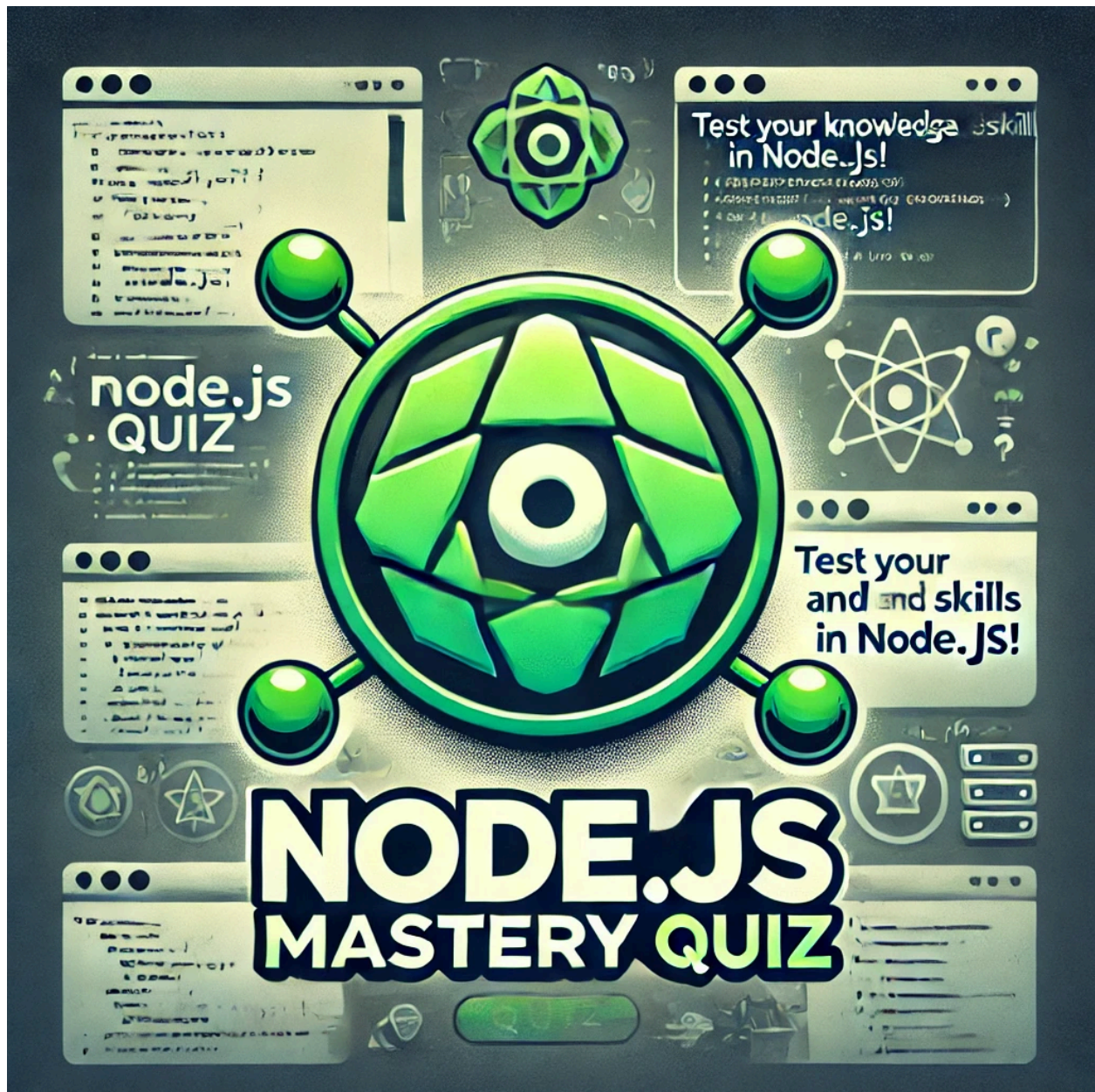


# 100 Node JS Questions



Welcome to the **Node.js Mastery Quiz**! This quiz is designed to challenge and enhance your understanding of Node.js, covering a wide range of topics including core modules, asynchronous programming, server creation, and more. Whether you're a beginner looking to test your

knowledge or a seasoned developer aiming to refresh your skills, this quiz offers something for everyone.

As you progress through the questions, you'll encounter various scenarios and problems that Node.js developers often face in real-world applications. The detailed explanations provided for each answer will help reinforce your learning and ensure you gain a deeper understanding of the concepts.

Good luck, and may your Node.js skills shine!

## What is Node.js primarily used for?

- A) Frontend development
- B) Backend development
- C) Mobile app development
- D) Database management

**Answer:** B) Backend development

**Explanation:** Node.js is primarily used for backend development. It allows developers to use JavaScript for server-side scripting, enabling the creation of scalable and high-performance applications.

---

## Which of the following is a core module in Node.js?

- A) React
- B) Express
- C) HTTP
- D) Lodash

**Answer:** C) HTTP

**Explanation:** The HTTP module is a core module in Node.js that allows Node.js to transfer data over the Hypertext Transfer Protocol (HTTP). React and Express are not core modules; they are external libraries. Lodash is a utility library, also not part of the Node.js core.

