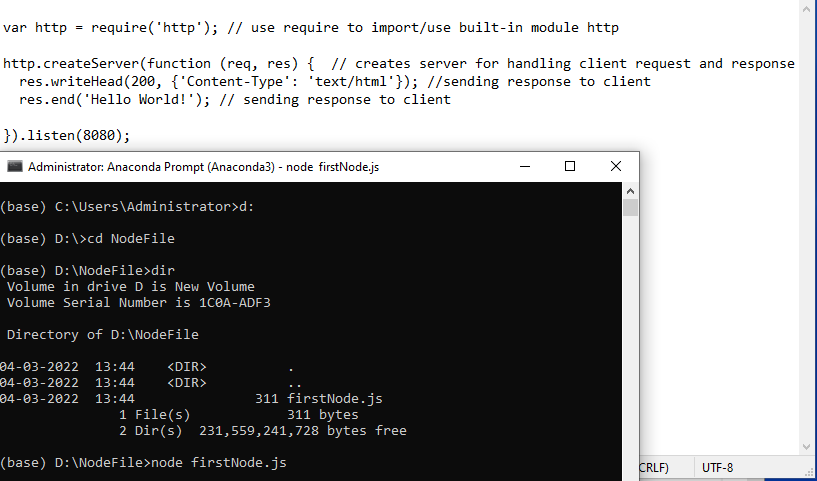
Dynamic web pages are created using server-side scripting. JavaScript is very successful in handling client-side scripting. The functionalities are extended to handle server-side scripting. The major advantage of using node.js is asynchronous communication. In synchronous communication system (ex php, ASP), client request is sent to server and it waits to fetch desired information from files. After fetching it sends content to client. Then its ready to handle the next request. Where as in asynchronous communication, client is sent to server then it is ready to take the next request. Once server is ready with content it is sent back to client. Node.js eliminates the waitingtime.

So Node.js runs single threaded, non-blocking, asynchronous programming and memory efficient.

# Download Node.js

The official Node.js website has installation instructions for Node.js: [https://nodejs.org](https://nodejs.org/)

## First Node.js program



Save the file in folder D:\NodeFile as firstNode.js, then run the server as node firstNode.js

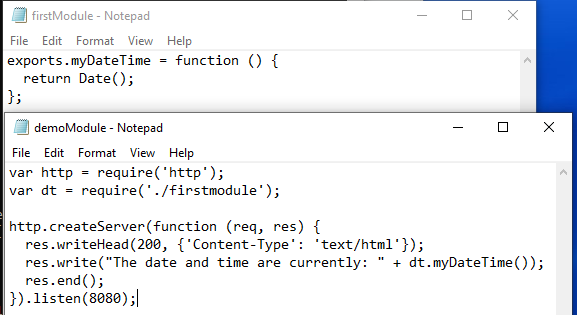
Open browser and request port 8080 http://localhost:8080 to view response.

## Create own Module

We can create own module and expose properties and methods.

// Create a module that returns the current date and time:

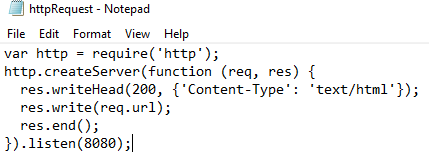
Save it as firstModule.js, start the node file



Create another file demoModule.js, then open node file in cmd then open http://localhost:8080/

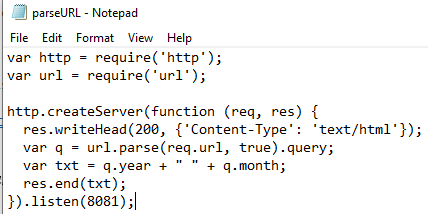
* 1. ***Querying HTTP request***

“Req.url” will read text from URL sent to server



http://localhost:8080/hello how are you

## Parsing the HTTP URL



http://localhost:8081/?year=1990&month=5

## File handling with node JS

Create file named demofile1.htmland add following content

<html>

<head>

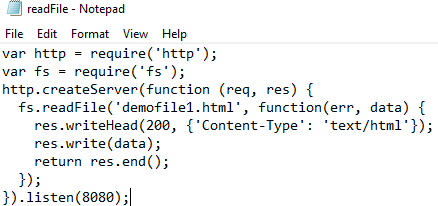
</head>

<body>

<b>Hello i am from HTML.

