Node.js Interview Questions

### **1) What is Node.js?**

-> Node.js is Server-side scripting which is used to build scalable programs. It is a web application framework built on Google Chrome's JavaScript Engine. It runs within the Node.js runtime on Mac OS, Windows, and Linux with no changes. This runtime facilitates you to execute a JavaScript code on any machine outside a browser.

### **2) Is Node.js free to use?**

-> Yes. It is released under MIT license and is free to use.

### **3) Is Node a single threaded application?**

-> Yes. Node is a single-threaded application with event looping.

### **4) What is the purpose of Node.js?**

-> These are the following purposes of Node.js:

* Real-time web applications
* Network applications
* Distributed systems
* General purpose applications

### **5) What are the advantages of Node.js?**

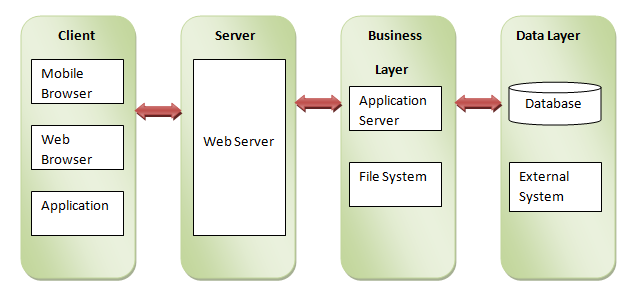
-> Following are the main advantages of Node.js:

* Node.js is asynchronous and event-driven. All API?s of Node.js library are non-blocking, and its server doesn't wait for an API to return data. It moves to the next API after calling it, and a notification mechanism of Events of Node.js responds to the server from the previous API call.
* Node.js is very fast because it builds on Google Chrome?s V8 JavaScript engine. Its library is very fast in code execution.
* Node.js is single threaded but highly scalable.
* Node.js provides a facility of no buffering. Its application never buffers any data. It outputs the data in chunks.

### **6) Explain Node.js web application architecture?**

-> A web application distinguishes into 4 layers:

* **Client Layer:**The Client layer contains web browsers, mobile browsers or applications which can make an HTTP request to the web server.
* **Server Layer:**The Server layer contains the Web server which can intercept the request made by clients and pass them the response.
* **Business Layer:**The business layer contains application server which is utilized by the web server to do required processing. This layer interacts with the data layer via database or some external programs.
* **Data Layer:**The Data layer contains databases or any source of data.



### **7) What do you understand by the term I/O?**

-> I/O stands for input and output. It accesses anything outside of your application. It loaded into the machine memory to run the program, once the application starts.

### **8) How many types of API functions are available in Node.js?**

-> There are two types of API functions in Node.js:

* Asynchronous, Non-blocking functions
* Synchronous, Blocking functions

### **9) What is error-first callback?**

-> Error-first callbacks are used to pass errors and data. If something goes wrong, the programmer has to check the first argument because it is always an error argument. Additional arguments are used to pass data.

1. fs.readFile(filePath, function(err, data) {
2. if (err) {
3. //handle the error
4. }
5. // use the data object
6. });

### **10) What is an asynchronous API?**

-> All the API's of Node.js library are asynchronous means non-blocking. A Node.js based server never waits for an API to return data. The Node.js server moves to the next API after calling it, and a notification mechanism of Events of Node.js responds to the server for the previous API call.

### **11) How can you avoid callbacks?**

-> To avoid callbacks, you can use any one of the following options:

* You can use **modularization**. It breaks callbacks into independent functions.
* You can use **promises**.
* You can use **yield** with Generators and Promises.

### **12) Does Node.js provide Debugger?**

-> Yes, Node.js provides a simple TCP based protocol and built-in debugging client. For debugging your JavaScript file, you can use debug argument followed by the js file name you want to debug.

#### **Syntax:**

1. node debug [script.js | -e "script" | **<host>**:**<port>**]

### **13) What is a control flow function?**

-> Control flow function is a generic piece of code that runs in between several asynchronous function calls.

### **14) How "Control Flow" controls the functions calls?**

->The control flow does the following job:

* Control the order of execution
* Collect data
* Limit concurrency
* Call the next step in a program

### **15) Is it possible to access DOM in Node?**

-> No, it is not possible to access DOM in Node.

### **16) What types of tasks can be done asynchronously using the event loop?**

->

* I/O operations
* Heavy computation
* Anything requiring blocking

### **17) What is REPL in Node.js?**

-> REPL stands for Read Eval Print Loop. It specifies a computer environment like a window console or Unix/Linux shell where you can enter a command, and the computer responds with an output.

REPL environment incorporates with Node.js.

### **18) Explain the tasks of terms used in Node REPL.**

-> Following are the terms used in REPL with their defined tasks:

**Read:** It reads user’s input; parse the input into JavaScript data-structure and stores in memory.

**Eval:** It takes and evaluates the data structure.

**Print:** It is used to print the result.

**Loop:** It loops the above command until user press ctrl-c twice to terminate.

### **19) Is it possible to evaluate simple expressions using Node REPL?**

-> Yes. You can evaluate simple expressions using Node REPL.

