Top Node.js interview Question and Answers for 2023

Description

Introduction

In this post we will go over node.js interview questions from most basic to advance level. These are some of the most widely asked questions when it comes to node.js beginner level to advanced level. Node.js has been a relief to Javascript developer specifically because, prior to node.js there was no provision to interact with databases, filesystem etc, when it comes to Javascript. Along with Node.js we can use various other modules to make a full fledged backend application.

What is Node.js and how it works?

Node.js is an *open-source*, *cross-platform* JavaScript *runtime environment* that executes JavaScript code outside of a web browser. Lets take a look at each term.

Open-source – Node.js has a large community of collaborators who work on keeping node.js feature rick and up-to-date.
Cross-platform – Node.js can run on Windows, Linux or MacOS.
Runtime environment – Node.js is built on top of Google Chrome's V8 engine and it provides everything that is needed to run Javascript outside the browser. This gives Node.js a stronger performance compared to other frameworks.

What is the difference between Javascript and Node.js

Below table shows the difference between Javascript and node.js

Javascript	Node.js
Javascript started of as a scripting language and evolved into a programming language containing object oriented and complex features.	Nodejs is a runtime environment for Javascript.
Javascript is more used at the client side.	Node.js is used for performing non blocking operations at Operating system level or database level.
The runtime engine is different for different browsers. V8 for Google Chrome, Spider monkey for Firefox, etc.	The runtime engine for Node.js is Google Chrome's V8 engine with an more capabilities.
Javascript can only be run inside the browsers.	Node.js can be run outside the browsers.

Javascript	Node.js
Javascript is an extended version of ECMA script which uses Google Chrome's V8 engine and written in C++.	Node.js is written in C, C++ and Javascript.

Is Node.js single threaded or multi threaded application

Node.js is single threaded and asynchronous which allows it to handle multiple requests to a server without the hassle of managing threads.

How does Node.js handle concurrency if it is single threaded.

The main loop in Node.js is single threaded and all async calls are managed by libuv library. Libuv sets up a thread pool to handle concurrency. The number of threads in the thread pool depends on the number of cores but we can override this.

What are the advantages of Node.js

- Using Node.js we can create server applications.
- Node.js is build on top of Google Chromes V8 engine hence Node.js has stronger performance compared to other frameworks.
- Node.js is single threaded and scalable in nature.
- Node.js is asynchronous and event-driven.
- All API's in Node.js are nonblocking and does not wait for the server to return data and moves on to work on the next task.

What kind of applications can be build using Node.js

- Traditional applications
- General purpose applications
- Backend services like API's
- Real time applications
- Streaming services
- CLI tools
- Multiplayer games

What is Event loop in Node.js

- The Event loop is the backbone of Node.js.
- It is basically a C program which is part of the libuv.
- Whatever is async is mostly managed by the event-loop using a queue and listener.
- Event loop is design pattern that orchestrates or coordinates the execution of synchronous and asynchronous code in Node.js

- If there is any async function to be executed, the main thread will send it to a different thread and v8 engine will keep executing the main code
- Event loop is a loop that is alive as long as the node.js application is up and running.
- In every iteration of the loop we come across 6 different queues. Each queue hold one or more callback function which needs to be executed on the call stack. Time queue, I/O queue, check queue, close queue, nextTick queue, promise queue are 6 different types of queue.
- Event loop solves the problem of multi-threading.
- Event loop can perform otherwise blocking operations in a non-blocking manner.
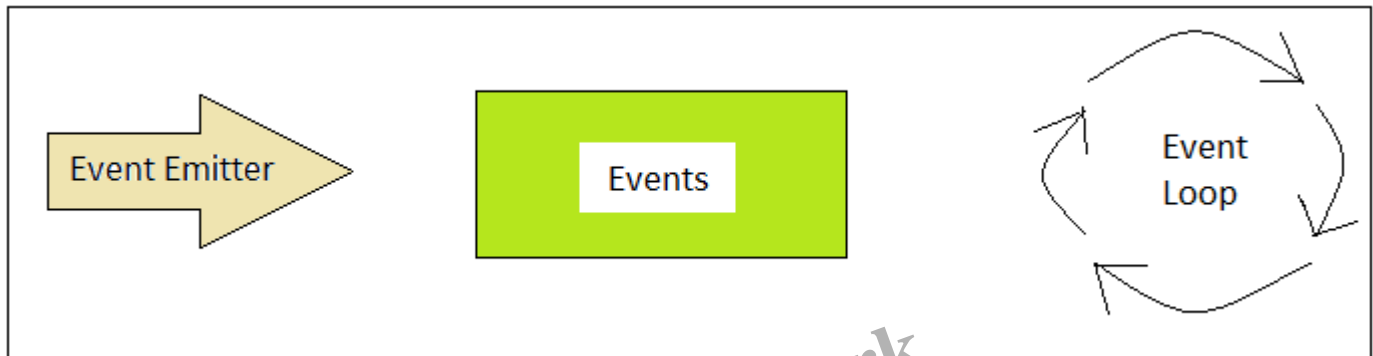
Node.js Event Loop

What are first class functions in Javascript

First class functions in Javascript are the functions which we can assign it to a variable and pass it as parameters to other function (callback) or a function can return another function which is a higher-order function. Some examples of higher order functions are map and filter.

