

# Node.js Interview Questions and Answers for Freshers

---

## ◆ Basic Node.js Interview Questions

### 1. What is Node.js?

Node.js is a runtime environment that allows you to execute JavaScript code on the server side. It is built on Chrome's V8 JavaScript engine.

### 2. How is Node.js different from JavaScript in the browser?

In the browser, JavaScript is used for client-side scripting. Node.js enables JavaScript to be run server-side, allowing for backend development.

### 3. What are the features of Node.js?

- Asynchronous and Event Driven
- Very fast due to V8 engine
- Single-threaded but highly scalable
- No buffering

### 4. What is the use of `npm`?

`npm` (Node Package Manager) is used to manage packages (libraries or modules) in a Node.js project.

### 5. What are modules in Node.js?

Modules are reusable JavaScript files that can be imported into other files using `require()`.

### 6. What is the difference between CommonJS and ES6 modules?

- CommonJS uses `require()` and `module.exports`
- ES6 uses `import` and `export`

### 7. What is `package.json`?

It is the configuration file that holds metadata about the project and the dependencies.

### 8. What is the role of the `require` function?

`require` is used to include modules in a Node.js application.

---

## ◆ Core Concepts

### 9. What is the event loop in Node.js?

The event loop allows Node.js to perform non-blocking I/O operations using a single thread.

### 10. Explain the callback function.

A callback is a function passed into another function as an argument, which is then executed after the parent function completes.

### 11. What are Promises and async/await?

Promises represent the completion of an asynchronous operation. `async/await` provides a way to write asynchronous code in a synchronous manner.

### 12. Difference between synchronous and asynchronous programming

- Synchronous: Code runs sequentially.
- Asynchronous: Code can execute while waiting for previous operations to complete.

### 13. What is a middleware in Node.js?

Middleware functions are functions that have access to `req`, `res`, and `next` in the application's request-response cycle.

### 14. Explain streams in Node.js.

Streams are objects that allow reading or writing data in a continuous fashion.

### 15. What is a buffer in Node.js?

Buffers are used to handle binary data in Node.js when working with file streams or network protocols.

---

## ◆ Node.js with Express.js

### 16. What is Express.js?

Express.js is a minimal and flexible Node.js web application framework.

### 17. How do you create a simple REST API using Express.js?

Use `express.Router()` to define endpoints and `app.listen()` to start the server.

## 18. What are the HTTP methods?

- GET
- POST
- PUT
- DELETE

## 19. What is `req` and `res` in Express?

- `req`: Request object containing data about the HTTP request.
- `res`: Response object used to send the HTTP response.

## 20. How do you handle routing in Express.js?

Using `app.get()`, `app.post()`, etc., to define routes.

## 21. What is CORS and how to enable it?

CORS (Cross-Origin Resource Sharing) is a security feature. Use the `cors` package to enable it in Express.

## 22. What is body-parser and why is it used?

`body-parser` is a middleware used to parse incoming request bodies.

---

## ◆ File System and OS Module

### 23. How to read and write files in Node.js?

Use the `fs` module: `fs.readFile()`, `fs.writeFile()`.

### 24. How do you create a directory in Node.js?

Use `fs.mkdir()` or `fs.mkdirSync()`.

### 25. What is the use of the `fs` and `os` modules?

- `fs`: Interact with the file system.
  - `os`: Get information about the system (CPU, memory, etc).
- 

## ◆ Error Handling and Debugging

### 26. How do you handle errors in Node.js?

Using `try...catch` blocks or error-first callback functions (`err` as the first argument).

### 27. What is the difference between `process.exit()` and `throw`?

- `process.exit()`: Immediately terminates the Node.js process.
- `throw`: Throws an exception that can be caught by `try...catch`.

### 28. What is the use of `try...catch` with `async/await`?

To handle errors in asynchronous code when using `await`.

---

## ◆ Miscellaneous

### 29. What is `process` in Node.js?

`process` is a global object that provides information and control over the current Node.js process.

### 30. What is the use of environment variables?

They are used to configure the application outside of the code (e.g., database URLs, API keys).

### 31. What is clustering in Node.js?

Clustering allows you to run multiple Node.js processes to handle the load of a single application.

### 32. How do you secure a Node.js app?

- Use `helmet` middleware
- Enable HTTPS
- Sanitize inputs to prevent injection attacks

### 33. What are some common Node.js libraries?

- Express
  - Mongoose
  - Axios
  - Lodash
  - Dotenv
-