---

## What does the 'require' function do in Node.js?

- A) It imports a module.
- B) It exports a module.
- C) It defines a new module.
- D) It creates a new HTTP server.

**Answer:** A) It imports a module.

**Explanation:** The `require` function in Node.js is used to import modules, JSON files, or local files into the current module. It allows Node.js to include various functionalities, such as built-in modules or third-party libraries.

---

**Which of the following command is used to install a package globally in Node.js?**

- A) `npm install <package_name>`
- B) `npm install -g <package_name>`
- C) `npm install --global`
- D) `npm install <package_name> --save`

**Answer:** B) `npm install -g <package_name>`

**Explanation:** The `-g` flag in `npm install -g <package_name>` installs the specified package globally on your system, making it accessible from any directory.

---

**What is the purpose of the package.json file in a Node.js project?**

- A) To store the project's database credentials
- B) To define the metadata for the project, including dependencies
- C) To configure the project's web server
- D) To keep track of version history

**Answer:** B) To define the metadata for the project, including dependencies

**Explanation:** The `package.json` file is a crucial part of any Node.js project. It contains metadata about the project, such as the name, version, and dependencies, and it is used by npm to manage packages.

---

**How do you create an HTTP server in Node.js?**

- A) By using the `http.createServer` method
- B) By using the `fs.createServer` method
- C) By using the `net.createServer` method
- D) By using the `express.createServer` method

**Answer:** A) By using the `http.createServer` method

**Explanation:** The `http.createServer` method is used in Node.js to create an HTTP server that listens to server ports and gives a response back to the client.

---

## Which of the following is true about Node.js event-driven architecture?

- A) It allows asynchronous processing
- B) It blocks the execution until an event is triggered
- C) It requires multithreading
- D) It is not suitable for I/O operations

**Answer:** A) It allows asynchronous processing

**Explanation:** Node.js uses an event-driven architecture to handle asynchronous operations efficiently. This means that while waiting for an operation to complete, Node.js can continue executing other code, making it well-suited for I/O operations.

---

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

- A) To handle HTTP requests
- B) To perform file system operations
- C) To manage network connections
- D) To parse URLs

**Answer:** B) To perform file system operations

**Explanation:** The `fs` module in Node.js provides an API for interacting with the file system, allowing you to read, write, and manipulate files.

---

## Which command would you use to initialize a new Node.js project?

- A) `npm init`
- B) `node init`

- C) `npm start`
- D) `node start`

**Answer:** A) `npm init`

**Explanation:** The `npm init` command is used to initialize a new Node.js project by creating a `package.json` file. This file will guide you through the setup of the project's basic information.

---

## What does the term "non-blocking I/O" refer to in Node.js?

- A) The code execution waits for I/O operations to complete
- B) I/O operations are offloaded to a separate thread
- C) The code execution continues without waiting for I/O operations to complete
- D) I/O operations are blocked until all code execution is finished

**Answer:** C) The code execution continues without waiting for I/O operations to complete

**Explanation:** "Non-blocking I/O" in Node.js refers to the ability to continue executing code without waiting for I/O operations (like reading from a file or making a network request) to complete. This allows for better performance, especially in I/O-heavy applications.

---

## Which method is used to read a file synchronously in Node.js?

- A) `fs.readFile()`
- B) `fs.readFileSync()`
- C) `fs.read()`
- D) `fs.readSync()`

**Answer:** B) `fs.readFileSync()`

**Explanation:** The `fs.readFileSync()` method is used to read a file synchronously in Node.js. Unlike `fs.readFile()`, it blocks the execution of the program until the file is read.

---

## Which of the following is true about the `EventEmitter` class in Node.js?

- A) It allows you to create custom events
- B) It is used to manage file systems
- C) It manages HTTP requests
- D) It is deprecated and should not be used

**Answer:** A) It allows you to create custom events

**Explanation:** The `EventEmitter` class in Node.js is part of the `events` module and allows you to create and handle custom events, making it a fundamental part of the event-driven architecture of Node.js.

---

## How do you export a module in Node.js?

- A) `module.export = ...`
- B) `module.exports = ...`
- C) `exports = ...`
- D) `export module = ...`

**Answer:** B) `module.exports = ...`

**Explanation:** In Node.js, you export a module using `module.exports`. This allows the exported module to be required and used in other files.

---

## What is the purpose of the `util` module in Node.js?

- A) To handle HTTP utilities
- B) To provide utility functions for debugging and formatting
- C) To manage file system operations
- D) To handle database connections

**Answer:** B) To provide utility functions for debugging and formatting

**Explanation:** The `util` module in Node.js offers various utility functions, such as `util.format`, `util.debug`, and `util.inherits`, which are useful for debugging, formatting strings, and working with object inheritance.

---



**Which of the following commands is used to uninstall a package in Node.js?**

- A) `npm remove <package_name>`
- B) `npm uninstall <package_name>`
- C) `npm delete <package_name>`
- D) `npm erase <package_name>`

**Answer:** B) `npm uninstall <package_name>`

**Explanation:** The `npm uninstall <package_name>` command is used to remove a package that was previously installed, removing it from the `node_modules` directory and updating the `package.json` file.