What does Javascript runtime consist of

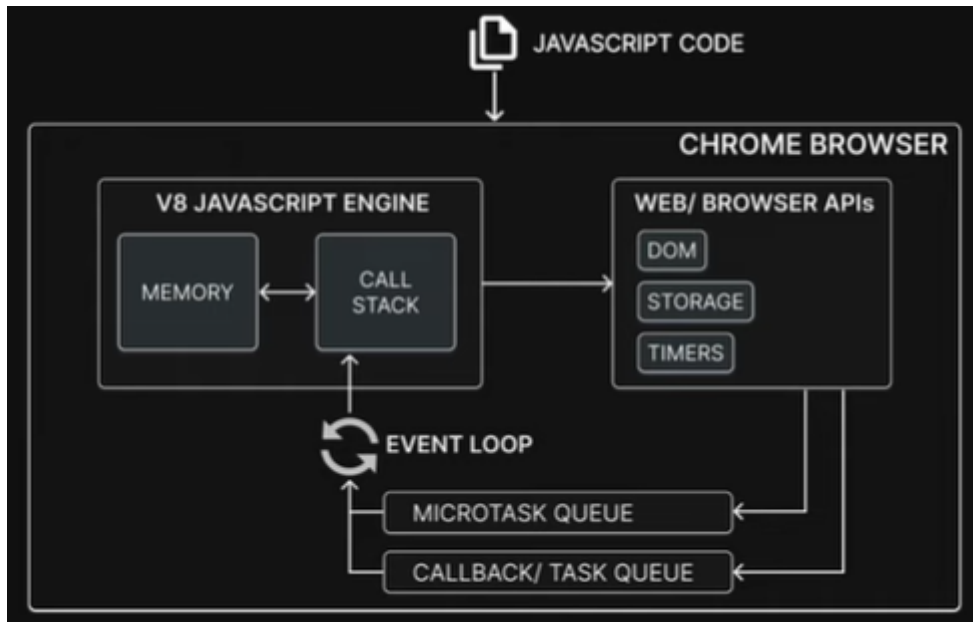Below is the diagram of Javascript runtime which consists of various components.

Image Credit – Codevolution

What does node.js runtime consists of
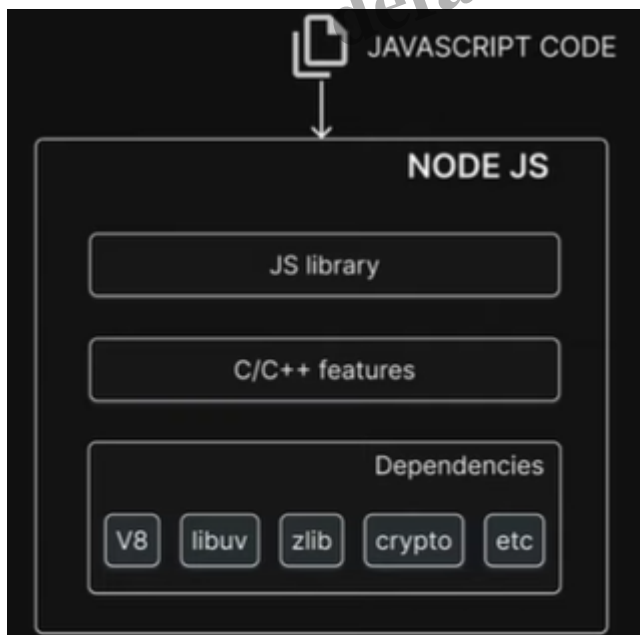
Node.js runtime consists of below things

Image credits – Codevolution

How to manage packages in Node.js

NPM and yarn are mostly used to download and manage the required packages for Node.js. Configuration files such as **package.json** and **package-lock.json** are used to maintain the

dependencies and the required versions.

What is callback hell in Node.js

Callback hell in node.js occurs when Javascript developer tries to execute multiple asynchronous operations one after the other. Below image shows an example of callback hell.

```
fetchCurrentUser(`api/user`, function(result) {
    fetchFollowersByUserId(`api/followers/${result.userId}`, function(result){
        fetchFollowerInterests(`api/interests/${result.followerId}`, function(result) {
            fetchInterestTags(`api/tags/${result.interestId}`, function(result) {
                fetchTagDescription(`api/description/${result.tagId}`, function(result) {
                    // Finally display the data
                })
            })
        })
    })
})
```

Callback hell can be avoided using promises.

