Q1. What is Node.js? Where can you use it?

Ans: Node.js:

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node. js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Application:

It is used for server-side programming, and primarily deployed for non-blocking, event-driven servers, such as traditional web sites and back-end API services

Q2. Explain callback in Node.js.

Ans: callback:

 A callback is a function which is called when a task is completed, thus helps in preventing any kind of blocking and a callback function allows other code to run in the meantime. Callback is called when task get completed and is asynchronous equivalent for a function.

Q3. What are the advantages of using promises instead of callbacks?

Ans: They can handle multiple asynchronous operations easily and provide better error handling than callbacks and events. In other words, also, we may say that, promises are the ideal choice for handling multiple callbacks at the same time, thus avoiding the undesired callback hell situation.

Q4. What is NPM?

Ans: npm is the package manager for Node. js. It was created in 2009 as an open-source project to help JavaScript developers easily share packaged modules of code. The npm Registry is a public collection of packages of open-source code for Node.

Q5. What are the modules in Node.js? Explain

Ans: In Node.js, Modules are the blocks of encapsulated code that communicates with an external application on the basis of their related functionality. Modules can be a single file or a collection of multiples files/folders. The reason programmers are heavily reliant on modules is because of their re-usability as well as the ability to break down a complex piece of code into manageable chunks.

Modules are of three types:

* Core Modules
* local Modules
* Third-party Modules

Core Modules: Node.js has many built-in modules that are part of the platform and comes with Node.js installation. These modules can be loaded into the program by using the require function

Core Modules Example:

|  |  |
| --- | --- |
| Core Modules | Description |
| fs | Used for handling the file system |
| path | Include the methods to deal with the file paths. |
| os | Provide the information about the operating system |
| http | Creates an HTTP serve in node.js |

Local Modules: Unlike built-in and external modules, local modules are created locally in your Node.js application. Let’s create a simple calculating module that calculates various operation.

Third-party modules: Third-party modules are modules that are available online using the Node Package Manager (NPM). These modules can be installed in the project folder or globally. Some of the popular third-party modules are mongoose, express, angular, and react.

Example:

* npm install express
* npm install mongoose