**How else can the JavaScript code below be written using Node.Js to produce the same output?**

console.log("first");

setTimeout(function() {

console.log("second");

}, 0);

console.log("third");

Output:

first

third

second

In Node.js version 0.10 or higher, setImmediate(fn) will be used in place of setTimeout(fn,0) since it is faster. As such, the code can be written as follows:

console.log("first");

setImmediate(function(){

console.log("second");

});

console.log("third");

Why is Node.js Single-threaded?

Node.js is single-threaded for async processing. By doing async processing on a single-thread under typical web loads, more performance and scalability can be achieved as opposed to the typical thread-based implementation.

### Explain callback in Node.js.

A callback function is called after a given task. It allows other code to be run in the meantime and prevents any blocking.  Being an asynchronous platform, Node.js heavily relies on callback. All APIs of Node are written to support callbacks.

### Explain the role of REPL in Node.js.

As the name suggests, REPL (Read Eval Print Loop) performs the tasks of – Read, Evaluate, Print and Loop. The REPL in Node.js is used to execute ad-hoc Javascript statements. The REPL shell allows entry to javascript directly into a shell prompt and evaluates the results. For the purpose of testing, debugging, or experimenting, REPL is very critical.

### Name the types of API functions in Node.js.

There are two types of functions in Node.js.:

* + **Blocking functions** - In a blocking operation, all other code is blocked from executing until an I/O event that is being waited on occurs. Blocking functions execute synchronously.

For example:  
const fs = require('fs');  
const data = fs.readFileSync('/file.md'); // blocks here until file is read  
console.log(data);  
// moreWork(); will run after console.log  
  
The second line of code blocks the execution of additional JavaScript until the entire file is read. moreWork () will only be called after Console.log

* + **Non-blocking functions** - In a non-blocking operation, multiple I/O calls can be performed without the execution of the program being halted. Non-blocking functions execute asynchronously.

For example:

const fs = require('fs');  
fs.readFile('/file.md', (err, data) => {  
  if (err) throw err;  
  console.log(data);  
});  
// moreWork(); will run before console.log

Since fs.readFile () is non-blocking, moreWork () does not have to wait for the file read to complete before being called. This allows for higher throughput.

1. Which is the first argument typically passed to a Node.js callback handler?

Typically, the first argument to any callback handler is an optional error object. The argument is null or undefined if there is no error.

Error handling by a typical callback handler could be as follows:

function callback(err, results) {  
    // usually we'll check for the error before handling results  
    if(err) {  
        // handle error somehow and return  
    }  
    // no error, perform standard callback handling  
}

1. What are the functionalities of NPM in Node.js?

NPM (Node Package Manager) provides two functionalities:

* 1. An online repository for Node.js packages.
  2. Command-line utility for installing packages, version management and dependency management of Node.js packages.

1. What is the difference between Node.js and Ajax?

Node.js and Ajax (Asynchronous JavaScript and XML) are the advanced implementations of JavaScript. They all serve entirely different purposes.

Ajax is primarily designed for dynamically updating a particular section of a page’s content, without having to update the entire page.

Node.js is used for developing client-server applications.

1. Explain chaining in Node.js.

Chaining is a mechanism whereby the output of one stream is connected to another stream creating a chain of multiple stream operations.

### ****1. Differentiate between JavaScript and Node.js.****

|  |  |  |
| --- | --- | --- |
| **Features** | **JavaScript** | **Node.js** |
| Type | Programming Language | Interpreter and environment for JavaScript |
| Utility | Used for any client-side activity for a web application | Used for accessing or performing any non-blocking operation of any operating system |
| Running Engine | Spider monkey (FireFox), JavaScript Core (Safari), V8 (Google Chrome), etc. | V8 (Google Chrome) |