What are the commonly used time related functions in Node.js

Below are commonly used time related functions in node.js

- **setInterval** – setInterval function is used to set execution of the code after a certain time period.
- **clearInterval** – clearInterval function is used to clear the time interval that was set.
- **setTimeout** – setTimeout function is used to implement delays in the code execution.
- **clearTimeout** – clearTimeout function is used to clear the timeout that was set.

How to use async await in Node JS

The Async await keywords allows us to write completely synchronous looking code while performing asynchronous tasks behind the scenes.

await keyword can be put in front of any async promise based function to pause your code until that promise settles and returns its results.

await keyword only works inside async functions. We cannot use await in normal functions.

What do you mean by modules in Node JS

Module is an encapsulated and reusable chunk of code which has its own context. Each module in Node.js serve a specific purpose. Each separate file is treated as a module in Node.js. The most common use of modules is to use code or functions across different modules. This allows us to keep our code organized, as each module can serve a specific purpose. **'require'** is used to import modules, JSON and local files.

Footer Tagline

```
const square = require('./square.js')
```

If the file is present in the current directory then we use dot slash before the filename.

Modules can also be exported so that other modules can reuse the module functions.

```
module.exports.addition= (a,b) => a+b;
```

What are different type of modules

Local modules – Modules we can create in our application.
Build-in modules – Modules that node.js ships with out of the box. **Example** – path, events, streams, fs, http.
Third party modules – Modules written by other developers that we can use in our application.

What are different type of modules present in Node.js

There are different types of modules present in Node.js. Below are some of the core modules provided by Node.js

Module Name	Module Description
path	path module in node.js provides utilities for working with file and directory paths.
fs	fs module in node.js provides methods to perform I/O operation.
http	http module contains classes, methods and events to create node.js server.
stream	stream module contains methods to handle streaming data.
util	util module contains utility functions required in the application which is helpful for developers.
zlib	zlib module contains methods to compress and decompress files used in an application.

What is CommonJS

CommonJS is a standard that states how a module should be structured and shared. Node.js adopted CommonJS when it started out and is what you will see in the code bases.

What is fs module in Node.js

The fs module also known as file system module allows you to work with file system on your computer. fs is a built id module that comes with Node.js. We can import the fs module as shown below.

```
const fs = require('fs')
```

There are two methods in fs to read a file i.e. fs.readFileSync (for synchronous operation) and fs.readFile (for asynchronous operation). If there are lot of concurrent users and file size is large then its better to use readFile.

Similar to read operation, there are two methods available for writing the contents to the file i.e. writeFileSync and writeFile.

What is http module is Node.js

HTTP module is a built in node.js module. Using http module we can create web server that can transfer data over HTTP. In order to use the HTTP module, we can import as below.

```
const http = require("http")
```

In order to create the server, we call createServer method on the http module. It takes in a callback function as input.

```
http.createServer((req, res) => {}
```

What is promise based fs in Node.js

We can write file operation code either using promise based or async await. The module can be imported like below

```
const fs = require("fs/promises")
```

Usually a callback version of the node fs module APIs are preferable over the promise based APIs when maximum performance is required both in terms of execution time and memory allocation. If that is not much concern then we can use promise based fs module also.

What are buffers in Node.js

Buffer in general terms refer to a temporary data storage that is used when moving the data. In case of video streaming websites like YouTube, the video data is fetched from the server, stored in the *buffer*, processed, and sent to your display. As more data is fetched, the buffer gets filled with newer data and old data is usually overwritten. If the data is not there in the buffer you might see a loading sign.

Node.js has a Buffer class that is based on JavaScriptâ€™s Uint8Array. Buffer objects as arrays that only contain integers from 0 to 255. The Buffer class is in the **global** definition, so we do not need to

import anything.

```
const buffer1 = Buffer.alloc(10);
console.log(buffer1);

const buffer2 = Buffer.from([210, -210, 8.9, '11']);
console.log(buffer2);

const buffer3 = Buffer.from('Hello Youtube');
console.log(buffer3);

console.log(buffer3.toString());
```

```
Output

<Buffer 00 00 00 00 00 00 00 00 00 00>
<Buffer d2 2e 08 0b>
<Buffer 48 65 6c 6c 6f 20 59 6f 75 74 75 62 65>
Hello Youtube
```
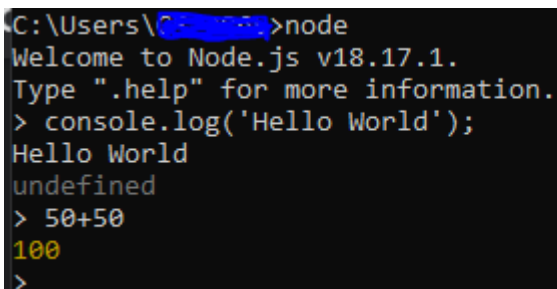
How to avoid callbacks

In order to avoid callbacks we can make use of **promises**. Promise is used to get an object after which it decides the action that needs to be taken after the async task completes. Hence promises are used to avoid callback hell.