---

**Which method in the fs module is used to rename a file?**

- A) `fs.renameFile()`
- B) `fs.rename()`
- C) `fs.move()`
- D) `fs.changeName()`

**Answer:** B) `fs.rename()`

**Explanation:** The `fs.rename()` method is used to rename a file in Node.js. It can also be used to move a file by specifying a new path.

---

**What does the path.join() method do in Node.js?**

- A) Combines two or more path segments into one
- B) Joins the content of two files
- C) Splits a file path into segments
- D) Removes a segment from a file path

**Answer:** A) Combines two or more path segments into one

**Explanation:** The `path.join()` method in Node.js is used to join multiple path segments into a single path, making it easier to work with file paths across different platforms.



---

## Which statement best describes the Node.js process object?

- A) It is used to manage child processes
- B) It provides information about the current Node.js process
- C) It is responsible for handling HTTP requests
- D) It manages event listeners

**Answer:** B) It provides information about the current Node.js process

**Explanation:** The `process` object in Node.js provides various information and methods related to the current Node.js process, such as environment variables, arguments passed to the script, and the process ID.

---

## How do you check if a specific package is installed globally using npm?

- A) `npm list <package_name>`
- B) `npm list -g <package_name>`
- C) `npm global <package_name>`
- D) `npm search <package_name>`

**Answer:** B) `npm list -g <package_name>`

**Explanation:** The `npm list -g <package_name>` command lists all globally installed packages, including whether a specific package is installed globally.

---

## Which method is used to convert a JavaScript object into a JSON string in Node.js?

- A) `JSON.stringify()`
- B) `JSON.parse()`
- C) `JSON.toString()`
- D) `JSON.encode()`

**Answer:** A) `JSON.stringify()`

**Explanation:** The `JSON.stringify()` method is used to convert a JavaScript object into a JSON string, making it suitable for storage or transmission.

---

## What is the default scope of variables declared with `var` in a Node.js script?

- A) Block scope
- B) Function scope
- C) Global scope
- D) Module scope

**Answer:** B) Function scope

**Explanation:** Variables declared with `var` have function scope, meaning they are only accessible within the function where they are declared. If declared outside a function, they have global scope.

---

## What does the `next()` function do in Express.js?

- A) It sends the next HTTP request
- B) It ends the current HTTP request
- C) It passes control to the next middleware function
- D) It renders the next view

**Answer:** C) It passes control to the next middleware function

**Explanation:** The `next()` function in Express.js is used to pass control from one middleware function to the next. This allows for more modular and reusable code in your Express applications.

---

## Which of the following is a valid method to create a new instance of an HTTP client in Node.js?

- A) `http.request()`
- B) `http.getClient()`
- C) `http.createClient()`
- D) `http.newRequest()`

**Answer:** A) `http.request()`

**Explanation:** The `http.request()` method is used to create a new instance of an HTTP client in Node.js, allowing you to send HTTP requests.

---

## What is the purpose of the cluster module in Node.js?

- A) To handle database connections
- B) To enable the use of multiple threads
- C) To create child processes that share the same server port
- D) To manage file system operations

**Answer:** C) To create child processes that share the same server port

**Explanation:** The `cluster` module in Node.js allows you to create multiple child processes that share the same server port, effectively enabling the use of all CPU cores to improve performance.

---

## Which of the following is not a valid HTTP method?

- A) GET
- B) POST
- C) SEND
- D) DELETE

**Answer:** C) SEND

**Explanation:** SEND is not a valid HTTP method. The most common HTTP methods are GET, POST, PUT, DELETE, PATCH, and OPTIONS.

---

## How can you handle exceptions in Node.js?

- A) By using try-catch blocks
- B) By using throw-catch blocks
- C) By using catch-try blocks
- D) By using catch-finally blocks

**Answer:** A) By using try-catch blocks

**Explanation:** In Node.js, exceptions are typically handled using try-catch blocks, allowing you to catch errors and handle them gracefully.

---

## What does the res.send() method do in Express.js?

- A) It sends an HTTP response to the client
- B) It sends a request to another server
- C) It ends the server process
- D) It redirects the client to another URL

**Answer:** A) It sends an HTTP response to the client

**Explanation:** The `res.send()` method in Express.js is used to send an HTTP response back to the client. It can send strings, objects, or arrays as a response.

---

## Which of the following commands is used to update all packages in a Node.js project to their latest versions?

- A) `npm update --all`
- B) `npm upgrade`
- C) `npm update`
- D) `npm install latest`

**Answer:** C) `npm update`

**Explanation:** The `npm update` command is used to update all the packages in your project to their latest versions according to the `package.json` and `package-lock.json` files.

---

## What is the purpose of the buffer module in Node.js?

- A) To manage HTTP buffers
- B) To handle binary data
- C) To buffer I/O operations
- D) To manage memory allocation

**Answer:** B) To handle binary data

**Explanation:** The `buffer` module in Node.js is used to handle binary data directly in memory. It is especially useful when dealing with streams and file I/O.

---

## Which of the following is the correct syntax to create a new instance of an Express.js application?

- A) `const app = new express();`
- B) `const app = express();`
- C) `const app = createExpressApp();`
- D) `const app = express.create();`

**Answer:** B) `const app = express();`

**Explanation:** The correct way to create a new instance of an Express.js application is by calling the `express()` function, which returns an instance of an Express app.