</body>

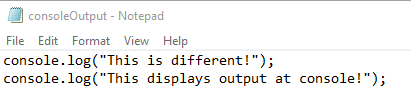
</html>



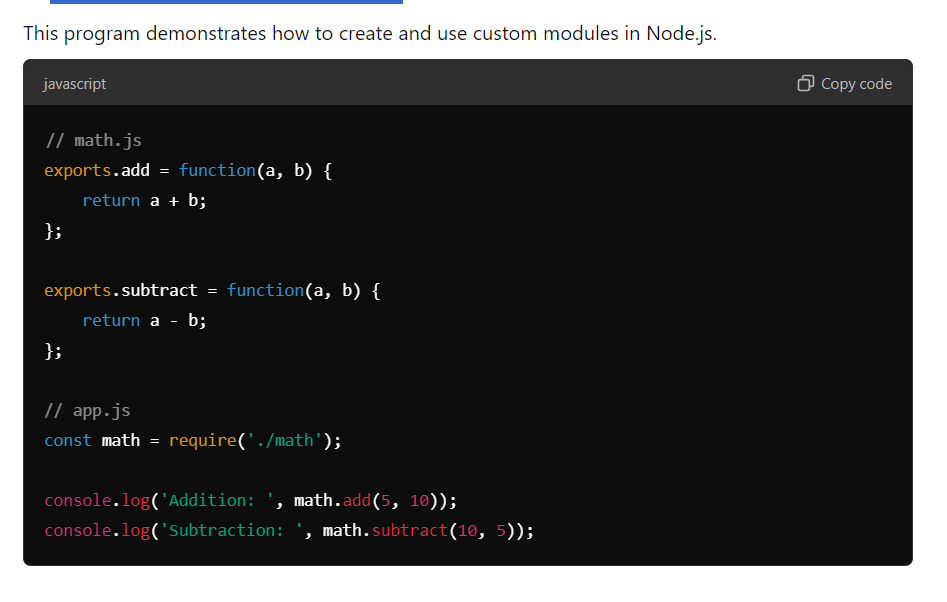
Create node file to read content from html file and display the output in browser.

http://localhost:8080/

***6. Displaying output at console***



Simple Math Module Example



# http Module:

Part B

1. Write a program in node.js to parse the given url.
   1. <https://portal.svkm.ac.in/MPSTME-NM-M/markAttendanceForm> ii) [https://www.google.com/search?q=google&rlz=1C1CHBD\_enIN941IN9](https://www.google.com/search?q=google&rlz=1C1CHBD_enIN941IN941&oq=google&aqs=chrome.0.0i131i355i433i512j46i131i199i433i465i512j0i131i433i512j69i60l5.2029j0j4&sourceid=chrome&ie=UTF-8) [41&oq=google&aqs=chrome.0.0i131i355i433i512j46i131i199i433i465i5](https://www.google.com/search?q=google&rlz=1C1CHBD_enIN941IN941&oq=google&aqs=chrome.0.0i131i355i433i512j46i131i199i433i465i512j0i131i433i512j69i60l5.2029j0j4&sourceid=chrome&ie=UTF-8) [12j0i131i433i512j69i60l5.2029j0j4&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=google&rlz=1C1CHBD_enIN941IN941&oq=google&aqs=chrome.0.0i131i355i433i512j46i131i199i433i465i512j0i131i433i512j69i60l5.2029j0j4&sourceid=chrome&ie=UTF-8)

Display the parsed content in console and in browser.

Code:

1)

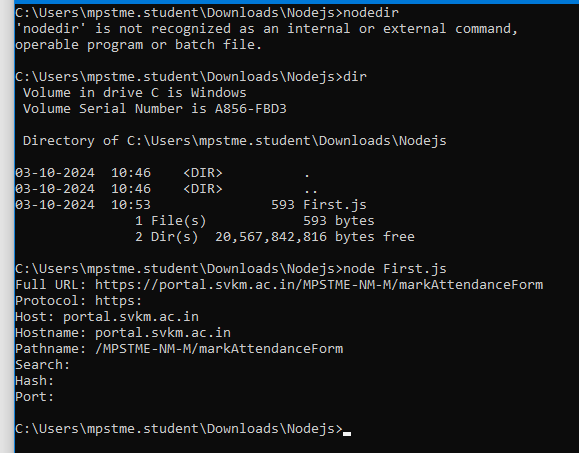
// Import the URL module  
const url = require('url');

// Given URL  
const givenUrl = 'https://portal.svkm.ac.in/MPSTME-NM-M/markAttendanceForm';

// Parse the URL  
const parsedUrl = new URL(givenUrl);

// Log the different parts of the URL  
console.log("Full URL:", parsedUrl.href);  
console.log("Protocol:", parsedUrl.protocol);  
console.log("Host:", parsedUrl.host);  
console.log("Hostname:", parsedUrl.hostname);  
console.log("Pathname:", parsedUrl.pathname);  
console.log("Search:", parsedUrl.search);  
console.log("Hash:", parsedUrl.hash);  
console.log("Port:", parsedUrl.port);

