**1. What is Node.js?**

**Answer:**  
Node.js is a runtime environment that lets you run JavaScript code outside the browser. It uses the **V8 engine** (the same one used by Chrome) and is. commonly used for building **backend applications**, especially servers

**Example:** You can create a web server using just a few lines of JavaScript code with Node.js.

**2. Is Node.js single-threaded or multi-threaded?**

**Answer:**  
Node.js is **single-threaded** but it uses **event-driven and non-blocking I/O** operations to handle multiple requests at once, making it very fast and efficient.

**3. What is the difference between Node.js and JavaScript?**

**Answer:**

* JavaScript runs inside the browser (like Chrome).
* Node.js runs JavaScript **outside** the browser, on the **server side**.
* Node.js also provides extra features like file system access, HTTP modules, etc.

**4. What is the use of npm in Node.js?**

**Answer:**  
npm stands for **Node Package Manager**. It’s used to install packages (or libraries) in your Node.js project.

**Example:**

bash

CopyEdit

npm install express

This installs the Express framework into your project.

**5. What is a callback function in Node.js?**

**Answer:**  
A **callback** is a function passed as an argument to another function and is executed after the completion of that function.

**Example:**

js

CopyEdit

fs.readFile('file.txt', (err, data) => {

if (err) throw err;

console.log(data.toString());

});

**6. What is the difference between synchronous and asynchronous code in Node.js?**

**Answer:**

* **Synchronous**: Blocks the execution until the current operation finishes.
* **Asynchronous**: Doesn’t block. Moves to the next task and uses callbacks/promises to handle results later.

**7. What is the require function in Node.js?**

**Answer:**  
require is used to **import modules** in Node.js.

**Example:**

js

CopyEdit

const fs = require('fs');

This imports the File System module.

**8. What is module.exports in Node.js?**

**Answer:**  
It allows you to **export functions, objects, or variables** from a file so they can be used in another file.

**Example (math.js):**

js

CopyEdit

module.exports.add = (a, b) => a + b;

**Example (app.js):**

js

CopyEdit

const math = require('./math');

console.log(math.add(2, 3)); // 5

**9. What is middleware in Node.js (especially with Express)?**

**Answer:**  
Middleware is a function that has access to the **request and response objects**. It can modify them or end the request-response cycle.

**Example:**

js

CopyEdit

app.use((req, res, next) => {

console.log('Middleware executed');

next(); // Pass control to next middleware or route

});

**10. What is Express.js and why do we use it?**

**Answer:**  
Express.js is a **minimal and flexible Node.js framework** used to build **web applications and APIs** easily.

**Example:**

js

CopyEdit

const express = require('express');

const app = express();

app.get('/', (req, res) => res.send('Hello World'));

app.listen(3000);

**11. How do you create a simple server using Node.js?**

**Answer:**

js

CopyEdit

const http = require('http');

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

res.write('Hello from Node.js!');

res.end();

}).listen(3000);

**12. What are some core modules in Node.js?**

**Answer:**  
Some commonly used core modules are:

* fs (File System)
* http (Web Server)
* url (URL parsing)
* path (File path handling)
* events (Event handling)

**13. What is an event loop in Node.js?**

**Answer:**  
The **event loop** handles all asynchronous callbacks. It allows Node.js to perform non-blocking I/O operations, even though JavaScript is single-threaded.

**14. What is an event emitter in Node.js?**

**Answer:** It’s a class in the events module. It allows you to create, fire, and listen to events.

**Example:**

js

CopyEdit

const EventEmitter = require('events');

const emitter = new EventEmitter();

emitter.on('greet', () => {

console.log('Hello from event!');

});

emitter.emit('greet');

**15. What is the difference between readFileSync and readFile?**

**Answer:**

* readFileSync: Synchronous, blocks execution.
* readFile: Asynchronous, doesn’t block.

Prefer readFile for better performance.

**16. What are streams in Node.js?**

**Answer:**  
Streams are used to handle **continuous data**. Useful for reading/writing **large files** or **video streaming**.

There are 4 types:

1. Readable
2. Writable
3. Duplex (both)
4. Transform (modifies data)

**17. How does Node.js handle child processes?**

**Answer:**  
Using the child\_process module. It allows you to run system commands.

**Example:**

js

CopyEdit

const { exec } = require('child\_process');

exec('ls', (err, stdout) => {

console.log(stdout);

});

**18. What are Promises in Node.js?**

**Answer:**  
Promises are used to handle **asynchronous operations**. They represent a value that may be available now or in the future.

**Example:**

js

CopyEdit

let promise = new Promise((resolve, reject) => {

resolve("Success");

});

promise.then(console.log);

**19. What is the use of async/await in Node.js?**

**Answer:**  
async/await is syntax sugar for promises. It makes asynchronous code look synchronous and easier to read .

**Example:**

js

CopyEdit

async function getData() {

let result = await fetchData();

console.log(result);

}

**20. What is the difference between process.nextTick() and setImmediate()?**

**Answer:**

* process.nextTick(): Runs before the next event loop tick.
* setImmediate(): Runs on the next event loop tick.