---

## What does the npm start command do in a Node.js project?

- A) It compiles the project
- B) It starts the Node.js server
- C) It installs the dependencies
- D) It updates the project to the latest version

**Answer:** B) It starts the Node.js server

**Explanation:** The `npm start` command is used to start the Node.js server according to the `start` script defined in the `package.json` file.

---

## How do you listen for a specific event on an EventEmitter instance in Node.js?

- A) `eventEmitter.listen('event', callback)`
- B) `eventEmitter.on('event', callback)`
- C) `eventEmitter.add('event', callback)`
- D) `eventEmitter.trigger('event', callback)`

**Answer:** B) `eventEmitter.on('event', callback)`

**Explanation:** The `eventEmitter.on('event', callback)` method is used to listen for a specific event on an `eventEmitter` instance. When the event is emitted, the callback function is executed.

---

## Which Node.js method is used to create a directory asynchronously?

- A) `fs.mkdirSync()`
- B) `fs.createDir()`
- C) `fs.mkdir()`
- D) `fs.makeDir()`

**Answer:** C) `fs.mkdir()`

**Explanation:** The `fs.mkdir()` method is used to create a directory asynchronously in Node.js. The `fs.mkdirSync()` method is the synchronous version of this function.

---

## What is the role of the next argument in Express middleware?

- A) It represents the next request in the queue
- B) It moves control to the next middleware function
- C) It sends the response back to the client
- D) It handles the error if the middleware fails

**Answer:** B) It moves control to the next middleware function

**Explanation:** The `next` argument in Express middleware is used to pass control to the next middleware function in the stack, allowing for a sequential execution of middleware functions.

---

**What is the correct command to install a specific version of a package using npm?**

- A) `npm install <package_name>@<version>`
- B) `npm install <package_name>#<version>`
- C) `npm install <package_name> <version>`
- D) `npm install <package_name>-<version>`

**Answer:** A) `npm install <package_name>@<version>`

**Explanation:** To install a specific version of a package, you use the syntax `npm install <package_name>@<version>`. For example, `npm install express@4.17.1`.

---

**Which method is used to check if a file exists asynchronously in Node.js?**

- A) `fs.existsFile()`
- B) `fs.checkFile()`
- C) `fs.existsSync()`
- D) `fs.access()`

**Answer:** D) `fs.access()`

**Explanation:** The `fs.access()` method is used to check if a file exists asynchronously in Node.js. It checks the accessibility of a file or directory, allowing you to determine if it exists.

---

**What does the `res.json()` method do in Express.js?**

- A) It parses a JSON request body
- B) It sends a JSON response
- C) It renders a JSON file
- D) It reads a JSON file



**Answer:** B) It sends a JSON response

**Explanation:** The `res.json()` method in Express.js is used to send a JSON response back to the client. It automatically sets the Content-Type header to `application/json`.

---

### Which of the following is a built-in module in Node.js?

- A) Express
- B) Lodash
- C) Crypto
- D) Axios

**Answer:** C) Crypto

**Explanation:** The `crypto` module is a built-in module in Node.js that provides cryptographic functionality. Express, Lodash, and Axios are third-party libraries.

---

### How do you read the environment variables in Node.js?

- A) `process.env.VARIABLE_NAME`
- B) `env.VARIABLE_NAME`
- C) `node.env.VARIABLE_NAME`
- D) `process.variable.VARIABLE_NAME`

**Answer:** A) `process.env.VARIABLE_NAME`

**Explanation:** In Node.js, environment variables can be accessed using `process.env.VARIABLE_NAME`, where `VARIABLE_NAME` is the name of the environment variable.

---

### Which command is used to install a package and save it as a development dependency in Node.js?

- A) `npm install <package_name> --dev`
- B) `npm install <package_name> --save-dev`

- C) `npm install <package_name> --save`
- D) `npm install <package_name> --dev-save`

**Answer:** B) `npm install <package_name> --save-dev`

**Explanation:** The `--save-dev` flag is used with `npm install` to install a package and save it as a development dependency in the `package.json` file.

---

## What is the default port used by a Node.js application?

- A) 3000
- B) 8000
- C) 8080
- D) 5000

**Answer:** A) 3000

**Explanation:** By convention, many Node.js applications use port 3000 as the default port, though this can be configured to any available port.

---

## Which method is used to write data to a stream in Node.js?

- A) `stream.writeData()`
- B) `stream.write()`
- C) `stream.send()`
- D) `stream.push()`

**Answer:** B) `stream.write()`

**Explanation:** The `stream.write()` method is used to write data to a stream in Node.js. It can be used to send data to a writable stream such as a file or a network socket.

---

## How do you create a child process in Node.js?

- A) `child_process.run()`
- B) `child_process.exec()`

- C) `child_process.spawn()`
- D) `child_process.fork()`

**Answer:** C) `child_process.spawn()`

**Explanation:** The `child_process.spawn()` method is used to create a child process in Node.js. It allows you to execute a command in a new process.

---

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

- A) To send a file as a response
- B) To end the server process
- C) To redirect the client to a different URL
- D) To send a JSON response

**Answer:** C) To redirect the client to a different URL

**Explanation:** The `res.redirect()` method in Express.js is used to redirect the client to a different URL. This can be useful for redirecting after a form submission or other user action.

