

Example

Write a function 'ArrayToObject' which takes in an array of arrays, and returns an object with each pair of elements in the array as a key-value pair and store the result in one arraytoobject.txt file.

Input=[['Country', 'India'], ['State', 'Gujarat'], ['City', 'Ahmedabad']]
Output= { Country : ' India ', State : ' Gujarat ', City : 'Ahmedabad' }

```
function ArrayToObject(a) {
  var temp = {};
  temp[a[0][0]] = a[0][1]; // First key-value pair
  temp[a[1][0]] = a[1][1]; // Second key-value pair
  temp[a[2][0]] = a[2][1]; // Third key-value pair
  return temp;
}

var data = [['Country', 'India'], ['State', 'Gujarat'], ['City', 'Ahmedabad']];
var output = ArrayToObject(data);
console.log(output);

// Store the output in a file
const fs = require('fs');
fs.writeFileSync('arraytoobject.txt', JSON.stringify(output));
```

Example

Write node.js script to print “Welcome to Home Page” with two links containing two pages named as “About Us” and “Contact Us” on home page of server. If user request for About Us page it should display “Welcome to LJ University” in bold font-style with blue color and if user request for Contact Us page it should display “Email:abc@ljinstitutes.edu.in” in italic font-style with red color if any other request is requested it shows “Page not found” message in plaintext.

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/html' });

  if (req.url === '/') {
```

```

res.end(`
  <h1>Welcome to Home Page</h1>
  <a href="/about">About Us</a> <a href="/contact">Contact Us</a>
`);
} else if (req.url === "/about") {
  res.end(`<h1 style="color: blue; font-weight: bold;">Welcome to LJ University</h1>`);
} else if (req.url === "/contact") {
  res.end(`<h2 style="color: red; font-style: italic;">Email: abc@ljinstitutes.edu.in</h2>`);
} else {
  res.writeHead(404, { 'Content-Type': 'text/plain' });
  res.end("Page not found");
}
});

server.listen(3000, () => {console.log("Server started");});

```

Example

Create HTTP webpages where Admin page displays “Sufficient Memory:” in bold blue color with font size of 24px along with available memory in GB with font size 32px and red color if available physical memory of the system is greater than 1 GB. Else it shows “Not Sufficient Memory” in simple text. For any other page requested then shows “You are not admin” message.

```

const http = require('http');
const os = require('os');

const server = http.createServer((req, res) => {
  if (req.url === "/admin") {
    const freeMem = os.freemem()/1024/1024/1024; // Convert bytes to GB

    if (freeMem > 1) {
      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.end(`
        <h1 style="color: blue; font-size: 24px; font-weight: bold;">Sufficient Memory:</h1>
        <h2 style="color: red; font-size: 32px;">${freeMem} GB</h2>
      `);
    } else {
      res.writeHead(200, { 'Content-Type': 'text/plain' });
      res.end("Not Sufficient Memory");
    }
  }
});

```

```

    }
  } else {
    res.writeHead(403, { 'Content-Type': 'text/plain' });
    res.end("You are not admin");
  }
});

server.listen(3000, () => { console.log("Server started");});

```

Example

Write a Node.js program for following action

- 1. Write a file having five random elements separated by white space in .txt file.**
- 2. append sorted array of these 5 elements in same file along with message : “Sorted array:” in new line.**
- 3. Find maximum number from that and append with message “maximum number=” in same file.**

```

const fs = require('fs');

fs.writeFileSync('num.txt', "20 30 13 7 22");
var d=fs.readFileSync("num.txt","utf-8");
d1= d.split(" ")

const sn = d1.sort((a, b) => a - b)
// Append sorted array message and sorted numbers
fs.appendFileSync('num.txt', `\nSorted array: ${sn}`);

const max = Math.max(...sn);
// Append maximum number message
fs.appendFileSync('num.txt', `\nmaximum number= ${max}`);

```

Example

Create HTTP webpage on which Home page display “Welcome to Log in page” in blue color and font size must be 32px, Login page shows one HTML file from static URL having Form with detail for Username, Password, submit and reset button, Gallery page reflect one Image “hello.jpg” and any other page shows “Page Not found”.

Write all necessary files to perform task. (Image already exist in same folder)

login.js

```

var h = require("http");

```

```
var url = require("url");
var fs = require("fs");
var addr="http://localhost:5051/login.html";
var server = h.createServer(function (req, res) {
  if (req.url == "/" ) {
    res.writeHead(200, { "Content-Type": "text/html" });
    res.write("<h1 style='color:blue;font-size:32px;'>Welcome to login page</h1>");
    res.end();
  } else if (req.url == "/login") {
    var q=url.parse(addr,true);
    data=fs.readFileSync("."+q.pathname);
    res.writeHead(200, { "Content-Type": "text/html" });
    res.write(data);
    res.end();
  } else if (req.url == "/gallery") {
    img = fs.readFileSync("1.png");
    res.writeHead(200, { "Content-Type": "image/png" });
    res.end(img); // Sending image data

