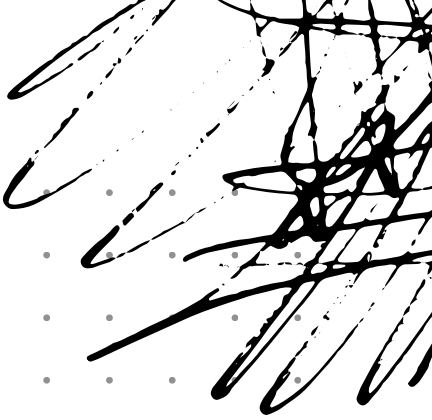
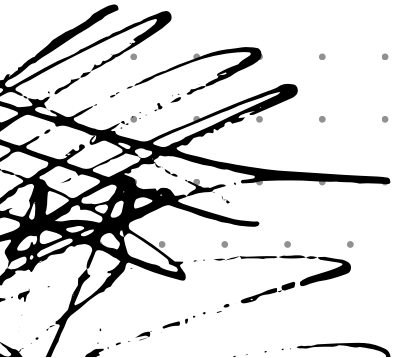


NODEJS QUESTIONS

TOP-100

BASIC TO ADVANCE ONE-LINER QUESTIONS



1. **What is Node.js?**
   Node.js is a runtime environment that executes JavaScript code on the server-side.

2. **What is npm?**
   npm (Node Package Manager) is a package manager for installing and managing Node.js packages/modules.

3. **Explain the event loop in Node.js.**
   The event loop is a core concept in Node.js that handles asynchronous operations by executing callbacks when events occur.

4. **What is a callback function in Node.js?**
   A callback function is a function passed as an argument to another function, often used for asynchronous operations.

5. **What are Promises in Node.js?**
   Promises are a way to handle asynchronous operations and provide a cleaner alternative to callbacks.

6. **What is the purpose of the `require` function in Node.js?**
   The `require` function is used to include modules in Node.js files.

7. **Explain the concept of middleware in Express.js.**
   Middleware are functions that process requests before they reach the route handlers in Express.js.

8. **What is the purpose of the `next` function in Express.js middleware?**
   The `next` function is used to pass control to the next middleware in the chain.

9. **What is the `module.exports` object in Node.js?**
   `module.exports` is used to export objects, functions, or values from a module.

10. **Explain the difference between `require` and `import` in Node.js.**
    `require` is used in CommonJS modules, while `import` is used in ES6 modules.

11. **What is the `process` object in Node.js?**
    The `process` object provides information about the Node.js process, environment variables, and more.

12. **What is the purpose of the `Buffer` class in Node.js?**
    The `Buffer` class is used to manipulate binary data, often used for handling streams and files.

13. **What is the Event Emitter in Node.js?**
    The Event Emitter is a built-in module that allows you to handle and emit events in Node.js applications.

14. **How does Node.js handle concurrency?**
    Node.js uses an event-driven, non-blocking I/O model to handle concurrency efficiently.

15. **Explain the concept of clustering in Node.js.**
    Clustering involves creating multiple instances (workers) of a Node.js process to utilize multiple CPU cores.

16. **What is the purpose of the `global` object in Node.js?**
    The `global` object provides access to global variables and functions across different modules.

17. **What are streams in Node.js?**
    Streams are objects that allow you to read or write data piece by piece, making them efficient for handling large amounts of data.

18. **What is the `__dirname` and `__filename` variables in Node.js?**
    `__dirname` holds the current directory name, and `__filename` holds the current file's absolute path.

19. **What is the purpose of the `fs` module in Node.js?**
    The `fs` module provides file system-related functionalities for reading, writing, and manipulating files.

20. **Explain the role of the `path` module in Node.js.**
    The `path` module is used for working with file and directory paths in a cross-platform way.

21. **What is the role of the `http` module in Node.js?**
    The `http` module provides functionalities to create an HTTP server and handle HTTP requests and responses.

22. **Explain the concept of the `setInterval` function in Node.js.**
    `setInterval` is used to repeatedly execute a function with a specified time interval.

23. **What are child processes in Node.js?**
    Child processes are spawned by the main Node.js process and allow you to execute external commands and scripts.

24. **What is the purpose of the `os` module in Node.js?**
    The `os` module provides information about the operating system, like CPU architecture and memory.

25. **Explain the concept of the `crypto` module in Node.js.**
    The `crypto` module is used for cryptographic functionalities, like creating secure hashes and encryption.

26. **What is the significance of the `net` module in Node.js?**
    The `net` module provides a way to create TCP servers and clients for network communication.

27. **What is the purpose of the `url` module in Node.js?**
    The `url` module provides utilities for working with URLs, like parsing and formatting.

28. **Explain the concept of the `cluster` module in Node.js.**
    The `cluster` module allows you to create multiple child processes that share server ports to utilize multiple cores.

29. **What is the role of the `querystring` module in Node.js?**
    The `querystring` module provides utilities for working with query strings in URLs.