---

## What does the `npm init` command do?

- A) It creates a new npm package
- B) It installs a package
- C) It initializes a new Node.js project
- D) It updates npm to the latest version

**Answer:** C) It initializes a new Node.js project

**Explanation:** The `npm init` command initializes a new Node.js project by creating a `package.json` file, which contains metadata about the project and its dependencies.

---

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

- A) `fs.appendFileSync()`
- B) `fs.writeFile()`
- C) `fs.appendFile()`
- D) `fs.addToFile()`

**Answer:** C) `fs.appendFile()`

**Explanation:** The `fs.appendFile()` method is used to append data to a file asynchronously in Node.js. It adds the specified data to the end of the file.

---

## What is the purpose of the npm audit command?

- A) To check for vulnerabilities in installed packages
- B) To update all packages to their latest versions
- C) To install the latest security patches
- D) To uninstall deprecated packages

**Answer:** A) To check for vulnerabilities in installed packages

**Explanation:** The `npm audit` command is used to check for known vulnerabilities in the installed packages. It provides a report of any issues found and suggests possible fixes.

---

## Which method is used to handle uncaught exceptions in Node.js?

- A) `process.on('uncaughtException', callback)`
- B) `process.catch('exception', callback)`
- C) `process.listen('exception', callback)`
- D) `process.handle('uncaughtException', callback)`

**Answer:** A) `process.on('uncaughtException', callback)`

**Explanation:** The `process.on('uncaughtException', callback)` method is used to handle uncaught exceptions in Node.js. It allows you to catch exceptions that are not handled within your code.

---

## Which of the following is true about Node.js streams?

- A) They are used to buffer data in memory
- B) They are used to handle asynchronous I/O operations
- C) They are used to manage HTTP requests
- D) They are always blocking

**Answer:** B) They are used to handle asynchronous I/O operations

**Explanation:** Node.js streams are used to handle asynchronous I/O operations efficiently by processing data piece by piece as it is received, rather than loading it all into memory at once.

---

## What does the `res.render()` method do in Express.js?

- A) It sends a JSON response
- B) It sends a file as a response
- C) It renders a view and sends the HTML to the client
- D) It ends the HTTP request

**Answer:** C) It renders a view and sends the HTML to the client

**Explanation:** The `res.render()` method in Express.js is used to render a view template and send the resulting HTML to the client. It is commonly used in applications with a templating engine.

---

## What does the `npm run` command do?

- A) It installs a package
- B) It runs a script defined in the `package.json` file
- C) It starts the Node.js server
- D) It updates npm to the latest version

**Answer:** B) It runs a script defined in the `package.json` file

**Explanation:** The `npm run` command is used to run a script defined in the `scripts` section of the `package.json` file. For example, `npm run start` will run the `start` script.

---

## Which method is used to read a file line by line in Node.js?

- A) `fs.readFileLineByLine()`
- B) `fs.readLines()`
- C) `readline.createInterface()`
- D) `fs.readFileLines()`

**Answer:** C) `readline.createInterface()`

**Explanation:** The `readline.createInterface()` method is used to read a file line by line in Node.js. It creates an interface for reading data from a readable stream, such as a file.

---

## What does the `util.promisify()` function do in Node.js?

- A) It converts a callback-based function to a promise-based one
- B) It converts a synchronous function to an asynchronous one
- C) It handles promise rejections
- D) It converts a promise-based function to a callback-based one

**Answer:** A) It converts a callback-based function to a promise-based one

**Explanation:** The `util.promisify()` function in Node.js converts a function that uses a callback to one that returns a promise, making it easier to work with asynchronous code.

---

## Which method is used to send a file as a response in Express.js?

- A) `res.sendFile()`
- B) `res.file()`
- C) `res.send()`
- D) `res.download()`

**Answer:** A) `res.sendFile()`

**Explanation:** The `res.sendFile()` method in Express.js is used to send a file as an HTTP response. It takes the path of the file as an argument.

---

## What is the purpose of the `os` module in Node.js?

- A) To manage HTTP requests
- B) To interact with the operating system
- C) To handle file system operations
- D) To manage network connections

**Answer:** B) To interact with the operating system

**Explanation:** The `os` module in Node.js provides a number of operating system-related utility methods, such as retrieving information about the CPU, memory, and network interfaces.

---

## How do you handle rejected promises in Node.js?

- A) Using `try-catch` blocks
- B) Using `promise.catch()`
- C) Using `promise.reject()`
- D) Using `promise.finally()`

**Answer:** B) Using `promise.catch()`

**Explanation:** Rejected promises in Node.js can be handled using the `catch()` method, which is chained to the promise. This method is used to catch and handle any errors that occur during the execution of the promise.

---

## Which of the following is not a valid stream type in Node.js?

- A) Readable
- B) Writable
- C) Duplex
- D) Piped

**Answer:** D) Piped

**Explanation:** Piped is not a valid stream type in Node.js. The valid stream types are Readable, Writable, Duplex, and Transform.

