**What is Node.js?**

Node.js is a javascript runtime environment built on Google Chrome's V8 Engine.

Run-time environment includes everything you need to execute a program written in JavaScript. With the help of Node.js we can use javaScript as a server side scripting.

Node.js also provides a rich library of various javascript modules which help us in web development upto a great extents.

Node.js = Runtime Environment + JavaScript Library

**What do you mean by Asynchronous API?**

All APIs of Node.js library are aynchronous that is non-blocking.

It 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.

**What are the benefits of using Node.js?**

Following are main benefits of using Node.js

* **Uses Asynchronous and Event Driven APIs –** 
  + All APIs of Node.js library are asynchronous that is non-blocking.
  + It 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.
* **Very Fast** –
  + As it is built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
* **Single Threaded but highly Scalable** –
  + Node.js uses a single threaded model with event looping(event queue).
  + Event mechanism helps server to respond in a non-bloking ways and makes server highly scalable.
* **No Buffering** –
  + Node.js applications never buffer any data. These applications simply output the data in chunks.

**What is NPM?**

* npm stands for Node Package Manager.
* npm is a package manager for javascript programming language. It comes with Node.js be default.
* It is biggest package manager in the software world.
* It provide us a variety of modules such as express, bodyparser, bcrypt modules which we can include in our project to make our job easier.
* Command line utility to install packages, do version management and dependency management of Node.js packages

**What is Package.json?**

package.json is present in the root directory of any Node application/module and is used to define the properties of a package.

Name some of the attributes of package.json?

Following are the attributes of Package.json

* **name** − name of the package
* **version** − version of the package
* **description** − description of the package
* **homepage** − homepage of the package
* **author** − author of the package
* **contributors** − name of the contributors to the package
* **dependencies** − list of dependencies. npm automatically installs all the dependencies mentioned here in the node\_module folder of the package.
* **repository** − repository type and url of the package
* **main** − entry point of the package
* **keywords** – keywords

**What is Callback?**

* Callback is an asynchronous equivalent for a function.
* A callback function is called at the completion of a given task.
* Node makes heavy use of callbacks.
* All APIs of Node are written is such a way that they supports callbacks. For example, a function to read a file may start reading file and return the control to execution environment immidiately so that next instruction can be executed. Once file I/O is complete, it will call the callback function while passing the callback function, the content of the file as parameter. So there is no blocking or wait for File I/O. This makes Node.js highly scalable, as it can process high number of request without waiting for any function to return result.

What is a blocking code?

If application has to wait for some I/O operation in order to complete its execution any further then the code responsible for waiting is known as blocking code.

How Node prevents blocking code?

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

What is Event Loop?

Node js is a single threaded application but it support concurrency via concept of event and callbacks. As every API of Node js are asynchronous and being a single thread, it uses async function calls to maintain the concurrency. Node uses observer pattern. Node thread keeps an event loop and whenever any task get completed, it fires the corresponding event which signals the event listener function to get executed.

What is Piping in Node?

Piping is a mechanism to connect output of one stream to another stream. It is normally used to get data from one stream and to pass output of that stream to another stream. There is no limit on piping operations. Consider the above example, where we've read test.txt using readerStream and write test1.txt using writerStream. Now we'll use the piping to simplify our operation or reading from one file and writing to another file.

What is difference between synchronous and asynchronous method of fs module?

Every method in fs module have synchronous as well as asynchronous form. Asynchronous methods takes a last parameter as completion function callback and first parameter of the callback function is error. It is preferred to use asynchronous method instead of synchronous method as former never block the program execution where the latter one does.

What are streams?

Streams are objects that let you read data from a source or write data to a destination in continuous fashion.

What is Chaining in Node?

Chanining is a mechanism to connect output of one stream to another stream and create a chain of multiple stream operations. It is normally used with piping operations.

