1. What does a client do in a client-server architecture?

  - A) Processes requests

  - B) Sends requests to the server

  - C) Stores data

  - D) Manages connections

  Answer: B) Sends requests to the server

2. In a client-server architecture, what role does the server play?

  - A) Initiates communication

  - B) Processes requests and provides resources

  - C) Manages user interface

  - D) Handles user inputs

  Answer: B) Processes requests and provides resources

3. Which of the following is a common protocol used in client-server communication?

  - A) FTP

  - B) HTTP

  - C) SMTP

  - D) POP3

  Answer: B) HTTP

4. What is the primary purpose of HTTP status codes?

  - A) To determine the type of data

  - B) To indicate the outcome of a request

  - C) To format data

  - D) To encrypt data

Answer: B) To indicate the outcome of a request

5. Which HTTP status code indicates that the request was successful?

 - A) 404

 - B) 500

 - C) 200

 - D) 301

Answer: C) 200

6. Which HTTP status code is used when a resource has been moved permanently?

 - A) 200

 - B) 301

 - C) 403

 - D) 500

Answer: B) 301

7. What does a 404 HTTP status code indicate?

 - A) Unauthorized access

 - B) Resource not found

 - C) Request timeout

 - D) Internal server error

Answer: B) Resource not found

8. In a request-response model, who initiates the request?

 - A) The server

 - B) The client

 - C) The database

 - D) The network

Answer: B) The client

9. Which of the following is a characteristic of a stateless protocol like HTTP?

   - A) It maintains session state between requests

   - B) Each request is independent and contains all necessary information

   - C) It uses sessions to track user activity

   - D) It encrypts data between client and server

   Answer: B) Each request is independent and contains all necessary information

10. What does the term "server-side scripting" refer to?

   - A) Code executed on the client's browser

   - B) Code executed on the server

   - C) Code executed by the database

   - D) Code executed by the network

   Answer: B) Code executed on the server

11. What is Node.js built on?

   - A) JavaScript Engine

   - B) Python Engine

   - C) Java Engine

   - D) Ruby Engine

   Answer: A) JavaScript Engine

12. Which of the following is a feature of Node.js?

   - A) Multi-threaded processing

   - B) Single-threaded event loop

- C) Synchronous I/O model

- D) Client-side rendering

Answer: B) Single-threaded event loop

13. What is the primary use case for Node.js?

- A) Creating client-side applications

- B) Building server-side applications

- C) Managing databases

- D) Designing user interfaces

Answer: B) Building server-side applications

14. Which module in Node.js is used to create an HTTP server?

- A) `http`

- B) `fs`

- C) `path`

- D) `url`

Answer: A) `http`

15. How does Node.js handle asynchronous operations?

- A) By using threads

- B) By blocking the main thread

- C) By using callbacks and promises

- D) By using synchronous operations

Answer: C) By using callbacks and promises

16. What is the purpose of npm in Node.js?

- A) To compile JavaScript code

- B) To manage packages and dependencies

- C) To debug code

- D) To create web servers

Answer: B) To manage packages and dependencies

17. Which method is used to include modules in a Node.js application?

- A) `require()`

- B) `import`

- C) `include`

- D) `load`

Answer: A) `require()`

18. What does `npm install` do?

- A) Uninstalls a package

- B) Installs a package from the npm registry

- C) Updates npm itself

- D) Creates a new Node.js application

Answer: B) Installs a package from the npm registry

19. What is the default file extension for Node.js modules?

- A) `.js`

- B) `.json`

- C) `.node`

- D) `.ts`

Answer: A) `.js`

20. Which of the following is NOT a core module in Node.js?

- A) `http`

- B) `fs`

- C) `express`

- D) `path`

Answer: C) `express`

Advantages and Disadvantages of Node.js

21. What is one of the main advantages of using Node.js for server-side programming?

   - A) Multi-threading support

   - B) Blocking I/O operations

   - C) Non-blocking I/O operations

   - D) Lack of package management

   Answer: C) Non-blocking I/O operations

22. Which of the following is a disadvantage of Node.js?

   - A) Large community

   - B) Single-threaded event loop

   - C) High performance

   - D) Asynchronous programming model