Output:



2)

Code:

// Import the URL module

const url = require('url');

// Given URL

const givenUrl = 'https://www.google.com/search?q=google&rlz=1C1CHBD\_enIN941IN9 41&oq=google&aqs=chrome.0.0i131i355i433i512j46i131i199i433i465i5 12j0i131i433i512j69i60l5.2029j0j4&sourceid=chrome&ie=UTF-8';

// Parse the URL

const parsedUrl = new URL(givenUrl);

// Log the different parts of the URL

console.log("Full URL:", parsedUrl.href);

console.log("Protocol:", parsedUrl.protocol);

console.log("Host:", parsedUrl.host);

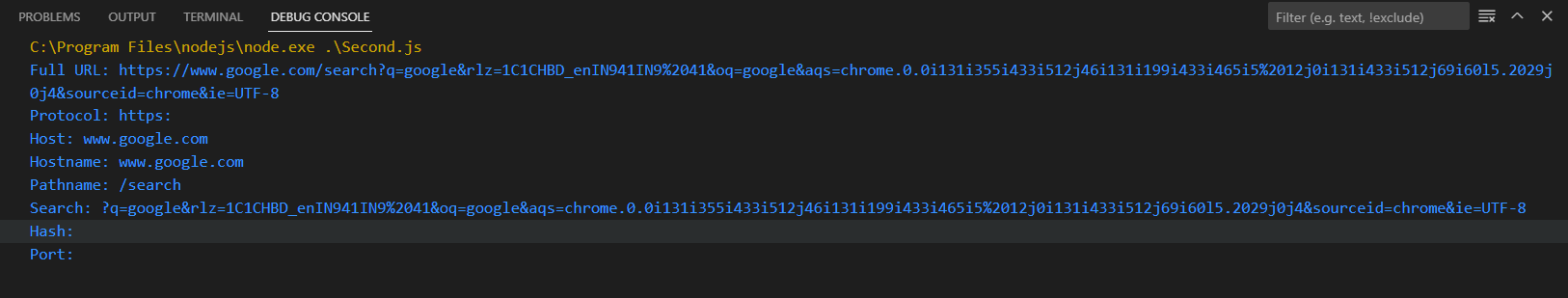
console.log("Hostname:", parsedUrl.hostname);

console.log("Pathname:", parsedUrl.pathname);

console.log("Search:", parsedUrl.search);

console.log("Hash:", parsedUrl.hash);

console.log("Port:", parsedUrl.port);



1. Create a txt file with following content and display content on browser and console.

File content as below

Node.js is a super popular s erver-side platform that more and more organizations are using. If you are preparing for a career change and have an upcoming job interview, it’s always a good idea to prepare and brush up on your interview skills beforehand. Although there are a few commonly asked Node.js interview questions that pop up during all

types of interviews, we also recommend that you prepare by focusing on exclusive questions to your specific industry.

Code:

// Import the required modules

const fs = require('fs');

const http = require('http');

// Content to be written to the file

const fileContent = `

Node.js is a super popular server-side platform that more and more organizations are using.

If you are preparing for a career change and have an upcoming job interview, it’s always a good idea to prepare and brush up on your interview skills beforehand.

Although there are a few commonly asked Node.js interview questions that pop up during all types of interviews,

we also recommend that you prepare by focusing on exclusive questions to your specific industry.

`;

// Write the content to a .txt file

fs.writeFileSync('file.txt', fileContent.trim());

// Display file content in the console

const data = fs.readFileSync('file.txt', 'utf8');

console.log("File content:\n", data);

// Create a simple HTTP server to display file content in the browser

http.createServer((req, res) => {

  // Set the response header to display plain text

  res.writeHead(200, { 'Content-Type': 'text/plain' });

  // Read the file and send the content as a response

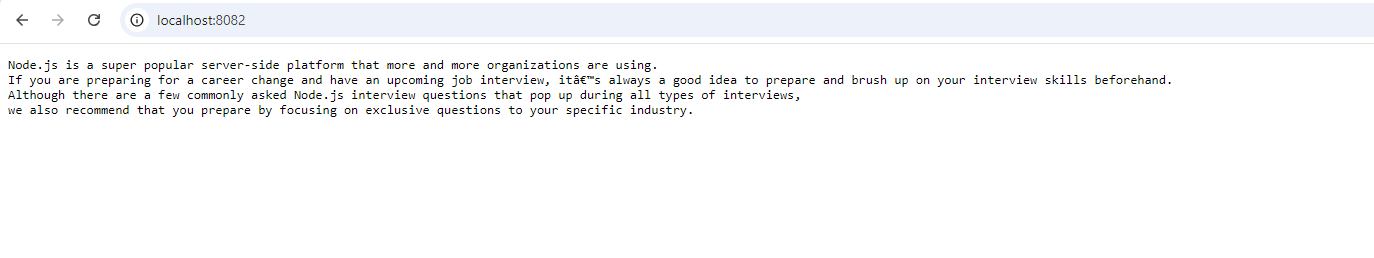
  res.end(data);