---

## Which method is used to parse a JSON string into a JavaScript object?



- A) `JSON.stringify()`
- B) `JSON.parse()`
- C) `JSON.toObject()`
- D) `JSON.decode()`

**Answer:** B) `JSON.parse()`

**Explanation:** The `JSON.parse()` method is used to parse a JSON string and convert it into a JavaScript object.

---

## How do you make a module globally accessible in Node.js?

- A) By using `global.require()`
- B) By using `global.export()`
- C) By attaching it to the `global` object
- D) By using `require.cache`

**Answer:** C) By attaching it to the `global` object

**Explanation:** In Node.js, you can make a module globally accessible by attaching it to the `global` object, though this practice is generally discouraged in favor of more modular and maintainable code.

---

## Which of the following is true about the `http` module in Node.js?

- A) It is a third-party module
- B) It is used to create HTTP servers and clients
- C) It is used to manage file systems
- D) It is used to handle database connections

**Answer:** B) It is used to create HTTP servers and clients

**Explanation:** The `http` module in Node.js is a core module that provides functionality for creating HTTP servers and clients, enabling communication over the web.

---

## What does the npm install command do when run without any arguments?

- A) It installs the latest version of npm
- B) It installs all dependencies listed in package . j son
- C) It installs Node.js
- D) It installs the latest version of Node.js

**Answer:** B) It installs all dependencies listed in package . j son

**Explanation:** When run without any arguments, the npm `install` command installs all the dependencies listed in the package . j son file for the project.

---

## How do you stop a running Node.js application?

- A) Ctrl + C
- B) Ctrl + Z
- C) Ctrl + X
- D) Ctrl + Q

**Answer:** A) Ctrl + C

**Explanation:** To stop a running Node.js application, you can use Ctrl + C in the terminal where the application is running. This sends a signal to terminate the process.

---

## Which of the following is a characteristic of Node.js?

- A) It is multi-threaded
- B) It is single-threaded but asynchronous
- C) It is synchronous by default
- D) It is designed for heavy CPU-bound operations

**Answer:** B) It is single-threaded but asynchronous

**Explanation:** Node.js is single-threaded but uses an asynchronous, non-blocking I/O model, which makes it efficient for handling concurrent operations.

---

## Which method is used to make a GET request in Node.js?

- A) `http.get()`
- B) `http.request()`
- C) `http.send()`
- D) `http.connect()`

**Answer:** A) `http.get()`

**Explanation:** The `http.get()` method in Node.js is used to make a GET request to a specified URL. It is a simpler version of `http.request()` for making GET requests.

---

## What does the `res.status()` method do in Express.js?

- A) It sets the HTTP status code for the response
- B) It sends a status update to the server
- C) It sends a status update to the client
- D) It checks the status of the HTTP request

**Answer:** A) It sets the HTTP status code for the response

**Explanation:** The `res.status()` method in Express.js is used to set the HTTP status code for the response before sending it back to the client.

---

## How can you use environment variables in a Node.js application?

- A) By using the `env` module
- B) By using the `dotenv` module
- C) By using the `process.env` object
- D) By using the `global.env` object

**Answer:** C) By using the `process.env` object

**Explanation:** Environment variables in Node.js can be accessed using the `process.env` object. For example, `process.env.PORT` would access the `PORT` environment variable.

---

**Which command is used to view the installed version of Node.js?**

- A) `node -v`
- B) `npm -v`
- C) `node --version`
- D) `node version`

**Answer:** A) `node -v`

**Explanation:** The `node -v` or `node --version` command is used to view the installed version of Node.js on your system.

---

**Which of the following is true about Node.js modules?**

- A) Modules are only available within the same file
- B) Modules cannot be reused across different projects
- C) Modules help in organizing code and promoting reusability
- D) Modules are deprecated and should be avoided

**Answer:** C) Modules help in organizing code and promoting reusability

**Explanation:** Node.js modules are a way to encapsulate and organize code into reusable components that can be easily imported and used across different parts of a project or across different projects.

---

**Which of the following methods can be used to make a POST request in Node.js?**

- A) `http.get()`
- B) `http.post()`
- C) `http.request()`
- D) `http.send()`

**Answer:** C) `http.request()`

**Explanation:** The `http.request()` method in Node.js is a versatile method that can be used to make POST requests by specifying the POST method in the options.

---

## What is the purpose of the `res.sendStatus()` method in Express.js?

- A) To send a custom status message
- B) To send a file as a response
- C) To send an HTTP status code with a short message
- D) To send a JSON response

**Answer:** C) To send an HTTP status code with a short message

**Explanation:** The `res.sendStatus()` method in Express.js is used to send an HTTP status code along with the corresponding short message (e.g., 200 OK).

---

## Which method is used to write data to a file asynchronously in Node.js?

- A) `fs.writeFileSync()`
- B) `fs.write()`
- C) `fs.writeFile()`
- D) `fs.appendFile()`

**Answer:** C) `fs.writeFile()`

**Explanation:** The `fs.writeFile()` method is used to write data to a file asynchronously in Node.js. It replaces the contents of the file if it already exists.

---

## What does the `util.inherits()` function do in Node.js?

- A) It converts a callback-based function to a promise-based one
- B) It sets up prototypal inheritance between two constructor functions
- C) It converts a promise-based function to a callback-based one
- D) It creates a new instance of a class

**Answer:** B) It sets up prototypal inheritance between two constructor functions

**Explanation:** The `util.inherits()` function in Node.js is used to set up prototypal inheritance between two constructor functions, allowing one to inherit methods from the other.