**What is Next ?**

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

Application of NodeJS:

NodeJS should be preferred to build:

* Real Time Chats,
* Complex Single-Page applications,
* Real-time collaboration tools,
* Streaming apps
* JSON APIs based application

### **Where Node.js can be used?**

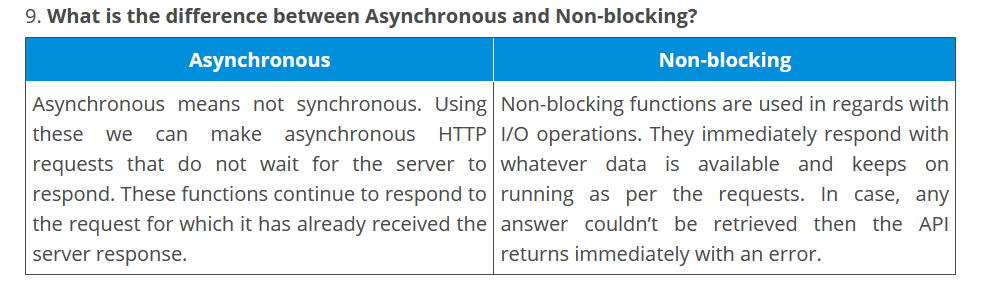
Node.js can be used to develop:

* Real-Time Web Applications
* Network Applications
* Distributed Systems
* General Purpose Applications

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

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

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

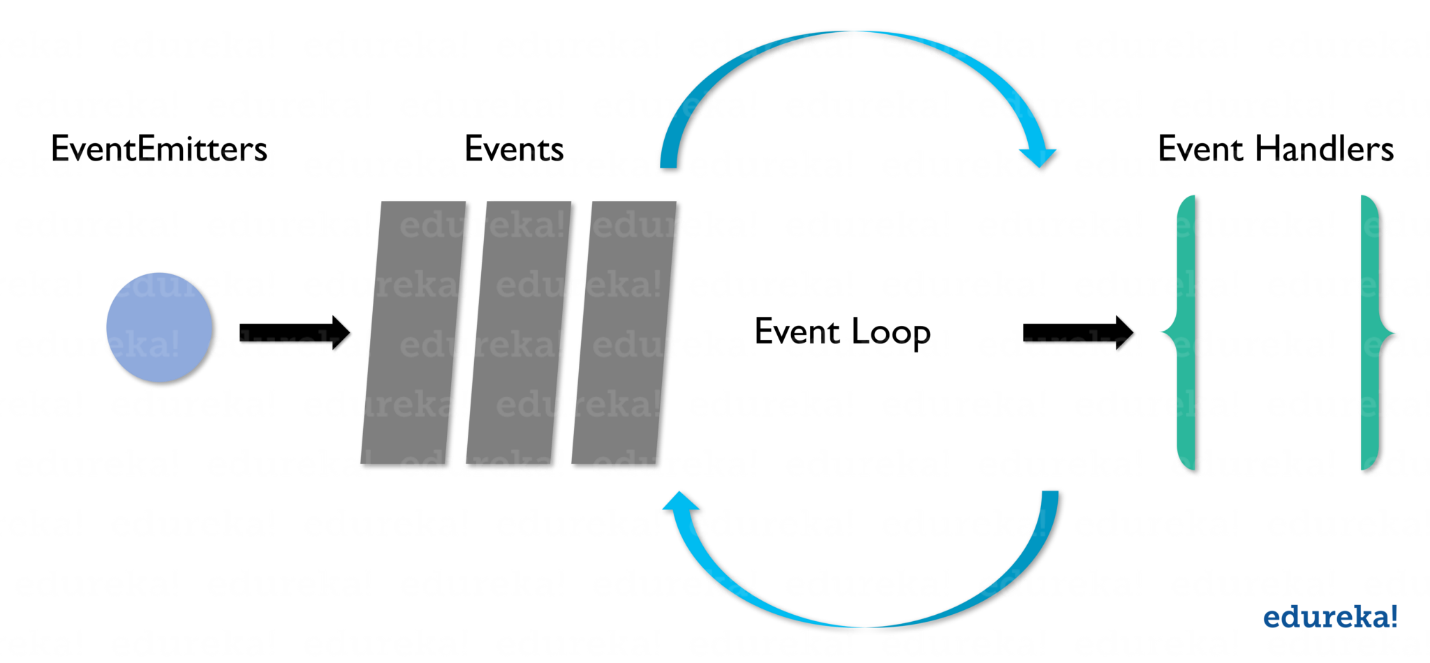
* Asynchronous, non-blocking functions
* Synchronous, blocking functions

