

before the page is sent to the user's web browser. Don't go to your next Nodejs interview without checking that list of 19 most advanced Node js interview questions that may get you your next full stack developer job offer!

## Q1: What is Node.js?

Entry



Node.js 88

### Answer

Node.js is a web application framework built on Google Chrome's JavaScript Engine (V8 Engine).

Node.js comes with runtime environment on which a Javascript based script can be interpreted and executed (It is analogous to JVM to JAVA byte code). This runtime allows to execute a JavaScript code on any machine outside a browser. Because of this runtime of Node.js, JavaScript is now can be executed on server as well.

*Node.js = Runtime Environment + JavaScript Library*

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: [tutorialspoint.com](https://www.tutorialspoint.com)

## Q2: What are the benefits of using Node.js?

Junior



Node.js 88

### Answer

Following are main benefits of using Node.js

- **Asynchronous and Event Driven** - All APIs of Node.js library are asynchronous that is non-blocking. It essentially means a Node.js based server never waits for a API to return data. Server moves to next API after calling it and a notification mechanism of Events of Node.js helps server to get response from the previous API call.

**Single Threaded but highly Scalable** - Node.js uses a single threaded model with event looping.

Event mechanism helps server to respond in a non-blocking ways and makes server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and same program can services much larger number of requests than traditional server like Apache HTTP Server.

- **No Buffering** - Node.js applications never buffer any data. These applications simply output the data in chunks.

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: tutorialspoint.com

### Q3: What is an error-first callback?

Junior



Node.js 88

#### Answer

*Error-first callbacks* are used to pass errors and data. The first argument is always an error object that the programmer has to check if something went wrong. Additional arguments are used to pass data.

```
fs.readFile(filePath, function(err, data) {  
  if (err) {  
    //handle the error  
  }  
  // use the data object  
});
```

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: tutorialspoint.com

### Q4: What is global installation of dependencies?

Junior



Node.js 88

#### Answer

Globally installed packages/dependencies are stored in <user-directory>/npm directory. Such dependencies can be used in CLI (Command Line Interface) function of any Node.js but can not be imported using `require()` in Node application directly. To install a Node project globally use `-g` flag.

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: tutorialspoint.com

## Café

**Answer**


By providing callback function. Callback function gets called whenever corresponding event triggered.

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: tutorialspoint.com



## 35 Top Angular 7 Interview Questions To Crack in 2019

 Angular 120

### Q6: If Node.js is single threaded then how it handles concurrency?

Mid

 Node.js 88

**Answer**

Node provides a single thread to programmers so that code can be written easily and without bottleneck. Node internally uses multiple POSIX threads for various I/O operations such as File, DNS, Network calls etc.

When Node gets I/O request it creates or uses a thread to perform that I/O operation and once the operation is done, it pushes the result to the **event queue**. On each such event, **event loop** runs and checks the queue and if the execution stack of Node is empty then it adds the queue result to execution stack.


This is how Node manages concurrency.

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: codeforgeek.com

### Q7: What is Event Emmitter?

Mid

 Node.js 88

**Answer**

All objects that emit events are members of EventEmitter class. These objects expose an `eventEmitter.on()` function that allows one or more functions to be attached to named events emitted by the object.

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

class MyEmitter extends EventEmitter {}

const myEmitter = new MyEmitter();
myEmitter.on('event', () => {
  console.log('an event occurred!');
});
myEmitter.emit('event');
```

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: [tutorialspoint.com](https://www.tutorialspoint.com)

## Q8: What's the Event Loop?

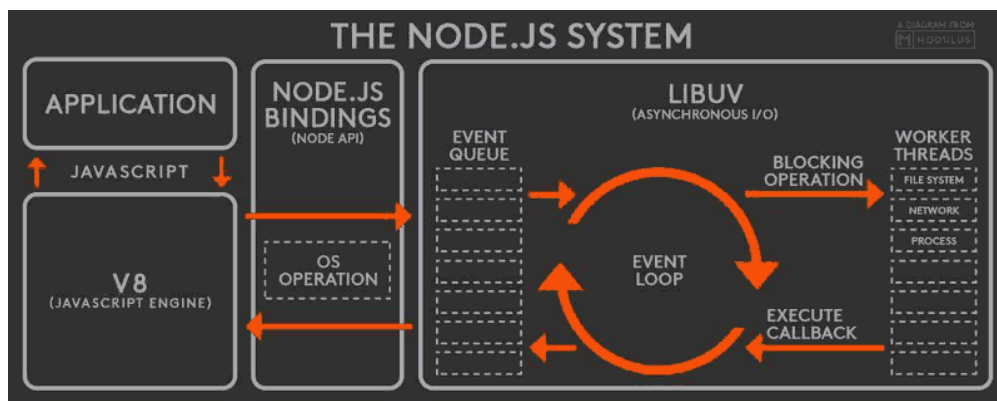
Mid



Node.js 88

### Answer




The **event loop** is what allows Node.js to perform non-blocking I/O operations — despite the fact that JavaScript is single-threaded — by offloading operations to the system kernel whenever possible.



Every I/O requires a callback - once they are done they are pushed onto the event loop for execution. Since most modern kernels are multi-threaded, they can handle multiple operations executing in the background. When one of these operations completes, the kernel tells Node.js so that the appropriate callback may be added to the poll queue to eventually be executed.


Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: [blog.risingstack.com](https://blog.risingstack.com)

 **Having Data Science & ML Interview?** Check  **MLStack.Cafe** - 1299 Data Science & ML Interview Questions & Answers! **Having ML & DS Interview?** Check  **MLStack.Cafe** - 1299 ML & DS Interview Questions and Answers

## Q9: Explain usage of `NODE_ENV`

Senior

 Node.js 88

### Answer

Node encourages the convention of using a variable called `NODE_ENV` to flag whether we're in production right now. This determination allows components to provide better diagnostics during development, for example by disabling caching or emitting verbose log statements. Setting `NODE_ENV` to production makes your application roughly 3 times faster.

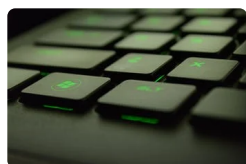
```
// Setting environment variables in bash before starting the node process
$ NODE_ENV=development
$ node
```

And reading variable:

```
if (process.env.NODE_ENV === "production")
  useCaching = true;
```

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: [github.com/i0natan/nodebestpractices](https://github.com/i0natan/nodebestpractices)



## 29 T-SQL Interview Questions Devs Must Know (ANSWERED for 2020)



T-SQL 51



## The separation for dev and prod environments

### Answer

A perfect and flawless configuration setup should ensure:

- keys can be read from file AND from environment variable
- secrets are kept outside committed code
- config is hierarchical for easier findability

Consider the following config file:

```
var config = {
  production: {
    mongo : {
      billing: '****'
    }
  },
  default: {
    mongo : {
      billing: '****'
    }
  }
}

exports.get = function get(env) {
  return config[env] || config.default;
}
```

And it's usage:

```
const config = require('./config/config.js').get(process.env.NODE_ENV);
const dbconn = mongoose.createConnection(config.mongo.billing);
```

Having Tech or Coding Interview? Check [🔗 88 Node.js Interview Questions](#)

Source: [github.com/i0natan/nodebestpractices](https://github.com/i0natan/nodebestpractices)

## Q11: What are the Timing features of Node.js?

Senior



Node.js 88

### Answer

The **Timers** module in Node.js contains functions that execute code after a set period of time.

- `setInterval/clearInterval` - can be used to execute a block of code multiple times
- `setImmediate/clearImmediate` - will execute code at the end of the current event loop cycle
  - `process.nextTick` - used to schedule a callback function to be invoked in the next iteration of the Event Loop

```
function cb() {  
  console.log('Processed in next iteration');  
}  
process.nextTick(cb);  
console.log('Processed in the first iteration');
```

Output:

```
Processed in the first iteration  
Processed in next iteration
```

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: [github.com/jimuyouyou](https://github.com/jimuyouyou)

## Q12: What is LTS releases of Node.js why should you care?

Senior



Node.js 88

### Answer

An **LTS(Long Term Support)** version of Node.js receives all the critical bug fixes, security updates and performance improvements.

LTS versions of Node.js are supported for at least 18 months and are indicated by even version numbers (e.g. 4, 6, 8). They're best for production since the LTS release line is focussed on stability and security, whereas the *Current* release line has a shorter lifespan and more frequent updates to the code. Changes to LTS versions are limited to bug fixes for stability, security updates, possible npm updates, documentation updates and certain performance improvements that can be demonstrated to not break existing applications.

Having Tech or Coding Interview? Check  88 Node.js Interview Questions

Source: [github.com/i0natan/nodebestpractices](https://github.com/i0natan/nodebestpractices)

## Q13: What tools can be used to assure consistent code style?

Senior



JavaScript 142

- JSLint by Douglas Crockford
- JSHint
- ESLint
- JSCS

These tools are really helpful when developing code in teams, to enforce a given style guide and to catch common errors using static analysis.

Having Tech or Coding Interview? Check  142 JavaScript Interview Questions

Source: [blog.risingstack.com](https://blog.risingstack.com)



## 25+ JavaScript Coding Interview Questions (SOLVED with CODE)



JavaScript 142

### Q14: Consider following code snippet

Expert



Node.js 88

#### Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to open Expert question!



### Q15: Explain what is Reactor Pattern in Node.js?

Expert



Node.js 88

#### Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to open Expert question!





 Ruby on Rails 72 OOP 54 Redis 25 Flutter 68 PHP 82 Kotlin 68 HTML5 55 Ionic 29 Design Patterns 45 Agile & Scrum 47

Having Data Science &amp; ML Interview? Check



MLStack.Cafe - 1299 Data Science

&amp; ML Interview Questions &amp; Answers! Having ML &amp; DS Interview? Check



MLStack.Cafe - 1299 ML &amp; DS Interview Questions and Answers

## Q16: Rewrite the code sample without `try/catch` block

Expert



Node.js 88

### Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to open Expert question!



## Q17: What is the difference between `process.nextTick()` and `setImmediate()` ?

Expert



Node.js 88

### Answer

Unlock **FullStack.Cafe** to open all answers and get your next figure job offer!

Share this blog post to open Expert question!