---

**Which of the following is a commonly used templating engine with Express.js?**

- A) EJS
- B) Lodash
- C) Mocha
- D) Mongoose

**Answer:** A) EJS

**Explanation:** EJS (Embedded JavaScript) is a commonly used templating engine with Express.js that allows you to generate HTML with JavaScript.

---

**What is the purpose of the `path.resolve()` method in Node.js?**

- A) To resolve a path to an absolute path
- B) To join multiple paths into one
- C) To split a path into segments
- D) To check if a path exists

**Answer:** A) To resolve a path to an absolute path

**Explanation:** The `path.resolve()` method in Node.js resolves a sequence of path segments into an absolute path.

---

**Which command is used to uninstall a package in Node.js?**

- A) `npm remove <package_name>`
- B) `npm uninstall <package_name>`
- C) `npm delete <package_name>`
- D) `npm erase <package_name>`

**Answer:** B) `npm uninstall <package_name>`

**Explanation:** The `npm uninstall <package_name>` command is used to remove a package from the `node_modules` directory and update the `package.json` file.

---

**Which Node.js module is used for streamlining the handling of routes in an Express.js application?**

- A) RouteHandler
- B) PathManager
- C) ExpressRouter
- D) Router

**Answer:** D) Router

**Explanation:** The Router module in Express.js is used to create modular, mountable route handlers, making it easier to manage complex routing in an application.

---

**Which method in Node.js is used to close a file descriptor?**

- A) `fs.closeFile()`
- B) `fs.close()`
- C) `fs.closeSync()`
- D) `fs.endFile()`

**Answer:** B) `fs.close()`

**Explanation:** The `fs.close()` method is used to close a file descriptor in Node.js. It takes the file descriptor and a callback function as arguments.

---

**What does the npm link command do?**

- A) It creates a symbolic link to a global package
- B) It installs a package globally
- C) It updates a package to the latest version
- D) It creates a symbolic link to a local package

**Answer:** A) It creates a symbolic link to a global package

**Explanation:** The `npm link` command creates a symbolic link from a globally installed



package to a local directory, allowing you to work on a package locally and test it as if it were installed globally.

---

**Which of the following is a valid method to create an HTTPS server in Node.js?**

- A) `http.createServer()`
- B) `https.createServer()`
- C) `tls.createServer()`
- D) `ssl.createServer()`

**Answer:** B) `https.createServer()`

**Explanation:** The `https.createServer()` method in Node.js is used to create an HTTPS server, which is similar to the HTTP server but with added security using SSL/TLS.

---

**What does the `os.cpus()` method return in Node.js?**

- A) The number of CPUs
- B) The architecture of the CPU
- C) An array of objects containing information about each CPU/core
- D) The speed of the CPU

**Answer:** C) An array of objects containing information about each CPU/core

**Explanation:** The `os.cpus()` method returns an array of objects containing information about each CPU/core, such as model, speed, and times.

---

**Which method in the `fs` module is used to delete a file asynchronously in Node.js?**

- A) `fs.delete()`
- B) `fs.removeFile()`

- C) `fs.unlink()`
- D) `fs.erase()`

**Answer:** C) `fs.unlink()`

**Explanation:** The `fs.unlink()` method is used to delete a file asynchronously in Node.js. It removes the specified file from the file system.

---

## What does the npm cache clean command do?

- A) It clears the npm cache
- B) It removes all installed packages
- C) It updates the npm cache
- D) It deletes the `node_modules` folder

**Answer:** A) It clears the npm cache

**Explanation:** The `npm cache clean` command is used to clear the npm cache, which can help resolve issues with corrupted packages or outdated files.

---

## Which of the following is a valid exit code for a successful Node.js process?

- A) 0
- B) 1
- C) 127
- D) 255

**Answer:** A) 0

**Explanation:** An exit code of 0 indicates that a Node.js process has completed successfully. Non-zero exit codes typically indicate an error.

---

## What is the purpose of the `os.uptime()` method in Node.js?

- A) To return the time elapsed since the OS was last rebooted
- B) To return the time since the Node.js process started
- C) To return the uptime of the server
- D) To return the uptime of the current session

**Answer:** A) To return the time elapsed since the OS was last rebooted

**Explanation:** The `os.uptime()` method in Node.js returns the system uptime in seconds, representing the time elapsed since the OS was last rebooted.

---

### Which of the following methods can be used to terminate a Node.js process?

- A) `process.terminate()`
- B) `process.end()`
- C) `process.exit()`
- D) `process.stop()`

**Answer:** C) `process.exit()`

**Explanation:** The `process.exit()` method is used to terminate a Node.js process. It can be called with an optional exit code.

---

### Which of the following is true about Node.js timers?

- A) `setTimeout()` is used to execute code after a delay
- B) `setInterval()` is used to execute code immediately
- C) `clearTimeout()` is used to reset a timer
- D) `setImmediate()` is used to execute code after a specified delay

**Answer:** A) `setTimeout()` is used to execute code after a delay

**Explanation:** The `setTimeout()` method is used to execute a function after a specified delay. `setInterval()` repeatedly executes a function with a fixed time delay between each call.