### **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.

### **What is an *Event loop* in Node.js and how does it work?**

An event loop in Node.js handles all the asynchronous callbacks in an application. It is one of the most important aspects of Node.js and the reason behind Node.js have non-blocking I/O. Since Node.js is an event-driven language, you can easily attach a listener to an event and then when the event occurs the callback will be executed by the specific listener. Whenever functions like setTimeout, http.get, and fs.readFile are called, Node.js executed the event loop and then proceeds with the further code without waiting for the output. Once the entire operation is finished, Node.js receives the output and then executes the callback function. This is why all the callback functions are placed in a queue in a loop. Once the response is received, they are executed one by one.

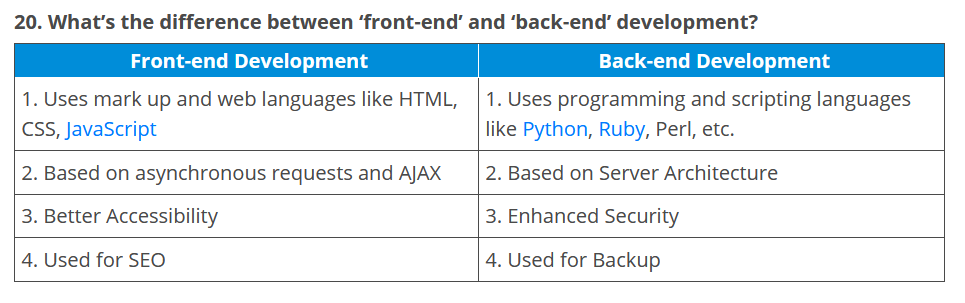


### **List down the tasks which should be done asynchronously using the event loop?**

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

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

### **Explain the purpose of module.exports?**

A module in Node.js is used to encapsulate all the related codes into a single unit of code which can be interpreted by shifting all related functions into a single file.

### **List down the major security implementations within Node.js?**

Major security implementations in Node.js are:

1. Authentications
2. Error Handling

### **Explain the concept of middleware in Node.js?**

In general, middleware is a function receives the Request and Response objects. In other words, in an application’s request-response cycle these functions have access to various request &  response objects along with the next function of the cycle. The next function of middleware is represented with the help of a variable, usually named next. Most commonly performed tasks by the middleware functions are:

* Execute any type of code
* Update or modify the request and the response objects
* Finish the request-response cycle
* Invoke the next middleware in the stack

### **Is cryptography supported in Node.js?**

Yes, Node.js does support cryptography through a module called Crypto. This module provides various cryptographic functionalities like cipher, decipher, sign and verify functions, a set of wrappers for open SSL’s hash HMAC etc. For example:

### What are modules in Node.js?

Modules are like JavaScript libraries that can be used in a Node.js application to include a set of functions. To include a module in a Node.js application, use the **require()** function with the parentheses containing the name of the module.

### Why is Node.js preferred over other backend technologies like Java and PHP?

Some of the reasons why Node.js is preferred include:

* Node.js is very fast
* Node Package Manager has over 50,000 bundles available at the developer’s disposal
* Perfect for data-intensive, real-time web applications, as Node.js never waits for an API to return data
* Better synchronization of code between server and client due to same code base
* Easy for web developers to start using Node.js in their projects as it is a JavaScript library