### ****List down the major benefits of using Node.js?****

|  |  |
| --- | --- |
| **Features** | **Description** |
| ***Fast*** | Node.js is built on Google Chrome’s V8 JavaScript Engine which makes its library very fast in code execution |
| ***Asynchronous*** | Node.js based server never waits for an API to return data thus making it asynchronous |
| ***Scalable*** | It is highly scalable because of its event mechanism which helps the server to respond in a non-blocking way |
| ***Open Source*** | Node.js has an extensive open source community which has contributed in producing some excellent modules to add additional capabilities to Node.js applications |
| ***No Buffering*** | Node.js applications simply output the data in chunks and never buffer any data |

**8. How many types of API functions are there in Node.js?**

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

* Asynchronous, non-blocking functions
* Synchronous, blocking functions

**What is the difference between Asynchronous and Non-blocking?**

|  |  |
| --- | --- |
| **Asynchronous** | **Non-blocking** |
| Asynchronous means not synchronous. Using these we can make asynchronous HTTP requests that do not wait for the server to respond. These functions continue to respond to the request for which it has already received the server response. | Non-blocking functions are used in regards with I/O operations. They immediately respond with whatever data is available and keeps on running as per the requests. In case, any answer couldn’t be retrieved then the API returns immediately with an error. |

### ****11. What do you understand by Event-driven programming?****

Event-driven programming is a programming approach that heavily makes use of events for triggering various functions. An event can be anything like a mouse click, key press, etc. When an event occurs, a call back function is executed that is already registered with the element. This approach mainly follows the publish-subscribe pattern. Because of [event-driven programming](https://www.edureka.co/blog/nodejs-tutorial/#events), Node.js is faster when compared to other technologies.

### ****13. Explain  REPL in the context of Node.js.****

REPL in Node.js stands for **R**ead, **E**val, **P**rint, and **L**oop. It represents a computer environment such as a window console or Unix/Linux shell where any command can be entered and then the system can respond with an output. Node.js comes bundled with a REPL environment by default. REPL can perform the below-listed tasks:

* **Read:** Reads the user’s input, parses it into JavaScript data-structure and then stores it in the memory.
* **Eval:** Receives and evaluates the data structure.
* **Print:**Prints the final result.
* **Loop:** Loops the provided command until *CTRL+C* is pressed twice.

### ****14. List down the tasks which should be done asynchronously using the event loop?****

Below is the list of the tasks which must be done asynchronously using the event loop:

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

### ****15. List down the steps using which “Control Flow” controls the function calls in Node.js?****

1. Control the order of execution
2. Collect data
3. Limit concurrency
4. Call the next step in the program

### ****[2) Explain CLI in Node.js?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled2)****

CLI stands for Command Line Interface. It is a utility or program on your computer where users type commands to perform some action or run some script rather than clicking on the screen.  
There are different types of command line interfaces depending on which operating system you are using. We have listed some of them below.

* **[9) Explain What is NPM ?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled9)**
* NPM stands for node package manager. It is default Package Manager for JavaScript programming language. NPM is used for installing/updating packages and modules of Javascript.

**[10) Explain Modules in Node Js ?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled10)**

**Modules** are reusable block of code whose existence does not impact other code in any way. It is not supported by Javascript. Modules are introduced in ES6. Modules are important for Maintainability, Reusability, and Namespacing of Code.

**[12) For what require() is used in Node Js ?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled12)**

**require()** is used to include modules from external files in Node Js. It is the easiest way to include a module in Node. Basically require is a function that takes a string parameter which contains the location of the file that you want to include. It reads the entire javascript file, executes the file, and then proceeds to return the exports object.  
Syntax:

require('path')

**[14) Is Node Js Single-threaded ?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled14)**

Yes, Node Js is single threaded to perform asynchronous processing. Doing async processing on a single thread could provide more performance and scalability under typical web loads than the typical thread-based implementation.

