### ****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 (Fire Fox), JavaScript Core (Safari), V8 (Google Chrome), etc. | V8 (Google Chrome) |

### ****2. What is Node.js ?****

Node.js is a powerful framework developed on **Chrome’s V8 JavaScript engine** that compiles the JavaScript directly into the native machine code. It is a lightweight framework used for creating server-side web applications and extends JavaScript API to offer usual server-side functionalities.

It is generally used for large-scale application development, especially for video streaming sites, single page application, and other web applications.

### **3. 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 |
| ***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 |

### ****4. What is the difference between Angular and Node.js ?****

|  |  |
| --- | --- |
| **Angular** | **Node.js** |
| 1. It is an **open source** web application development framework | 1. It is a cross-platform **run-time environment** for applications |
| 2. It is written in TypeScript | 2. It is written in **C, C++ and JavaScript** languages |
| 3. Used for building **single-page** client-side web applications | 3. Used for building f**ast and scalable**server-side networking applications |
| 4. Angular itself is a **web application framework** | 4. Node.js has many **different frameworks**like Sails.js, Partial.js, and Express.js, etc. |

### ****5. Why Node.js is single threaded ?****

### (Yes! Node uses a single threaded model with event looping)

Node.js uses a single threaded model in order to support async processing. With async processing, an application can perform better and is more scalable under web loads.

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

Node.js can be used to develop :

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

### ****7. 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

|  |  |
| --- | --- |
|  |  |

### ****8. What is package.json ?****

This will be present in the root directory of any Node module and will be used to define the properties of a package. and help to install packages from npm

{

"name": "NodeJs-Mysql",

"version": "0.0.1",

"description": "NodeJs with mysql connectivity",

"dependencies": {

"express": "^4.13.1",

"mysql": "^2.8.0"

}

}

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

### An event-driven programming approach uses events to trigger various functions. An event can be anything, such as typing a key or clicking a mouse button. A call-back function is already registered with the element executes whenever an event is triggered

### ****10. Explain REPL in the context of Node.js ?****

REPL in Node.js stands for **R**ead, **E**val, **P**rint, and **L**oop. 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.

### ****11. 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

### ****12. 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

**16. Who developed Node.js & When ?**

Node.js was developed in 2009 by Ryan Dahl.

**17. What do you mean by the term I/O ?**

I/O is shorthand for input and output, and it will access anything outside of your application. It will be loaded into the machine memory to run the program, once the application is started.

**18. How Node.js work ?**

Node.js works on a v8 environment, it is a virtual machine that utilizes JavaScript as its scripting language and achieves high output via non-blocking I/O and single threaded event loop

**19. What is Advantage of Node.js ?**

It provides an easy way to build scalable network programs.

\* Faster in processing

\* Generally fast

\* Event driven and Asynchronous

\* Single Threaded but highly Scalable.

**20. What is control flow function ?**

A generic piece of code which runs in between several asynchronous function calls is known as control flow function

**21. What is the command to stop REPL in Node.js ?**

Use ctrl + c command twice to stop REPL

**22.What is npm ?**

npm stands for Node Package Manager. which help to install node.js packages/plug-ins Command line, also does version management and dependency management of Node.js packages

**23.How to get npm version in node.js.?**

npm –version

**25.What is the command to check the already installed dependencies which are globally installed using npm ?**

npm ls -g

**26. What is callback in Node.js ?**

Callback is called once the asynchronous operation has been completed. Node.js heavily uses callbacks and all API’s of Node.js are written to support callbacks.

**27.Can you access DOM in node ?**

No, you cannot access DOM in node

**28. Explain FS module ?**

FS stands for “File System” and fs module is used for File I/O. You can include FS module by following way

var fsLib = require("fs");

**29. Explain Console in Node.JS ?**

“Console” is a global object and will be used for printing to stderr and stdout and this will be used in synchronous manner in case of destination is either file or terminal or else it is used in asynchronous manner when it is a pipe.

**30. Explaine OS module in Nodejs ?**

OS module is used for some basic operating system related utility functions. Below is the syntax for including OS module in nodejs

var osLib = require(“os”);

|  |
| --- |
|  |

**31. Explain Path module in Node.JS ?**

Path module will be used for transforming and handling file paths. Below is the syntax of path module

var pathLib = require(“path”);

**32. Explain the difference between frontend and backend development ?**

|  |  |
| --- | --- |
| **Front-end** | **Back-end** |
| Frontend refers to the client-side of an application | Backend refers to the server-side of an application |
| It is the part of a web application that users can see and interact with | It constitutes everything that happens behind the scenes |
| It typically includes everything that attributes to the visual aspects of a web application | It generally includes a web server that communicates with a database to serve requests |
| HTML, CSS, JavaScript, AngularJS, and ReactJS are some of the essentials of frontend development | Java, PHP, Python, and Node.js are some of the backend development technologies |

### 33. What are some of the most commonly used libraries in Node.js ?

There are two commonly used libraries in Node.js:

* **ExpressJS** - Express is a flexible Node.js web application framework that provides a wide set of features to develop web and mobile applications.
* **Mongoose** - Mongoose is also a Node.js web application framework that makes it easy to connect an application to a database.

### 34. What are the pros and cons of Node.js ?

|  |  |
| --- | --- |
| **Node.js Pros** | **Node.js Cons** |
| Fast processing and an event-based model | Not suitable for heavy computational tasks |
| Uses JavaScript, which is well-known amongst developers | Using callback is complex since you end up with several nested callbacks |
| Node Package Manager has over 50,000 packages that provide the functionality to an application | Dealing with relational databases is not a good option for Node.js |

### 35. What is the command used to import external libraries ?

varhttp

### 36. What is the Express.js package ?

Express is a flexible Node.js web application framework that provides a wide set of features to develop both web and mobile applications.

### 37. What are streams in Node.js ?

There are four types of streams:

**Readable –** Used for reading operations

**Writable −** Used for write operations

**Duplex −** Can be used for both reading and write operations

**Transform −** A type of duplex stream where the output is computed based on input

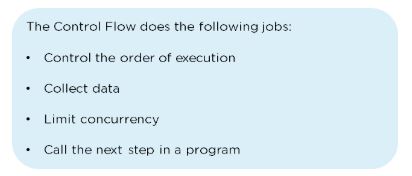
### 38. How do you install, update, and delete a dependency ?



### 39. Explain asynchronous and non-blocking APIs in Node.js ?

* All Node.js library APIs are asynchronous, which means they are also non-blocking
* A Node.js-based server never waits for an API to return data. Instead, it moves to the next API after calling it, and a notification mechanism from a Node.js event responds to the server for the previous API call

### 40. How does control flow manage the function calls ?



### 41. What is the buffer class in Node.js ?

Buffer class stores raw data similar to an array of integers but corresponds to a raw memory allocation outside the V8 heap. Buffer class is used because pure JavaScript is not compatible with binary data.

### 42. What is callback hell ?

* Callback hell, also known as the pyramid of doom, is the result of intensively nested, unreadable, and unmanageable callbacks, which in turn makes the code harder to read and debug
* improper implementation of the asynchronous logic causes callback hell

### 43. What are the different types of HTTP requests ?

HTTP defines a set of request methods used to perform desired actions. The request methods include:

**GET:**Used to retrieve the data

**POST:**Generally used to make a change in state or reactions on the server

**HEAD:**Similar to the GET method, but asks for the response without the response body

**DELETE:** Used to delete the predetermined resource

### 44. How would you connect a MongoDB database to Node.js ?

To create a database in MongoDB:

* Start by creating a MongoClient object
* Specify a connection URL with the correct IP address and the name of the database you want to create