### **20) What is the use of the underscore variable in REPL?**

-> In REPL, the underscore variable is used to get the last result.

1. C:\Nodejs\_WorkSpace**>**node
2. **>** var x = 10
3. undefined
4. **>** var y = 20
5. undefined
6. **>** x + y
7. 30
8. **>** var sum = \_
9. undefined
10. **>** console.log(sum)
11. 30
12. undefined
13. **>**

### **21) Does Node.js supports cryptography?**

-> Yes, Node.js Crypto module supports cryptography. It provides cryptographic functionality that includes a set of wrappers for open SSL's hash HMAC, cipher, decipher, sign and verify functions. For example:

1. const crypto = require('crypto');
2. const secret = 'abcdefg';
3. const hash = crypto.createHmac('sha256', secret)
4. .update('Welcome to JavaTpoint')
5. .digest('hex');
6. console.log(hash);

### **22) What is npm? What is the main functionality of npm?**

-> npm stands for Node Package Manager. Following are the two main functionalities of npm:

* Online repositories for node.js packages/modules which are searchable on search.nodejs.org
* Command line utility to install packages, do version management and dependency management of Node.js packages.

### **23) What tools can be used to assure a consistent style in Node.js?**

-> Following is a list of tools that can be used in developing code in teams, to enforce a given style guide and to catch common errors using static analysis.

* JSLint
* JSHint
* ESLint
* JSCS

### **24) What is the difference between operational and programmer errors?**

-> Operational errors are not bugs, but create problems with the system like request timeout or hardware failure. On the other hand, programmer errors are actual bugs.

### **25) What is the difference between the global installation of dependencies and local installation of dependencies?**

->

* Global installation of dependencies is stored in /npm directory. While local installation of dependencies stores in the local mode. Here local mode refers to the package installation in node\_modules directory lying in the folder where Node application is present.
* Globally deployed packages cannot be imported using require() in Node application directly. On the other hand, locally deployed packages are accessible via require().
* To install a Node project globally -g flag is used.
  1. C:\Nodejs\_WorkSpace**>**npm install express ?g
* To install a Node project locally, the syntax is:
  1. C:\Nodejs\_WorkSpace**>**npm install express

### **26) What is the use of a buffer class in Node.js?**

-> The Node.js provides Buffer class to store raw data similar to an array of integers but corresponds to a raw memory allocation outside the V8 heap. It is a global class and can be accessed in an application without importing a buffer module. Buffer class is used because pure JavaScript is not compatible with binary data. So, when dealing with TCP streams or the file system, it's necessary to handle octet streams.

### **27) What is the role of assert in Node.js?**

-> The Node.js Assert is a way to write tests. It provides no feedback when running your test unless one fails. The assert module provides a simple set of assertion tests that can be used to test invariants. The module is intended for internal use by Node.js, but can be used in application code via require ('assert'). For example:

1. var assert = require('assert');
2. function add (a, b) {
3. return a + b;
4. }
5. var expected = add(1,2);
6. assert( expected === 3, 'one plus two is three');

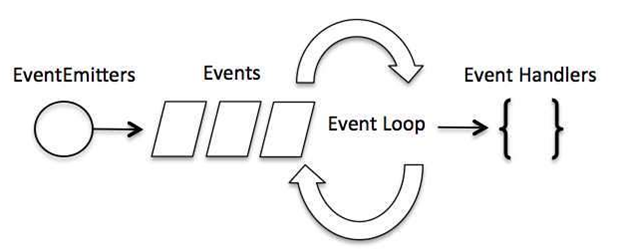
### **28) What are the streams in Node.js?**

-> The Streams are the objects that facilitate you to read data from a source and write data to a destination. There are four types of streams in Node.js:

* **Readable:**This stream is used for reading operations.
* **Writable:**This stream is used for write operations.
* **Duplex:**This stream can be used for both reading and write operations.
* **Transform:**It is a type of duplex stream where the output computes according to input.

### **29) What is event-driven programming in Node.js?**

-> In Node.js, event-driven programming means as soon as Node starts its server, it initiates its variables, declares functions and then waits for an event to occur. It is one of the reasons why Node.js is pretty fast compared to other similar technologies.



### **30) What is the difference between events and callbacks in Node.js?**

-> Although, Events and Callbacks look similar the differences lies in the fact that callback functions are called when an asynchronous function returns its result whereas event handling works on the observer pattern. Whenever an event gets fired, its listener function starts executing. Node.js has multiple in-built events available through the events module and EventEmitter class which is used to bind events and event listeners.

### **31) What is the Punycode in Node.js?**

-> The Punycode is an encoding syntax which is used to convert Unicode (UTF-8) string of characters to ASCII string of characters. It is bundled with Node.js v0.6.2 and later versions. If you want to use it with other Node.js versions, then use npm to install Punycode module first. You have to used require ('Punycode') to access it.

#### **Syntax:**

1. punycode = require('punycode');

### **32) What does Node.js TTY module contains?**

-> The Node.js TTY module contains tty.ReadStream and tty.WriteStream classes. In most cases, there is no need to use this module directly. You have to used require ('tty') to access this module.

#### **Syntax:**

1. var tty = require('tty');