---

## Which of the following is a common use case for the `child_process` module in Node.js?

- A) To manage HTTP requests
- B) To execute shell commands
- C) To handle file system operations
- D) To parse URLs

**Answer:** B) To execute shell commands

**Explanation:** The `child_process` module in Node.js is commonly used to execute shell commands and create child processes, allowing you to run external processes from within a Node.js script.

---

## What does the `os.totalmem()` method return in Node.js?

- A) The total memory used by the Node.js process
- B) The total memory available to the OS
- C) The total memory available to the Node.js process
- D) The total memory of the system

**Answer:** D) The total memory of the system

**Explanation:** The `os.totalmem()` method returns the total amount of system memory in bytes.

---

## Which of the following is true about Node.js process events?

- A) The `exit` event is emitted when the process is about to exit
- B) The `error` event is emitted when the process starts
- C) The `disconnect` event is emitted when a child process is forked
- D) The `beforeExit` event is emitted when a script starts

**Answer:** A) The `exit` event is emitted when the process is about to exit

**Explanation:** The `exit` event in Node.js is emitted when the process is about to exit, allowing you to perform synchronous cleanup operations.

---

## What does the `os.platform()` method return in Node.js?

- A) The name of the platform (e.g., 'win32', 'darwin', 'linux')
- B) The version of the operating system
- C) The architecture of the CPU
- D) The hostname of the system

**Answer:** A) The name of the platform (e.g., 'win32', 'darwin', 'linux')

**Explanation:** The `os.platform()` method returns the platform on which the Node.js process is running, such as 'win32' for Windows, 'darwin' for macOS, and 'linux' for Linux.

---

## Which of the following is not a core module in Node.js?

- A) `fs`
- B) `http`
- C) `express`
- D) `path`

**Answer:** C) `express`

**Explanation:** `express` is not a core module in Node.js. It is a popular third-party web framework for building web applications. Core modules include `fs`, `http`, and `path`.

---

## What does the `npm outdated` command do?

- A) It checks for outdated npm packages
- B) It removes outdated npm packages
- C) It updates outdated npm packages
- D) It lists all installed npm packages

**Answer:** A) It checks for outdated npm packages

**Explanation:** The `npm outdated` command checks for outdated npm packages in your project and lists the current, wanted, and latest versions of each package.

---

**Which method in the `os` module is used to return the hostname of the system?**

- A) `os.hostname()`
- B) `os.getHostname()`
- C) `os.getHost()`
- D) `os.name()`

**Answer:** A) `os.hostname()`

**Explanation:** The `os.hostname()` method in Node.js returns the hostname of the system on which the Node.js process is running.

---

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

- A) `fs.appendFileSync()`
- B) `fs.writeFileSync()`
- C) `fs.addToFileSync()`
- D) `fs.syncAppend()`

**Answer:** A) `fs.appendFileSync()`

**Explanation:** The `fs.appendFileSync()` method is used to append data to a file synchronously in Node.js, adding the data to the end of the file.

---

**Which of the following is true about the `crypto` module in Node.js?**

- A) It is used for cryptographic functions such as encryption and hashing
- B) It is a third-party module
- C) It is used for managing file systems
- D) It is used for creating HTTP servers

**Answer:** A) It is used for cryptographic functions such as encryption and hashing

**Explanation:** The `crypto` module in Node.js provides cryptographic functionality, including encryption, decryption, hashing, and generating random values.

---

**Which method in the `child_process` module is used to execute a command in a shell and buffer the output?**

- A) `exec()`
- B) `spawn()`
- C) `fork()`
- D) `run()`

**Answer:** A) `exec()`

**Explanation:** The `exec()` method in the `child_process` module is used to execute a command in a shell and buffer the output, returning it via a callback function.

---

**Which method is used to remove a directory asynchronously in Node.js?**

- A) `fs.rmdir()`
- B) `fs.deleteDir()`
- C) `fs.removeDir()`
- D) `fs.unlinkDir()`

**Answer:** A) `fs.rmdir()`

**Explanation:** The `fs.rmdir()` method is used to remove a directory asynchronously in Node.js. If the directory is not empty, an error will be thrown unless the `recursive` option is set to `true`.

---

**What does the `os.networkInterfaces()` method return in Node.js?**



- A) An object containing the network interfaces of the system
- B) The IP address of the system
- C) The MAC address of the system
- D) The DNS server addresses

**Answer:** A) An object containing the network interfaces of the system

**Explanation:** The `os.networkInterfaces()` method returns an object containing information about the network interfaces of the system, including IP addresses and MAC addresses.

---

## What does the npm rebuild command do?

- A) It rebuilds the npm package cache
- B) It reinstalls all packages
- C) It rebuilds native Node.js modules
- D) It updates npm to the latest version

**Answer:** C) It rebuilds native Node.js modules

**Explanation:** The `npm rebuild` command rebuilds any native Node.js modules that have been installed. This can be useful if you change Node.js versions or encounter issues with native modules.

---

## Which method in the os module returns the architecture of the CPU in Node.js?

- A) `os.arch()`
- B) `os.cpuArch()`
- C) `os.getArch()`
- D) `os.cpu()`

**Answer:** A) `os.arch()`

**Explanation:** The `os.arch()` method in Node.js returns the architecture of the CPU, such as 'x64', 'arm', or 'ia32'.