1. What is Node.js?

Node.js is Server-side scripting which is used to build scalable programs. It is a web application framework built on Google Chrome's JavaScript Engine. It runs within the Node.js runtime on Mac OS, Windows, and Linux with no changes. This runtime facilitates you to execute a JavaScript code on any machine outside a browser.

1. Explain Node.js web application architecture?

A web application distinguishes into 4 layers:

* **Client Layer:** The Client layer contains web browsers, mobile browsers or applications which can make an HTTP request to the web server.
* **Server Layer:** The Server layer contains the Web server which can intercept the request made by clients and pass them the response.
* **Business Layer:** The business layer contains application server which is utilized by the web server to do required processing. This layer interacts with the data layer via database or some external programs.
* **Data Layer:** The Data layer contains databases or any source of data.

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

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

* Asynchronous, Non-blocking functions
* Synchronous, Blocking functions

### What do you understand by the first class function in JavaScript?

When functions are treated like any other variable, then those functions are called first-class functions. Apart from JavaScript, many other programming languages, such as Scala, Haskell, etc. follow this pattern. The first class functions can be passed as a param to another function (callback), or a function can return another function (higher-order function). Some examples of higher-order functions that are popularly used are map() and filter().

What is the difference between JavaScript and Node.js?

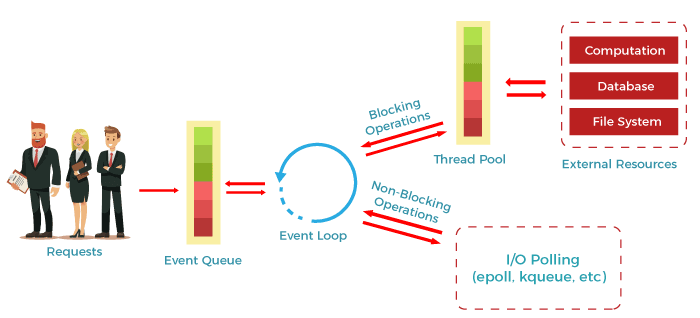
**Difference between JavaScript and Node.js**

The following table specifies the crucial differences between JavaScript and Node.js:

|  |  |  |
| --- | --- | --- |
| **Comparison features** | **JavaScript** | **Node.js** |
| Type | JavaScript is a programming language. More precisely, you can say that it is a scripting language used for writing scripts on the website. | Node.js is an interpreter and run time environment for JavaScript. |
| Utility | JavaScript is used for any client-side activity for a web application. | Node.js is used for accessing or performing any non-blocking operation of any operating system. |
| Running Engine | The running engine for JavaScript is Spider monkey (Firefox), JavaScript Core (Safari), V8 (Google Chrome), etc. | The running engine for Node.js is V8 (Google Chrome). |
| Browser compatibility | JavaScript can only be run in browsers. | The Node.js code can be run outside the browser. |
| Platform dependency | JavaScript is basically used on the client-side and is used in frontend development. | Node.js is mostly used on the server-side and is used in server-side development. |
| HTML compatibility | JavaScript is capable enough to add HTML and play with the DOM. | Node.js is not compatible enough to add HTML tags. |
| Examples | Some examples of the JavaScript frameworks are RamdaJS, TypedJS, etc. | Some examples of the Node.js modules are Lodash, express, etc. We have to import these modules from npm. |
| Written in | JavaScript is the upgraded version of ECMA script that uses Chrome's V8 engine and is written in C++. | Node.js is written in C, C++, and Javascript. |

### Explain the working of Node.js?

The workflow of a Node.js web server typically looks like the following diagram. Let us see the flow of operations in detail:



* According to the above diagram, the clients send requests to the webserver to interact with the web application. These requests can be non-blocking or blocking and used for querying the data, deleting data, or updating the data.
* js receives the incoming requests and adds those to the Event Queue.
* After this step, the requests are passed one by one through the Event Loop. It checks if the requests are simple enough not to require any external resources.
* The event loop then processes the simple requests (non-blocking operations), such as I/O Polling, and returns the responses to the corresponding clients.
* A single thread from the Thread Pool is assigned to a single complex request. This thread is responsible for completing a particular blocking request by accessing external resources, such as computation, database, file system, etc.
* Once the task is completed, the response is sent to the Event Loop that sends that response back to the client.

### What do you understand by the term fork in Node.js?

Generally, a fork is used to spawn child processes. In Node.js, it is used to create a new instance of the V8 engine to run multiple workers to execute the code.

### What are the modules in Node.js? Which are the different modules used in Node.js?

In Node.js applications, modules are like JavaScript libraries and include a set of functions. To include a module in a Node.js application, we must use the require() function with the parentheses containing the module's name.

Node.js has several modules which are used to provide the basic functionality needed for a web application. Following is a list of some of them:

|  |  |
| --- | --- |
| **Core Modules** | **Description** |
| HTTP: | The HTTP module includes classes, methods, and events to create a Node.js HTTP server. |
| util: | The util module includes utility functions required in the application and is very useful for developers. |
| url: | The url module is used to include the methods for URL parsing. |
| fs: | The fs module includes events, classes, and methods to handle the file I/O operations. |
| stream: | The stream module is used to include the methods to handle streaming data. |
| query string: | The query string module is used to include the methods to work with a query string. |
| zlib: | The zlib module is used to include the methods to compress or decompress the files used in an application |

### What is an asynchronous API?

All the API's of Node.js library are asynchronous means non-blocking. A Node.js based server never waits for an API to return data. The Node.js server moves to the next API after calling it, and a notification mechanism of Events of Node.js responds to the server for the previous API call.

### What is REPL in Node.js?

REPL stands for Read Eval Print Loop. It specifies a computer environment like a window console or Unix/Linux shell where you can enter a command, and the computer responds with an output. It is very useful in writing and debugging the codes. REPL environment incorporates Node.js.

### What is npm? What is the main functionality of npm?

npm stands for Node Package Manager. Following are the two main functionalities of npm:

* Online repositories for node.js packages/modules which are searchable on search.nodejs.org
* Command line utility to install packages, do version management and dependency management of Node.js packages.