**[17) How to create a simple server in Node js that returns Hello World ?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled17)**

By writing following line of code, you can **create a server in Node Js**.

var http =require('http');

http.createServer(function(req,res){

res.writeHead(200,{'Content-Type':'text/plain'});

res.end('Hello World\n');

}).listen(1320,'127.0.0.3');

### ****[22) List some features of Express JS.](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled22)****

Some of the main features of Express JS are listed below: –

* It is used for setting up middlewares so as to provide a response to the HTTP or RESTful requests.
* With the help of express JS, the routing table can be defined for performing various HTTP operations.
* It is also used for dynamically rendering [HTML](https://www.onlineinterviewquestions.com/html-interview-questions/)pages which are based on passing arguments to the templates.
* It provides each and every feature which is provided by core Node JS.
* The performance of Express JS is adequate due to the presence of a thin layer prepared by the Express JS.
* It is used for organizing the web applications into the MVC architecture.
* Everything from routes to rendering view and performing HTTP requests can be managed by Express JS.

### ****[23) Write the steps for setting up an Express JS application.](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled23)****

Following are the steps used to set up an express JS application: –

1. A folder with the same name as the project name is created.
2. A file named package.json is created inside the folder created.
3. “npm install” command is run on the command prompt. It installs all the libraries present in package.json.
4. A file named server.js is created.
5. “Router” file is created inside the package which consists of a folder named index.js.
6. “App” is created inside the package which has the index.html file.

This way, an express JS application is set up.

### ****[24) What do you mean by Express JS?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled24)****

Express JS is an application framework which is light-weighted node JS. A number of flexible, useful and important features are provided by this JavaScript framework for the development of mobile as well as web applications with the help of node JS.

**[26) What is the use of Express JS?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled26)**

Express.js is a lightweight web application which helps in organizing the web application into MVC architecture on the server side.

### ****[27) What function are arguments available to Express JS route handlers?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled27)****

The arguments which are available to an Express JS route handler-function are-

* Req – the request object
* Res – the response object
* Next (optional) – a function which is used to pass control to one of the subsequent route handlers.

The third argument is optional and may be omitted, but in some cases, it is useful where there is a chain of handlers and control can be passed to one of the subsequent route handlers skipping the current one.

### ****[28) How to config properties in Express JS?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled28)****

In Express JS, there are two ways for configuring the properties:

* With process.ENV:
* A file with the name “.env” is to be created inside the project folder.
* All the properties are to be added in the “.env” file.
* Any of the properties can be used in server.js.
* With require JS:
* A file with the name “config.json” is to be created in the config folder inside the project folder.
* The config properties are to be added in the config.json file.
* Now, require should be used to access the config.json file.
* **[30) How to authenticate users in express JS?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled30)**
* Since authentication is an opinionated area which is not ventured by express JS, therefore any authentication scheme can be used in express JS for the authentication of users.

**[31) Which template engine is supported by express JS?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled31)**

Express JS supports any template engine that conforms to the (path, locals, callback) signature.

### ****[33) Why to use Express.js?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled33)****

Below are the few reasons why to use Express with Node.js

* Express js is built on top of Node.js. It is the perfect framework for ultra-fast Input / Output.
* Cross Platform
* Support MVC Design pattern
* Support of NoSQL databases out of the box.
* Multiple templating engine support i.e. Jade or EJS which reduces the amount of HTML code you have to write for a page.
* Support Middleware, basic web-server creation, and easy routing tools.

### ****[35) List types of Http requests?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled35)****

Http defines a set of request methods to perform the desired actions. These request methods are:

1. **GET**: The GET method asked for the representation of the specifies resource. This request used to retrieve the data.
2. **POST**: The POST technique is utilized to present an element to the predetermined resource, generally causing a change in state or reactions on the server.
3. **HEAD**: The HEAD method is similar to the GET method but asks for the response without the response body.
4. **PUT**: This method is used to substitute all current representations with the payload.
5. **DELETE**: It is used to delete the predetermined resource.
6. **CONNECT**: This request is used to settle the TCP/IP tunnel to the server by the target resource
7. **OPTION**: This method is used to give back the HTTP strategies to communicate with the target resource.
8. **TRACE**: This method echoes the message which tells the customer how much progressions have been made by an intermediate server.
9. **PATCH**: The PATCH method gives partial modifications to a resource.