}).listen(8082, () => {

  console.log('Server is running at http://localhost:3000/');

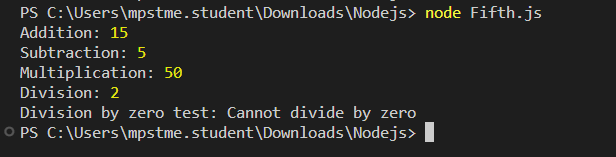
});

Output:



1. Write a Node.js program where you create and use at least two custom functions.
2. **Create a custom function** to:
   * Add two numbers and return the result.
   * Subtract two numbers and return the result.
3. **Create another function** to:
   * Multiply two numbers and return the result.
   * Divide two numbers and return the result (make sure to handle division by zero).
4. // Function to add and subtract two numbers
5. function addAndSubtract(num1, num2) {
6. const add = num1 + num2;
7. const subtract = num1 - num2;
8. return { add, subtract };
9. }
11. // Function to multiply and divide two numbers
12. function multiplyAndDivide(num1, num2) {
13. const multiply = num1 \* num2;
14. const divide = num2 !== 0 ? num1 / num2 : "Cannot divide by zero";
15. return { multiply, divide };
16. }
18. // Test the functions
19. const num1 = 10;
20. const num2 = 5;
22. const result1 = addAndSubtract(num1, num2);
23. console.log("Addition:", result1.add);
24. console.log("Subtraction:", result1.subtract);
26. const result2 = multiplyAndDivide(num1, num2);
27. console.log("Multiplication:", result2.multiply);
28. console.log("Division:", result2.divide);
30. // Test division by zero
31. const result3 = multiplyAndDivide(num1, 0);
32. console.log("Division by zero test:", result3.divide);

Output:



1. Write a Node.js program using the **HTTP module** that creates a web server. The server should respond with a different message for three routes:
2. /: Respond with **"Welcome to the Home Page"**.
3. /about: Respond with **"This is the About Page"**.
4. /contact: Respond with **"Feel free to contact us at contact@example.com"**.

For any other route, the server should return a **404** message with **"Page Not Found"**.

Code:

// Import the required HTTP module

const http = require('http');

// Create an HTTP server

const server = http.createServer((req, res) => {

    // Set the response header

    res.writeHead(200, { 'Content-Type': 'text/plain' });

    // Handle different routes

    switch (req.url) {

        case '/':

            res.end("Welcome to the Home Page");

            break;

        case '/about':

            res.end("This is the About Page");

            break;

        case '/contact':

            res.end("Feel free to contact us at contact@example.com");

            break;

        default:

            res.writeHead(404, { 'Content-Type': 'text/plain' });

            res.end("Page Not Found");

            break;

    }

});

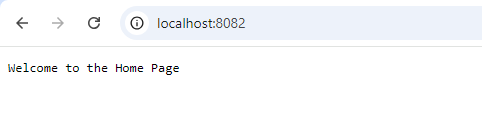
// Server listens on port 3000

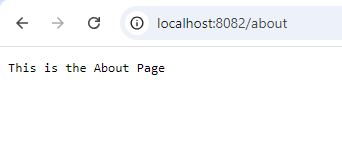
server.listen(8082, () => {

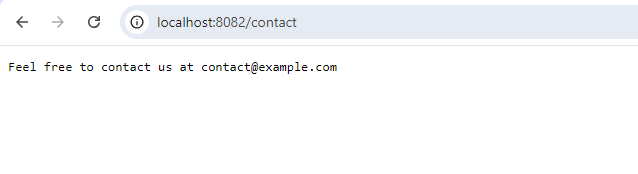
    console.log('Server is running at http://localhost:8082/');

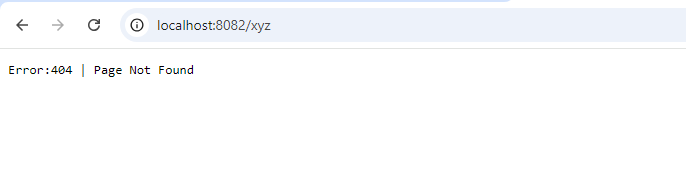
});

Output:









Conclusion :-

We studied Node.js and used the HTTP technique to connect to the server.

.