   Answer: B) Single-threaded event loop

23. Why might Node.js be preferred for real-time applications?

   - A) It uses a synchronous I/O model

   - B) It supports non-blocking operations

   - C) It is primarily a client-side technology

   - D) It does not support event-driven programming

Answer: B) It supports non-blocking operations

24. Which programming paradigm is Node.js particularly well-suited for?

   - A) Object-oriented programming

   - B) Functional programming

   - C) Event-driven programming

   - D) Declarative programming

   Answer: C) Event-driven programming

25. What is a common challenge when using Node.js for CPU-intensive tasks?

   - A) It uses a multi-threaded model

   - B) It may block the event loop

   - C) It supports multi-core processors

   - D) It has a built-in debugger

   Answer: B) It may block the event loop

26. Which command is used to check the installed version of Node.js?

   - A) `node -v`

   - B) `node --version`

   - C) `npm -v`

   - D) `npm version`

   Answer: A) `node -v`

27. Which file is created by running `npm init`?

   - A) `package.json`

   - B) `node_modules`

   - C) `index.js`

   - D) `server.js`

Answer: A) `package.json`

28. How do you start a Node.js server with a file named `server.js`?

   - A) `node server.js`

   - B) `npm start server.js`

   - C) `node start server.js`

   - D) `npm run server.js`

   Answer: A) `node server.js`

29. Which method is used to create an HTTP server in Node.js?

   - A) `http.createServer()`

   - B) `http.startServer()`

   - C) `http.listen()`

   - D) `http.runServer()`

   Answer: A) `http.createServer()`

30. What is the purpose of the `listen` method in an HTTP server?

   - A) To send a response to the client

   - B) To start the server and listen for incoming connections

   - C) To handle errors

   - D) To terminate the server

   Answer: B) To start the server and listen for incoming connections

31. In Express, which method is used to handle GET requests?

   - A) `app.get()`

   - B) `app.post()`

- C) `app.put()`

- D) `app.delete()`

Answer: A) `app.get()`

32. How do you define a route in Express that responds to a POST request?

  - A) `app.post('/path', callback)`

  - B) `app.get('/path', callback)`

  - C) `app.put

('/path', callback)`

  - D) `app.delete('/path', callback)`

Answer: A) `app.post('/path', callback)`

33. Which method in Express is used to send a JSON response?

  - A) `res.json()`

  - B) `res.send()`

  - C) `res.end()`

  - D) `res.write()`

Answer: A) `res.json()`

34. What is middleware in Express?

  - A) A tool for debugging

  - B) Code executed before the request reaches the route handler

  - C) A way to handle asynchronous operations

  - D) A method to start the server

Answer: B) Code executed before the request reaches the route handler

35. Which function is used to handle errors in an Express application?

   - A) `app.use()`

   - B) `app.error()`

   - C) `app.handle()`

   - D) `app.exception()`

   Answer: A) `app.use()`

36. What is the purpose of `module.exports` in Node.js?

   - A) To import a module

   - B) To define a module's interface for exporting

   - C) To delete a module

   - D) To list all modules

   Answer: B) To define a module's interface for exporting

37. Which command is used to install a package globally using npm?

   - A) `npm install -g package_name`

   - B) `npm install package_name --global`

   - C) `npm global install package_name`

   - D) `npm add package_name -g`

   Answer: A) `npm install -g package_name`

38. How do you import a module in a Node.js application?

   - A) `import module from 'module'`

   - B) `require('module')`

   - C) `include('module')`

   - D) `load('module')`

Answer: B) `require('module')`

39. What does the `npm start` command do?

   - A) Starts the Node.js application defined in the `package.json` scripts

   - B) Installs all dependencies

   - C) Creates a new Node.js project

   - D) Updates npm to the latest version

   Answer: A) Starts the Node.js application defined in the `package.json` scripts

40. What is a package.json file used for?

   - A) To manage application dependencies and scripts

   - B) To execute Node.js scripts

   - C) To configure the Node.js runtime

   - D) To store static assets

   Answer: A) To manage application dependencies and scripts

Handling Static Pages with File Stream

41. Which Node.js module is used to handle file system operations?