30. **What is the purpose of the `util` module in Node.js?**
    The `util` module provides various utility functions for formatting and inspecting values.

31. **Explain the role of the `assert` module in Node.js.**
    The `assert` module

provides assertion functions for testing conditions and throwing errors if they fail.

32. **What are environment variables in Node.js?**
    Environment variables are values set in the system environment that can be accessed by Node.js applications.

33. **Explain the concept of the `zlib` module in Node.js.**
    The `zlib` module provides compression and decompression functionalities for files and streams.

34. **What is the purpose of the `async` module in Node.js?**
    The `async` module provides utilities for handling asynchronous operations, like managing callbacks and control flow.

35. **Explain the concept of the `timers` module in Node.js.**
    The `timers` module provides functions to create and manage timers for executing code at specific intervals.

36. **What is the role of the `dns` module in Node.js?**
    The `dns` module provides functions to perform DNS (Domain Name System) lookups.

37. **What is the purpose of the `os` module in Node.js?**
    The `os` module provides information about the operating system, like CPU architecture and memory.

38. **Explain the concept of the `readline` module in Node.js.**
    The `readline` module provides an interface for reading input from readable streams, like the command line.

39. **What is the role of the `util` module in Node.js?**
    The `util` module provides various utility functions for formatting and inspecting values.

40. **What is the purpose of the `events` module in Node.js?**
    The `events` module provides a way to handle and emit custom events in Node.js applications.

41. **Explain the concept of the `child_process` module in Node.js.**
    The `child_process` module allows you to create and manage child processes for executing external commands and scripts.

42. **What is the role of the `https` module in Node.js?**
    The `https` module provides functionalities to create an HTTPS server and handle HTTPS requests and responses.

43. **What is the purpose of the `stream` module in Node.js?**
    The `stream` module provides a way to work with streaming data, like reading and writing large files.

44. **Explain the concept of the `console` object in Node.js.**
    The `console` object provides methods for logging information, warnings, and errors to the console.

45. **What is the role of the `assert` module in Node.js?**
    The `assert` module provides assertion functions for testing conditions and throwing errors if they fail.

46. **What is the purpose of the `path` module in Node.js?**
   The `path` module is used for working with file and directory paths in a cross-platform way.

47. **Explain the concept of the `url` module in Node.js.**
   The `url` module provides utilities for working with URLs, like parsing and formatting.

48. **What is the role of the `dns` module in Node.js?**
   The `dns` module provides functions to perform DNS (Domain Name System) lookups.

49. **What is the purpose of the `cluster` module in Node.js?**
   The `cluster` module allows you to create multiple child processes that share server ports to utilize multiple cores.

50. **Explain the concept of the `querystring` module in Node.js.**
   The `querystring` module provides utilities for working with query strings in URLs.

51. **What are the benefits of using Node.js?**
    Node.js offers fast, scalable, and non-blocking I/O capabilities, making it ideal for building real-time applications and APIs.

52. **Explain the concept of the CommonJS module system.**
   CommonJS is a module system used in Node.js for organizing and structuring code into separate reusable modules.

53. **What is the purpose of the `process.argv` property in Node.js?**
   `process.argv` holds an array of command-line arguments provided to the Node.js process.

54. **Explain the concept of middleware in the context of Express.js.**
     Middleware functions in Express.js intercept and process HTTP requests and responses, performing tasks like authentication and logging.

55. **What is a package.json file in Node.js?**
    `package.json` is a metadata file that contains information about a Node.js project, including dependencies and scripts.

56. **What are streams in the context of file handling in Node.js?**
   Streams are used for reading or writing data in chunks, making them efficient for handling large files.

57. **What is the `global` object in Node.js?**
   The `global` object in Node.js contains variables and functions that are available globally across modules.

58. **Explain the role of the `os` module in Node.js.**
    The `os` module provides methods to retrieve information about the operating system, like CPU architecture and memory.

59. **What is the purpose of the `url` module in Node.js?**
   The `url` module provides utilities for parsing and formatting URLs.

60. **What is the `process.nextTick()` method in Node.js?**
   `process.nextTick()` schedules a function to be executed in the next iteration of the event loop.

61. **What is the significance of the `path.resolve()` method in Node.js?**
    `path.resolve()` converts relative paths to absolute paths and resolves any symlinks.

62. **Explain the concept of the `crypto` module in Node.js.**
    The `crypto` module provides cryptographic functionalities, such as generating hashes and creating digital signatures.

63. **What is the purpose of the `process.env` object in Node.js?**
    `process.env` provides access to environment variables, allowing configuration of application behavior.

64. **What is the role of the `dns` module in Node.js?**
    The `dns` module in Node.js is used for DNS (Domain Name System) lookups.

65. **Explain the concept of a duplex stream in Node.js.**
    A duplex stream is a type of stream that can be used for both reading and writing.

66. **What is the `Buffer` class in Node.js?**
    The `Buffer` class is used to handle binary data in Node.js, such as images and files.

