NodeJS Interview Question

Q) What do you mean by I/O?

I/O stands for input/output, which helps write and read files and network operations.

Q) Can we run Node.js on Windows?

Yes, it is possible to run Node.js on Windows.

Q) Which extension is used to save Node.js files?

.js file extension

Q) What is Node red?

Node red is a visual programming tool for Node.js that is used to wire hardware devices and online services as part of IoT applications.

Q) How is operational error different from programming error?

An operational error occurs naturally and is part of the application flow, while programming errors are referred to as bugs that are caused by developers.

Q) What is the blocking code?

Blocking code is code that cannot be executed until the current code is completely executed.

Q) What is setTimeout() Function ?

In Node.js, the `setTimeout()` function is a part of the JavaScript runtime and is used to schedule the execution of a function after a specified delay. It allows you to introduce a delay in the execution of a particular piece of code, making it useful for scenarios such as implementing timeouts, delaying the execution of asynchronous tasks, or creating intervals between function calls.

Here's a basic syntax of the `setTimeout()` function:

```
setTimeout(callback, delay, arg1, arg2, ...);
```

- `callback`: The function to be executed after the specified delay.
- `delay`: The delay (in milliseconds) before the callback function is executed.
- `arg1, arg2, ...`: Optional arguments to pass to the callback function.

Here's an example of using `setTimeout()` in Node.js:

```
function sayHello() {
  console.log("Hello, world!");
}

// Schedule the sayHello function to be called after 2000
milliseconds (2 seconds)
setTimeout(sayHello, 2000);
```

Q) How is JavaScript different from Node.js?

JavaScript is a programming language mainly used for making websites interactive. It runs in web browsers and allows you to manipulate webpage content, respond to user actions, and enhance the user experience.

Node.js is a runtime environment that allows you to run JavaScript on servers instead of just in web browsers. It's used for building server-side applications and handling tasks like processing data, managing databases, and serving web pages.

- JavaScript is the language, and Node.js is the environment that lets you run JavaScript on servers.
- Learning JavaScript is essential for both front-end (browser) and back-end (server) development.

Q) Is Node.js single or multi-threaded?

Node.js is single-threaded.

Q) Then How nodeJS Handle Multiple Task at once ?

Imagine you have a to-do list, and you're the only person who can complete the tasks on that list. In the world of Node.js, it's similar – there's one main "worker" (like you) that handles all the tasks.

Being "single-threaded" means that this worker can only do one thing at a time. It's like you working through your to-do list one task at a time, without being able to do multiple tasks simultaneously.

Now, you might wonder, "How does Node.js handle many users and tasks if it can only do one thing at a time?"

Here's the clever part: Node.js is good at handling tasks that involve waiting, like reading files or making network requests. While it's working on one task, it can "pause" that task and move on to the next one. When the paused task is ready to continue, Node.js goes back to it. This makes it seem like it's doing many things at once, even though it's still technically one worker.

Think of it like this: You're waiting for a cake to bake in the oven. Instead of just staring at the oven, you decide to start setting the table or doing other tasks. When the cake is ready, you go back to it. Node.js does something similar, juggling different tasks while waiting for certain things to complete.

This approach is especially useful for handling a large number of simultaneous requests in web servers, making Node.js efficient for certain types of applications. However, it's essential to note that not all tasks can be easily divided and managed this way, and the single-threaded nature of Node.js has its trade-offs depending on the type of application you're building.

Q) How to obtain the IP address of the user in Node.js?

We use req.connection.remote address to get the IP address.
Sometimes we also use this

```
const clientIP = req.headers['x-forwarded-for'] ||
req.connection.remoteAddress;
```

Keep in mind that the `remoteAddress` might not always be the actual IP address of the client due to factors like proxy servers. If your application is behind a proxy, you might need to use other headers like `X-Forwarded-For` to get the actual client IP address.

Q) How to install the Node body-parser module?

Run the command “npm install body-parser” to install the body-parser module and add it to your project's dependencies.

Q) What is the use of Nodemon?

`Nodemon` (short for Node Monitor) is a utility for Node.js that helps in the development process by monitoring for changes in files and automatically restarting the Node.js application when changes are detected.

Q) What is the call-back function used for?

The call-back function is used to execute a function after a certain event has occurred.

Q) What is the function of the fs module?

The fs module is used to create and manipulate files. It also provides functions for interacting with the file system.

Q) Define os module in Node.js?

The os module provides a set of tools for interacting with the operating system. It provides an functions for getting information about the system including memory, processor, file system, and network interfaces.

Q) What is the meaning of HTTP status code 504?

HTTP status code 504 indicates that the server is unable to process the request. This can be due to several reasons, such as an overloaded server or a network issue.