   - A) `fs`

   - B) `http`

   - C) `path`

   - D) `url`

   Answer: A) `fs`

42. What does the `fs.createReadStream` method do?

   - A) Creates a writable stream for writing data to a file

   - B) Creates a readable stream for reading data from a file

- C) Creates a directory stream

- D) Creates a file system watcher

Answer: B) Creates a readable stream for reading data from a file

43. How do you serve static files in an Express application?

   - A) `app.use(express.static('public'))`

   - B) `app.get('/static', express.static('public'))`

   - C) `app.serve('public')`

   - D) `app.static('public')`

   Answer: A) `app.use(express.static('public'))`

44. What is the purpose of `fs.createWriteStream`?

   - A) To create a readable stream

   - B) To write data to a file

   - C) To read data from a file

   - D) To watch file changes

   Answer: B) To write data to a file

45. Which method is used to append data to a file in Node.js?

   - A) `fs.appendFile()`

   - B) `fs.writeFile()`

   - C) `fs.createReadStream()`

   - D) `fs.createWriteStream()`

   Answer: A) `fs.appendFile()`

46. Which event is triggered in Node.js when an exception is not caught?

   - A) `error`

- B) `exception`

- C) `warning`

- D) `crash`

Answer: A) `error`

47. How can you handle exceptions in a Node.js application?

   - A) Using `try-catch` blocks

   - B) Using `if-else` conditions

   - C) Using `try-finally` blocks

   - D) Using `error-handling` middleware

   Answer: A) Using `try-catch` blocks

48. Which method is used to handle uncaught exceptions in a Node.js process?

   - A) `process.on('uncaughtException')`

   - B) `process.catch()`

   - C) `process.handle()`

   - D) `process.error()`

   Answer: A) `process.on('uncaughtException')`

49. What is the purpose of the `catch` method in Promises?

   - A) To handle successful asynchronous operations

   - B) To handle errors in asynchronous operations

   - C) To chain multiple Promises

   - D) To create new Promises

   Answer: B) To handle errors in asynchronous operations

50. Which of the following is a good practice for handling exceptions in a Node.js server?

- A) Ignoring exceptions

  - B) Logging exceptions and sending a generic error message to the client

  - C) Exposing detailed stack traces to the client

  - D) Restarting the server on each exception

  Answer: B) Logging exceptions and sending a generic error message to the client

Additional Questions on Node.js Concepts

51. What is the main advantage of using Node.js for building APIs?

  - A) Multi-threaded architecture

  - B) High performance with non-blocking I/O

  - C) Built-in database support

  - D) Synchronous processing

  Answer: B) High performance with non-blocking I/O

52. Which method is used to read the contents of a file synchronously in Node.js?

  - A) `fs.readFileSync()`

  - B) `fs.readSync()`

  - C) `fs.read()`

  - D) `fs.fileSync()`

  Answer: A) `fs.readFileSync()`

53. What is the purpose of the `res.send()` method in Express?

  - A) To send a JSON response

  - B) To send a file response

  - C) To send a response to the client

  - D) To send a redirect response

Answer: C) To send a response to the client

54. Which module in Node.js is used for URL manipulation?

   - A) `url`

   - B) `http`

   - C) `fs`

   - D) `path`

   Answer: A) `url`

55. What is a common use case for using streams in Node.js?

   - A) Processing large files

   - B) Managing user sessions

   - C) Encrypting data

   - D) Handling web requests

   Answer: A) Processing large files

56. How do you create a writable stream for a file in Node.js?

   - A) `fs.createWriteStream()`

   - B) `fs.createStream()`

   - C) `fs.openStream()`

   - D) `fs.writeStream()`

   Answer: A) `fs.createWriteStream()`

57. Which method is used to handle HTTP requests and responses in an Express application?

   - A) `app.handle()`

   - B) `app.process()`

   - C) `app.request()`

   - D) `app.use()`

Answer: D) `app.use()`

58. Which of the following is NOT a feature of Node.js?

  - A) Asynchronous programming

  - B) Multi-threaded execution

  - C) Non-blocking I/O

  - D) Event-driven architecture

  Answer: B) Multi-threaded execution

59. What is the purpose of the `fs.unlink()` method?

  - A) To delete a file

  - B) To create a file

  - C) To append data to a file

  - D) To rename a file

  Answer: A) To delete a file