67. **What is the purpose of the `util.promisify()` function in Node.js?**
    `util.promisify()` converts callback-based functions into Promise-based functions.

68. **Explain the concept of the `cluster` module in Node.js.**
    The `cluster` module allows you to create multiple child processes to utilize multiple CPU cores.

69. **What is the purpose of the `process.cwd()` method in Node.js?**
    `process.cwd()` returns the current working directory of the Node.js process.

70. **What is a native addon in Node.js?**
    A native addon is a dynamically linked shared library that can be loaded by Node.js modules.

71. **Explain the concept of garbage collection in Node.js.**
    Garbage collection is the process of automatically freeing up memory by identifying and cleaning up unused objects.

72. **What is the purpose of the `util.inspect()` method in Node.js?**
    `util.inspect()` is used to convert objects into human-readable strings, aiding in debugging.

73. **What is the role of the `os.tmpdir()` method in Node.js?**
    `os.tmpdir()` returns the operating system's default directory for temporary files.

74. **Explain the concept of an EventEmitter in Node.js.**
    An EventEmitter is a class in Node.js that facilitates communication between objects by emitting and handling events.

75. **What is the purpose of the `stream.Readable` class in Node.js?**
    The `stream.Readable` class is used for reading data from a source, such as a file or a network socket.

76. **What is the Node.js `async_hooks` module used for?**
    The `async_hooks` module provides a way to track asynchronous operations across different contexts.

77. **Explain the concept of the `process.on()` method in Node.js.**
    `process.on()` is used to register event listeners for various process-related events.

78. **What is the purpose of the `util.inherits()` function in Node.js?**
    `util.inherits()` is used to create inheritance relationships between constructor functions.

79. **What are child processes and how are they managed in Node.js?**
    Child processes are separate instances of the Node.js runtime, created using the `child_process` module, and they can be managed using features like `spawn` and `fork`.

80. **Explain the concept of the `async` module in Node.js

.**
    The `async` module provides functions for handling asynchronous operations, simplifying control flow.

81. **What is the `stream.Writable` class used for in Node.js?**
    The `stream.Writable` class is used for writing data to a destination, such as a file or a network socket.

82. **What is the purpose of the `util.format()` method in Node.js?**
    `util.format()` is used to format strings by replacing placeholders with values.

83. **Explain the concept of the `util.inspect.defaultOptions` object in Node.js.**
    `util.inspect.defaultOptions` is an object that holds default options for the `util.inspect()` method.

84. **What is the role of the `readline` module in Node.js?**
    The `readline` module provides an interface for reading input from readable streams, particularly the command line.

85. **What is the purpose of the `util.isDeepStrictEqual()` method in Node.js?**
    `util.isDeepStrictEqual()` is used to compare values for deep equality, accounting for differences in objects.

86. **Explain the concept of a transform stream in Node.js.**
    A transform stream is a type of stream that can modify data as it passes through, making it useful for processing data during streaming.

87. **What is the purpose of the `util.deprecate()` function in Node.js?**
    `util.deprecate()` marks a function as deprecated, logging a warning message when it is used.

88. **What is the role of the `cluster.isMaster` property in Node.js?**
    `cluster.isMaster` is a boolean value that indicates whether the current process is the master process in a cluster.

89. **Explain the concept of the `util.inspect.custom` symbol in Node.js.**
    `util.inspect.custom` is a symbol used to customize the output of the `util.inspect()` method for user-defined objects.

90. **What is the purpose of the `util.promisify.custom` symbol in Node.js?**
    `util.promisify.custom` is a symbol used to customize the behavior of the `util.promisify()` function.

91. **What is the role of the `module._compile()` function in Node.js?**
`module._compile()` is an internal function that compiles and executes a module's source code.

92. **Explain the concept of the `os.freemem()` method in Node.js.**
`os.freemem()` returns the amount of free system memory in bytes.

93. **What is the purpose of the `os.totalmem()` method in Node.js?**
`os.totalmem()` returns the total amount of system memory in bytes.

94. **What is the role of the `os.arch()` method in Node.js?**
`os.arch()` returns the architecture of the operating system's CPU.

95. **Explain the concept of a readable stream in Node.js.**
A readable stream is an abstraction for reading data from a source in a chunked manner.

96. **What is the purpose of the `console.time()` and `console.timeEnd()` methods in Node.js?**
`console.time()` and `console.timeEnd()` are used to measure the time taken by a particular operation.

97. **What is the role of the `fs.watch()` method in Node.js?**
`fs.watch()` is used to monitor changes in files and directories.

98. **Explain the concept of a duplex stream in Node.js.**
A duplex stream is a type of stream that can be used for both reading and writing data.

99. **What is the purpose of the `http.STATUS_CODES` object in Node.js?**
`http.STATUS_CODES` is an object that contains HTTP status codes as keys and their corresponding messages as values.

100. **Explain the concept of a transform stream in Node.js.**
A transform stream is a type of stream that can modify data as it passes through, making it useful for processing data during streaming.