Can we access DOM in Node

No. We cannot access DOM in Node.

What is REPL in Node JS

REPL in Node.js stands for Read Evaluate Print Loop. REPL is like a command line environment in which we can provide instructions and it will give us the result. In order to initiate the REPL environment we need to hit **node**. Below is the example of REPL

```
C:\Users\         >node
Welcome to Node.js v18.17.1.
Type ".help" for more information.
> console.log('Hello World');
Hello World
undefined
> 50+50
100
>
```

What is NPM

Node.js stands for Node Package Manager. NPM is like a repository or registry and managed by NPM Inc. It enables all 3rd party developers to publish to their repository. Developers can mention the dependency they want inside their package.json file and it will download the required dependencies and make it available in your project.

Node.js has over 1,000,000 open-source packages hosted on npm. This has helped Node.js grow drastically. It is often easier to use a package rather than writing code from scratch. Code reusability has enhanced greatly with the use of packages.

Which tools can be used to ensure consistent style in Node.js

Below are the tools that can be used to ensure consistent style in node.js.

- JSHint
- ESLint
- JSLint
- JSCS

What is the difference between global installation of dependencies and local installation of dependencies

- Global installable dependencies are stores in npm directory. Local installable dependencies are stored in local mode. local mode refers to packages installed in node_modules directory which is present in the created node application.
- Locally deployed packages are accessible via require(). Globally deployed packages cannot be imported via require().
- To install a node project locally we need to run below command.

```
npm install lodash
```

- To install a node project from globally we need to run below command.

```
npm install lodash ?g
```

What are streams in Node.js

Footer Tagline

Streams are objects which assist to read data from the source and write data to the destination. Inother words a Stream is a sequence of data that is being moved from one point to another point overtime.

Example – A stream of data moved from one computer to another over the internet. A stream of data being transferred from one file to another with the same computer.

There are four types of streams.

- **Readable:** Readable stream is used for read operations. **E.g.** Reading from file as a readable stream.
- **Writable:** Writable stream is used for write operations. **E.g.** Writing to a file as writable stream.
- **Duplex:** Duplex stream can be used for both read and write operations. **E.g.** Sockets as duplex stream.
- **Transform:** Transform stream takes input and compute to a specific output. **E.g.** File compression to compress data while writing and decompress data while reading from file as transform stream.

What are Events in Node.js

Events in Node.js is very important because Node.js is mostly built around an asynchronous, event-driven architecture. Core APIs such as **'fs'** module , **'net'** module uses events to communicate with the program. In case of **'fs'** module, when it is ready to provide data to be read, it signals or emits an event.

What are EventEmitters in Node.js

EventEmitter is a class used to emit events. EventEmitter belongs to 'events' module. Here we attach one or more functions to a specific event which gets executed when an event occurs. These *attached* functions are also called **listeners**, as they *listen* for events to be emitted. Furthermore, listener functions are called synchronously once the object emits a specific event. This makes sure that the sequence of events is preserved and no race conditions or logic errors arise. If you want to attach a function to an event, then we use 'on' function and if we want to sent an event then we use 'emit' function.

```
const EventEmitter = require('events');
const eventEmitter = new EventEmitter();

listenerFunction = function (){
  console.log('Wohooo! Finally Event has occurred');
}
eventEmitter.on('event', listenerFunction);
eventEmitter.emit('event');
```

Footer Tagline

```
Output

Wohooo! Finally Event has occurred
```

What is punnycode in Node.js

Punnycode in Node.js is used to convert Unicode (UTF-8) string of characters to ASCII string of characters. in order to use punnycode module we need to install using npm first and then require in your code.

```
punycode = require('punycode');
```

Describe the exit codes of Node.js

Below are some of the exit codes of Node.js

Error Number	Error Code	Description
0	N/A	Node.js will normally exit with a 0 status code when no more async operations are pending.
1	Uncaught Fatal Exception	There was an uncaught exception, and it was not handled by a domain or an 'uncaughtException' event handler.
2	N/A	Reserved by Bash for builtin misuse
3	Internal JavaScript Parse Error	The JavaScript source code internal in Node.js's bootstrapping process caused a parse error. This is extremely rare, and generally can only happen during development of Node.js itself.

Click here to find more exit error codes

What is a thread pool and which library handles it in Node.js

Thread pool in Node.js is handled by libuv library. libuv is a multi-platform C library that provides support for asynchronous I/O-based operations such as file systems, networking, and concurrency.

Category

1. Interview

Date Created
September 14, 2023
Author
kk-ravi144gmail-com