60. Which command initializes a new Node.js project with npm?

  - A) `npm init`

  - B) `npm start`

  - C) `npm install`

  - D) `npm create`

  Answer: A) `npm init`

61. What is the main purpose of the `fs` module in Node.js?

  - A) To handle file system operations

  - B) To manage HTTP requests

- C) To process URL paths

- D) To create databases

Answer: A) To handle file system operations

62. Which of the following methods is used to read a file asynchronously?

- A) `fs.readFile()`

- B) `fs.openFile()`

- C) `fs.read()`

- D) `fs.getFile()`

Answer: A) `fs.readFile()`

63. What is a common use of the `fs.createReadStream()` method?

- A) Writing data to a file

- B) Reading data from a file in chunks

- C) Deleting a file

- D) Creating a new file

Answer: B) Reading data from a file in chunks

64. How can you handle static files in an Express application?

- A) Using `express.static` middleware

- B) Using `app.static` method

- C) Using `app.files` method

- D) Using `express.files` middleware

Answer: A) Using `express.static` middleware

65. Which method is used to end a writable stream in Node.js?

- A) `end()`

- B) `close()`

- C) `finish()`

- D) `stop()`

Answer: A) `end()`

66. What does the `fs.readdir()` method do?

  - A) Reads the contents of a directory

  - B) Reads a file's contents

  - C) Deletes a file

  - D) Writes data to a file

  Answer: A) Reads the contents of a directory

67. How do you handle exceptions thrown by asynchronous code in Node.js?

  - A) Using `try-catch` blocks

  - B) Using `catch` method of Promises

  - C) Using synchronous code only

  - D) Ignoring the exceptions

  Answer: B) Using `catch` method of Promises

68. Which module allows you to work with URL paths in Node.js?

  - A) `url`

  - B) `path`

  - C) `fs`

  - D) `http`

  Answer: B) `path`

69. What is the use of `fs.stat()` method?

- A) To get information about a file or directory

- B) To write data to a file

- C) To read data from a file

- D) To delete a file

Answer: A) To get information about a file or directory

70. Which method in Express is used to handle HTTP PUT requests?

- A) `app.put()`

- B) `app.post()`

- C) `app.get()`

- D) `app.delete()`

Answer: A) `app.put()`

Advanced Concepts and Practices

71. What is the purpose of the `eventEmitter` class in Node.js?

- A) To handle file system events

- B) To manage HTTP requests

- C) To handle and emit events

- D) To perform database operations

Answer: C) To handle and emit events

72. Which module is used to create and manage child processes in Node.js?

- A) `child_process`

- B) `process`

- C) `os`

- D) `cluster`

Answer: A) `child_process`

73. What is the role of the `cluster` module in Node.js?

   - A) To manage multiple threads

   - B) To create and manage multiple processes

   - C) To handle file system operations

   - D) To manage network connections

   Answer: B) To create and manage multiple processes

74. Which method is used to start a Node.js HTTP server on a specific port?

   - A) `server.listen()`

   - B) `server.start()`

   - C) `server.run()`

   - D) `server.open()`

   Answer: A) `server.listen()`

75. What is the purpose of `process.env` in Node.js?

   - A) To access environment variables

   - B) To handle file operations

   - C) To manage HTTP requests

   - D) To execute child processes

   Answer: A) To access environment variables

76. What is the default port number for HTTP servers in Node.js?

   - A) 80

   - B) 8080

   - C) 3000

   - D) 5000

Answer: A) 80

77. Which method is used to set up a server to listen on a specific port and host?

   - A) `server.listen(port, host)`

   - B) `server.start(port, host)`

   - C) `server.open(port, host)`

   - D) `server.bind(port, host)`

   Answer: A) `server.listen(port, host)`

78. What is a common pattern for handling asynchronous operations in Node.js?

   - A) Callbacks

   - B) Synchronous code

   - C) Blocking I/O

   - D) Threads

   Answer: A) Callbacks

79. Which of the following tools can be used for debugging Node.js applications?

   - A) `node --inspect`

   - B) `npm debug`

   - C) `node-debug`

   - D) `inspect`

   Answer: A) `node --inspect`

80. Which Node.js feature allows you to define and use custom events?

   - A) `EventEmitter` class

   - B) `http` module

   - C) `fs` module

- D) `process` module

Answer: A) `EventEmitter` class

81. What is EJS used for?

  - A) Creating static HTML pages

