**Node.js**

**Q1. When a javaScript function is invoked (called) in Node, where is a new frame placed?**

* the call stack
* the event loop
* the poll phase
* the events queue

**Explanation:** From javascripttutorial: [reference](https://www.javascripttutorial.net/javascript-call-stack/#:~:text=If%20a%20function%20calls%20another,top%20of%20the%20call%20stack.)

**Q2. Which of the following is a core module in Node?**

* webpack
* crypto
* request
* chalk

**Explanation:** From flaviocopes docs: [reference](https://flaviocopes.com/node-core-modules/)

**Q3. Which of the following Buffer class methods returns an uninitialized buffer?**

* allocUnsafe
* concat
* from
* alloc

**Explanation:** From official docs: [reference](https://nodejs.org/dist/latest-v13.x/docs/api/buffer.html#buffer_class_method_buffer_allocunsafe_size)

**Q4. Which of the following modules is NOT a built-in module in Node?**

* ftp
* events
* dgram
* http2

**Explanation:** From flaviocopes docs: [reference](https://flaviocopes.com/node-core-modules/)

**Q5. Which fs module method can be used to read the content of a file without buffering it in memory?**

* read
* readFile
* createReadStream
* readFileSync

**Explanation:** *From official docs:*[*reference*](https://nodejs.org/api/fs.html#fs_dir_read)*To minimize memory costs, when possible prefer streaming via fs.createReadStream().*

**Q6. Which of the following DNS module methods uses the underlying OS facilities and does not necessarily perform any network communication?**

* lookup
* resolve
* resolve4
* reverse

**Explanation:** From official docs: [reference](https://nodejs.org/api/dns.html#dns_dns_lookup_hostname_options_callback)

**Q7. How do you check that a value is a date object in Node?**

* util.types.isDate(value)
* assert.isDate(value)
* console.isDate(value)
* util.date(value)

**Explanation:** From official docs: [reference](https://nodejs.org/api/util.html#util_util_types_isdate_value)

**Q8. Can you create an https web server with Node.js?**

* no, there are no modules supporting it yet
* yes, with the https or http2 modules
* yes, through the path module
* yes, with the http module

**Explanation:** From official docs: [reference](https://nodejs.dev/learn/making-http-requests-with-nodejs)

**Q9. What is the Api that is designed to insulate Addons from changes in the underlying JavaScript engine?**

* A-API
* Z-API
* N-API
* X-API

**Explanation:** From official docs: [reference](https://nodejs.org/api/n-api.html#n_api_node_api)

**Q10. Which CLI option can you use to debug a node script in Chrome DevTools?**

* --dev-tools
* --inspect
* --chrome
* --debug

**Explanation:** From official docs: [reference](https://nodejs.org/en/docs/guides/debugging-getting-started/)

**Q11. How can you count the number of logical CPUs on the machine that is running Node?**

* node -p "process.cpus"
* node -p "util.cpus().size"
* node -p "process.os.cpus"
* node -p "os.cpus().length"

**Explanation:** From coderrocketfuel docs: [reference](https://coderrocketfuel.com/article/get-the-number-of-system-cpu-cores-using-node-js)

**Q12. Which of the following is a method on the console object?**

* exit
* test
* time
* print

**Explanation:** From official docs: [reference](https://nodejs.org/api/console.html)

**Q13. Which object is used to manage the cache of required modules?**

* global.cache
* module.cache
* process.cache
* require.cache

**Explanation:** From official docs: [reference](https://nodejs.org/api/modules.html#modules_require_cache)

**Q14. What is the command to silence all process warnings?**

* node index.js --trace-warnings
* node --no-warnings
* node -trace-warnings
* node index.js --no-warnings

**Explanation:** From official docs: [reference](https://nodejs.org/api/cli.html#cli_no_warnings)

**Q15. How can you use the promise API with a callback-based function such as child\_process.exec?**

* new Promise(child\_process.exec())
* util.promisify(child\_process.exec())
* util.promisify(child\_process.exec)
* new Promise(child\_process.exec)

**Explanation:** From official docs: [reference](https://nodejs.org/api/child_process.html)

**Q16. Which of the following is NOT a Node repl command?**

* .break
* .history
* .editor
* .save

**Explanation:** From official docs: [reference](https://nodejs.org/api/repl.html#repl_commands_and_special_keys)

**Q17. Which statement is true when you run the code shown below?**

require('child\_process').fork('script.js');

* The forked process shares the event loop with the parent process
* A new VM instance is created and the two VM instances will be shared between the forked process and the parent process.
* The forked process will have its own VM instance.
* The forked process shares the same VM thread with the parent process.