### ****[36) What is difference between put and patch?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled36)****

The main difference between the put and the patch is

|  |  |
| --- | --- |
| Put | Patch |
| The embedded entity is believed to the modified version of the resources that are deposited on the original server. It is requested to the client to replace the stored is substituted. | In this, the information regarding the way of modifying the original server which has the resources to produce a new version is found. |
| At the time of updating the resource, you need to forward full payload as the request. | At the time of updating the resource, you only need to send the parameter of the resource which you want to update. |

**[39) Write a simple code to enable CORS in Node js?](https://www.onlineinterviewquestions.com/node-js-interview-questions/" \l "collapseUnfiled39)**

**CORS** stands for Cross-Origin Resource Sharing. It a is a mechanism that uses additional HTTP headers to tell a browser to let a web application running at one origin (domain) have permission to access selected resources from a server at a different origin.

**Use below code to enable CORS on NodeJS**

app.use(function(req, res, next) {

res.header("Access-Control-Allow-Origin", "\*");

res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");

next();

});

**24)   Mention the framework most commonly used in node.js?**

“Express” is the most common framework used in node.js

**15)   What are the two arguments that async.queue takes?**

The two arguments that async.queue takes

a)      Task function

b)      Concurrency value

**16)   What is an event loop in Node.js ?**

To process and handle external events and to convert them into callback invocations an event loop is used. So, at I/O calls, node.js can switch from one request to another .

**17)   Mention the steps by which you can async in Node.js?**

By following steps you can async Node.js

a)      First class functions

b)      Function composition

c)       Callback Counters

d)      Event loops

**18)    What are the pros and cons of Node.js?**

**Pros:**

a)      If your application does not have any CPU intensive computation, you can build it in Javascript top to bottom, even down to the database level if you use JSON storage object DB like MongoDB.

b)      Crawlers receive a full-rendered HTML response, which is far more SEO friendly rather than a single page application or a websockets app run on top of Node.js.

**Cons:**

a)       Any intensive CPU computation will block node.js responsiveness, so a threaded platform is a better approach.  
b)      Using relational database with Node.js is considered less favourable

**19)   How Node.js overcomes the problem of blocking of I/O operations?**

Node.js solves this problem by putting the event based model at its core, using an event loop instead of threads.

**20)   What is the difference between Node.js vs Ajax?**

The difference between Node.js and Ajax is that, Ajax (short for Asynchronous Javascript and XML) is a client side technology, often used for updating the contents of the page without refreshing it. While,Node.js is Server Side Javascript, used for developing server software. Node.js does not execute in the browser but by the server.

**21)   What are the Challenges with Node.js ?**

Emphasizing on the technical side, it’s a bit of challenge in Node.js to have one process with one thread to scale up on multi core server.

**22)** **What does it mean “non-blocking” in node.js?**

In node.js “non-blocking” means that its IO is non-blocking.  Node uses “libuv” to handle its IO in a platform-agnostic way. On windows, it uses completion ports for unix it uses epoll or kqueue etc. So, it makes a non-blocking request and upon a request, it queues it within the event loop which call the JavaScript ‘callback’ on the main JavaScript thread.

**23)   What is the command that is used in node.js to import external libraries?**

Command “require” is used for importing external libraries, for example, “var http=require (“http”)”.  This will load the http library and the single exported object through the http variable.

**10)   Why Node.js is single threaded**?

For async processing, Node.js was created explicitly as an experiment. It is believed that more performance and scalability can be achieved by doing async processing on a single thread under typical web loads than the typical thread based implementation.

**11)   Does node run on windows?**

Yes – it does. Download the MSI installer from <http://nodejs.org/download/>

**12)   Can you access DOM in node?**

No, you cannot access DOM in node.

**13)   Using the event loop what are the tasks that should be done asynchronously?**

a)      I/O operations

b)      Heavy computation

c)       Anything requiring blocking

**What does event-driven programming mean?**

In computer programming, event driven programming is a programming paradigm in which the flow of the program is determined by events like messages from other programs or threads. It is an application architecture technique divided into two sections 1) Event Selection 2) Event Handling