  - B) Creating dynamic HTML pages with embedded JavaScript

  - C) Managing databases

  - D) Compiling JavaScript code

  Answer: B) Creating dynamic HTML pages with embedded JavaScript

82. Which syntax is used to embed variables in an EJS template?

  - A) `<%= variable %>`

  - B) `{{ variable }}`

  - C) `{{= variable}}`

  - D) `<% variable %>`

  Answer: A) `<%= variable %>`

83. How do you include another EJS template in a parent EJS template?

  - A) `<% include 'filename.ejs' %>`

  - B) `<% import 'filename.ejs' %>`

  - C) `<% require 'filename.ejs' %>`

  - D) `<% attach 'filename.ejs' %>`

  Answer: A) `<% include 'filename.ejs' %>`

84. Which of the following is used to render an EJS template in Express?

  - A) `res.render()`

  - B) `res.send()`

- C) `res.view()`

- D) `res.template()`

Answer: A) `res.render()`

85. Which EJS tag is used for control flow logic, such as if statements?

- A) `<% %>`

- B) `<%= %>`

- C) `<%-- %>`

- D) `<%# %>`

Answer: A) `<% %>`

86. How do you render a view template with data in Express using EJS?

- A) `res.render('viewName', { data: data })`

- B) `res.view('viewName', { data: data })`

- C) `res.send('viewName', { data: data })`

- D) `res.template('viewName', { data: data })`

Answer: A) `res.render('viewName', { data: data })`

87. What is the purpose of the `<%- %>` tag in EJS?

- A) To output raw HTML without escaping

- B) To escape HTML characters

- C) To include external scripts

- D) To handle form submissions

Answer: A) To output raw HTML without escaping

88. Which EJS tag is used to output data as HTML, without escaping?

- A) `<%- %>`

- B) `<%= %>`

- C) `<% %>`

- D) `<%# %>`

Answer: A) `<%- %>`

89. How can you pass multiple variables to an EJS template from an Express route?

   - A) By including them in an object passed to `res.render()`

   - B) By creating a global context

   - C) By using query parameters

   - D) By setting environment variables

   Answer: A) By including them in an object passed to `res.render()`

90. Which of the following is NOT a feature of EJS?

   - A) Template inheritance

   - B) HTML escaping

   - C) Asynchronous templates

   - D) Dynamic JavaScript evaluation

   Answer: C) Asynchronous templates

91. What does the `npm run` command do?

   - A) Executes a script defined in the `package.json` file

   - B) Installs a package

   - C) Starts the Node.js application

   - D) Updates npm to the latest version

   Answer: A) Executes a script defined in the `package.json` file

92. How do you handle a GET request in Express?

   - A) `app.get('/path', callback)`

   - B) `app.request('GET', '/path', callback)`

   - C) `app.use('/path', callback)`

   - D) `app.post('/path', callback)`

   Answer: A) `app.get('/path', callback)`

93. Which command initializes a new Git repository?

   - A) `git init`

   - B) `git start`

   - C) `git new`

   - D) `git create`

   Answer: A) `git init`

94. What is the purpose of the `.env` file in a Node.js project?

   - A) To store environment variables

   - B) To manage dependencies

   - C) To configure the application

   - D) To handle HTTP requests

   Answer: A) To store environment variables

95. Which method is used to handle HTTP DELETE requests in Express?

   - A) `app.delete()`

   - B) `app.remove()`

   - C) `app.get()`

   - D) `app.put()`

   Answer: A) `app.delete()`

96. How do you handle POST data in Express?

   - A) By using `body-parser` middleware

   - B) By using `query-parser` middleware

   - C) By using `url-parser` middleware

   - D) By using `data-parser` middleware

   Answer: A) By using `body-parser` middleware

97. What is the purpose of the `npm cache clean` command?

   - A) To clear the npm cache

   - B) To install packages

   - C) To initialize a new project

   - D) To update npm

   Answer: A) To clear the npm cache

98. Which method is used to make HTTP requests in Node.js?

   - A) `http.request()`

   - B) `http.get()`

   - C) `http.post()`

   - D) `http.send()`

   Answer: A) `http.request()`

99. What is the purpose of the `res.redirect()` method in Express?

   - A) To redirect the client to a different URL

   - B) To render a view

   - C) To send a file response

   - D) To end the response

Answer: A) To redirect the client to a different URL

100. Which command is used to check the version of npm installed?

- A) `npm -v`

- B) `npm version`

- C) `npm check`

- D) `npm info`

Answer: A) `npm -v`