**Explanation:** From official docs: [reference](https://nodejs.org/api/child_process.html#child_process_child_process_fork_modulepath_args_options)

**Q18. If EventEmitter is in scope, which of the following lines of code will have an event emitter emitting a change event?**

* EventEmitter.emit('change');
* EventEmitter.new().emit('change');
* (new EventEmitter()).emit('change');
* new EventEmitter('change');

**Explanation:** *Because the EventEmitter is already in scope. No need to create new one.*

**Q19. Which of the following objects is a stream**

* process.uptime
* process.stdout
* process
* Buffer

**Explanation:** *process.stdout is Buffer type.*

**Q20. Which module variable holds the resolved absolute path of the current module file?**

* \_\_pathname
* \_\_location
* \_\_flder
* \_\_filename

**21. If the child\_process module methods are in scope, what is a current way to execute the command ps -ef using a child process?**

* spawn("ps -ef")
* exec("ps -ef")
* exec("ps", "-ef")
* fork("ps -ef")

**Q22.Which console method can be used to print the stack trace to the point of its execution?**

* stack
* trace
* debug
* print

**Q23.When you run JavaScript in a Node.js application, which of the following elements in a Node.js stack actually executes that JavaScript?**

* the libuv library
* the c-ares library
* the VM (like VS or Chakra)
* the repl module

**Q24. Looking at the code below, what does the console show?**

const http = require('http');

const hostname = '127.0.0.1'; const port = 3000;

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

res.statusCode = 200; res.setHeader("Content-Type", "text/plain"); res.end("Hello World\n");

});

server.listen(port, hostname, () => { console.log(`server running at http://${hostname}:${port}/`); });

* server running at <http://localhost:3000/>
* server running at port 3000
* server running at <http://localhost:4000/>
* server running at <http://127.0.0.1:3000/>

**Q25.What is the purpose of the path module?**

* to provide utilities to play with file and directory paths
* to provide utilities to add and remove files
* It is a retiring module.
* to provide utilities to test files

**Q26.How do you make an HTTP server object active and listen to requests on certain ports?**

* server. start
* server.activate
* server.listen
* server. run

**Q27.What does the code shown below do?**

const fs = require('fs'); const os = require('os');

const system = os.platform(); const user = os.userInfo().username;

fs.appendFile('hello.txt', `Hello ${user} on ${system}`, (err) => { if (err) throw err; console.log('The data was appended to file!');}

);

* creates a text file hello.txt and appends customized text
* creates an image file
* console logs system information
* creates a file named data and append numbers

**Q28.How do you start a Node application, if the entry file is indexjs?**

* nodemon start
* start index.js
* node index.js
* node start

**Q29.What is the purpose of the file system (fs) module?**

* to provide methods to work with requests and responses
* to provide methods to work with files
* to provide methods to work with databases
* to find new file systems

**Q30.What is the Node LTS version?**

* It is the current unstable version and is to be avoided.
* It is the version that will be retired soon.
* It is the version with the latest features.
* It is the safest version for long-term support.

**Q31. Which of the following is NOT a valid stream in Node?**

* process. stdinfo
* process. stdin
* process. stdout
* process. stderr

**Q32. You have a script.js file with the single line of code shown here. What will be the output of executing script.js with the node command?**

console.log(arguments);

* ReferenceError: arguments is not defined
* an empty string
* undefined
* an object representing an array that has five elements

**Q33. Which choice is not a valid method on event emitters?**

* start
* on
* once
* off

**Q34. Which special object is an instance of EventEmitter?Which special object is an instance of null?**

* process
* Buffer
* root
* require

**Q35. What is the command to get a list of available commands for Node.js?What is the command to get a list of available commands for Node.js?**

* node index.js -x
* node -v
* node -h
* node index.js -h

**Q36. When a request event is received in the HTTP module, what is the type of the first argument passed to that event, usually named req?**

* http.IncomingMessage
* http.ServerRequest
* http.ClientRequest
* http.ServerResponse

**Q37. What are the arguments passed to the module wrapper function?**

* exports, \_\_filename, \_\_dirname
* exports, process, require, module, \_\_filename, \_\_dirname
* exports, module, \_\_filename, \_\_dirname
* exports, require, module, \_\_filename, \_\_dirname

**Q38. Which library provides Node.js with the event loop?**

* V8
* c-ares
* libuv
* events

**Q39. What does the .node file extension represent?**

* a C++ file that can have a .node extension and that Node will be able to execute directly.
* a C++ Addon file that is built with node-gyp
* a JSON file that can have a .node extension as well as the .json extension
* a JavaScript file that can have a .node extension as well as the .js extension

**Q40. What can you export with module.exports?**

* only objects.
* only functions
* only variables and arrays
* functions, objects, arrays, or anything you assign to the module

**Q41. Which core module in Node can you use to take advantage of multicore systems?**

* os
* util
* cluster
* net

**Q42. Which core Node module has wrappers for OpenSSL methods?**

* SSL
* hash
* crypto
* TLS

**Q43. Which line imports a promise-based version of the readFile method?**

* const { readFile } = require(fs).promises
* const { readFile } = require(fs)
* const { readFilePromises: readFile } = require(fs)
* const { readFile } = require(promises)

**Q44. According to the rules of semantic versioning, what does a release incrementing the third number in an npm version string communicate to users about the release changes?**

* Changes are not backwards compatible.
* Changes might not be backward compatible and might break existing code.
* Changes are just bug fixes and no new features were added.
* Changes will add new functionality but will not break any existing code.