  } else {
    res.writeHead(404, { "Content-Type": "text/plain" });
    res.write("Page not found");
    res.end("\nPlease check the URL");
  }
});

server.listen(5051)
```

login.html

```
<html>
<body>
  <h2>Login Form</h2>
  <form action="#" method="POST">
    <label for="username">Username:</label>
    <input type="text" required><br><br>

    <label for="password">Password:</label>
    <input type="password" required><br><br>

    <button type="submit">Submit</button>
    <button type="reset">Reset</button>
```

```
</form>
</body>
</html>
```

Example

Write node js script to fetch values from url given below and display output as asked.

"https://www.google.com/exam.txt?c1=Hello&c2=FSD2 T1 Test&c3=Welcome to LJU#AllTheBest"

1) Data must be written as below in file named "exam.txt". File name must be fetched from the url given above.

Output:

Hello!

Welcome to LJU FSD2 T1 Test

#AllTheBest

2) Read content from file "exam.txt" and send response to server and display data in "/" page in same format as above but in H1 tag and in red color.

3) If any other page is requested it shows "Page not found" message in plain text.

```
const http = require("http");
const fs = require("fs");
const url = require("url");

const url1 =
"https://www.google.com/exam1.txt?c1=Hello&c2=FSD2+T1+Test&c3=Welcome+to+LJU#AllTheBest";

const parsedUrl = url.parse(url1,true);
console.log(parsedUrl.pathname)

const c1 = parsedUrl.query.c1;
const c2 = parsedUrl.query.c2;
const c3 = parsedUrl.query.c3;
const hash = parsedUrl.hash;
const filename = "."+parsedUrl.pathname
```

```

const fileContent = c1+"!\n"+c3+" "+c2+"\n"+hash;

fs.writeFileSync(filename, fileContent);

const server = http.createServer((req, res) => {
  if (req.url === "/") {
    data= fs.readFileSync(filename, "utf-8")
    res.writeHead(500, { "Content-Type": "text/html" });
    res.end("<h1 style='color:red'><pre>"+data+"<pre></h1>");
  }
  else {
    res.writeHead(404, { "Content-Type": "text/plain" });
    res.end("Page not found");
  }
});

server.listen(3000)

```

Example

Write a nodeJS script to fire an event named calculate which calculates the total marks of 5 subjects about of 25 marks and displays the total marks on console as an output. The calculate event fires another event name percentage which takes total marks as argument and percentage should get displayed in console.

```

const EventEmitter = require("events");
const ee = new EventEmitter();

// Event to calculate total marks
ee.on("calculate", () => {
  const marks = [22, 18, 20, 25, 24];
  let total = 0;
  for (let i = 0; i < marks.length; i++) {
    total += marks[i];
  }
  console.log("Total Marks:", total);
  ee.emit("percentage", total);
});

ee.on("percentage", (total) => {
  const totalMaxMarks = 5 * 25;
  const percentage = (total / totalMaxMarks) * 100;
  console.log("Percentage:", percentage + "%");
});

```

```
});  
  
ee.emit("calculate");
```

Example

Write node js script designed for laptop having 6GB of RAM which require to monitor system memory usage and emit a custom event named “Threshold”. When the memory usage exceeds specified threshold of 50% there will be a message “Memory Threshold Exceeded” along with used data should be displayed on console after every 1 second of Interval.

```
const os = require("os");  
const EventEmitter = require("events");  
const eventEmitter = new EventEmitter();  
  
eventEmitter.on("threshold", () => {  
  setInterval(() => {  
    const tm = 6  
    const fm = os.freemem()/1024/1024/1024;  
    const um = tm - fm;  
    const up = (um / tm) * 100;  
    console.log(up)  
  
    if (up > 50) {  
      console.log("Used Memory"+um+" Memory threshold exceeded!");  
    }  
  }, 1000);  
});  
  
eventEmitter.emit("threshold");
```

Example

Write a node.js script using Event handling to perform following tasks in sequence:

- a) Create file in it named abc.txt and enter data into it.
- b) Append data to that file abc.txt and print message "Data Appended Successfully".
- c) Read the content of the file abc.txt and print the content on http web server.
- d) Do all the operations of File using asynchronous file system module. And Lastly print the message "All operations performed successfully" on console.

Example

Write a node.js script to write the text "This is data" to new.txt file. After that append the text "That is data" to same new.txt file. After that read the file & print the file content on console. After finishing read operation print the line "Thank you for using program". Write, append & read sequence must be maintained & all operations are asynchronous.

Example

Write a nodeJS script to print "My PC is on fire" on server at port no 5555 if having more than 4gb physical available memory otherwise,"I need more more memory" should get displayed.

Example

Write a node js script to write the text "You are creating a file" to help.txt file. After that append the text "you are appending data" to same help.txt file. After that read the file and print file contents on console. After finishing read operation , print the line "Thanks for using my program" on console. write ,append,read sequence must be maintain. all read ,write and append operations are asynchronous.