**Q45. What does REPL stand for?**

* run, examine, put, loop
* read, eval, print, loop
* run, edit, print, loop
* read, extend, print, loop

**Q46. Which file does node-gyp use to read the build configuration of a module?**

* .gyprc
* binding.gyp
* gyp.json
* package.gyp

**Q47. Which core module in Node can you use for testing?**

* chai
* jest
* assert
* mocha

**Q48. Which core module in Node provides an API to register callbacks to track asynchronous resources created inside a Node.js application?**

* cluster
* async\_hooks
* dgram
* inspector

**Q49. Which Node.js module should you use when you need to decode raw data into strings?**

* buffer
* util
* string\_decoder
* string\_buffer

**Q50. Which global object acts like a bridge between a Node script and the host operating system?**

* v8
* env
* process
* child\_process

**Explanation:** *process is an global object and act like a bridge, the others aren't (please see*[*https://nodejs.org/api/globals.html*](https://nodejs.org/api/globals.html)*or*[*https://nodejs.org/api/process.html#process\_process*](https://nodejs.org/api/process.html#process_process)*).*

**Q51. Which statement is true about Node.js and threads?**

* Every Node process runs in a single thread, and all the I/O work is run in that same thread.
* Every Node process gets four threads that it can share between its JavaScript VM and the event loop.
* The event loop is single-threaded, but a JavaScript VM can use multiple threads.
* JavaScript execution in Node.js is single-threaded, but I/O operations are executed using multiple threads.

**Explanation:** [*https://www.geeksforgeeks.org/why-node-js-is-a-single-threaded-language/*](https://www.geeksforgeeks.org/why-node-js-is-a-single-threaded-language/)

**Q52. Which statement about event emitters is false?**

* Event names must be camelCase strings.
* The emit method allows a arbitrary set of arguments to be passed to the listener functions.
* Any values returned by the listeners for an emitted events are ignored.
* When an event emitter objeect emits an event, all of the functions attached to that specific event are called synchronously.

**Q53. Which core module in Node can you use to compile and run JavaScript code in a sandbox environment?**

* sandbox
* buffer
* vm
* v8

**Q54. How would you determine the number of cluster instances to start when using the cluster module?**

* const numInstances = cluster.instances().length;
* const numInstances = cluster.instances();
* const numInstances = require('os').cpus().length;
* const numInstances = process.cpus().length;

**Explanation:** *From official docs:*[*https://nodejs.org/api/cluster.html#cluster\_cluster*](https://nodejs.org/api/cluster.html#cluster_cluster)

**Q55. You have to read a large text file, replace some words in it, and write it back to a new file. You know that the memory on your target system is limited. What should you do?**

* Use regular expressions directly on the file.
* Use Promises and async/await to offload the task to libuv.
* Copy the file into a database and perform the operations there.
* Use readline together with streams to read and transform and write the file contents line by line.

**Explanation:** *From official docs:*[*https://nodejs.org/api/readline.html#readline\_example\_read\_file\_stream\_line\_by\_line*](https://nodejs.org/api/readline.html#readline_example_read_file_stream_line_by_line)

**Q56. Which choice is not a Node global object?**

* process
* exports
* setTimeout
* Buffer

**Explanation:** *process, exports and setTimeout are global objects, Buffer isn't (please see*[*https://nodejs.org/api/globals.html*](https://nodejs.org/api/globals.html)*)*

**Q57. What is the correct way to pipe a readable stream and a writable stream?**

* readableStream.pipe(writableStream)
* readableStream.on(pipe, writableStream)
* writableStream.pipe(readableStream)
* writableStream.on(pipe, readableStream)

**Q58. How can you convert path segments into a string using the platform-specific separator as a delimiter?**

* path.concat
* path.join
* path.format
* path.parse

**Q59. What is the purpose of N-API?**

* to allow users to make requests to the server
* to insulate Addons from changes in the underlying JavaScript engine
* to execute multi-threaded code in the Node environment
* to provide a quick way for users to create REST APIs

**Q60. What is a process object and its role?**

* a locally scoped object that provides information about the current node process
* a global object that provides information about files
* a global object that provides information about the database
* a global object that provides information about the current node process

**Q61. What will this code log to the console?**

// File: person.js

exports.name = "Jane";

// File: index.js

const person = require('./person.js');

console.log(person);

* {'Jane'}
* { name: 'Jane' }
* {}
* Jane

**Q62. What will this code log to the console?**

// File: person.js

exports = "John";

// File: index.js

const person = require('./person.js');

console.log(person);

* John
* Undefined
* {'John'}
* {}

**Q63. Is it possible to write tests in Node.js without an external library?**

* yes, through the assert module
* yes, through the debugger module
* yes, through the console module
* no

**From the article:** [Making a Testing Framework in Node.js (Without any External Libraries)](https://www.sohamkamani.com/blog/javascript/making-a-node-js-test-runner/)

**Q64. Which assert module method is usually used to test the error-first argument in callbacks?**

* fail
* doesNotThrow
* deepStrictEqual
* ifError

**Q65. Which choice is not a method on the util module?**

* promisify
* asyncify
* types
* callbackify

**Q65. Which choice is not a subclass of the Error class?**

* GlobalError
* TypeError
* RangeError
* AssertionError