

Practical: 1

Q.1 Differentiate JavaScript and Node.js.

JavaScript	Node.js
<ul style="list-style-type: none">• JavaScript is a programming language that is used for writing scripts on the website.	<ul style="list-style-type: none">• NodeJS is a JavaScript runtime environment.
<ul style="list-style-type: none">• JavaScript can only be run in the browsers.	<ul style="list-style-type: none">• We can run JavaScript outside the browser with the help of NodeJS.
<ul style="list-style-type: none">• It is basically used on the client-side.	<ul style="list-style-type: none">• It is mostly used on the server-side.
<ul style="list-style-type: none">• JavaScript is capable enough to add HTML	<ul style="list-style-type: none">• NodeJS does not have capability to add HTML tags.
<ul style="list-style-type: none">• JavaScript is used in frontend development.	<ul style="list-style-type: none">• NodeJS is used in server-side development.

Q.2 What is the difference between ‘front-end’ and ‘back-end’ development Web Application?

front-end	back-end
<ul style="list-style-type: none">• Front-end developers design the visual aspects of websites for users to interact with, including colors, layout, and fonts.	<ul style="list-style-type: none">• Back-end developers create the invisible structure that helps websites function properly.
<ul style="list-style-type: none">• Front-end development focuses on the user-facing side of a website. Front-end developers ensure that visitors can easily interact with and navigate sites by using programming languages, design skills, and other tools. They produce the drop-down menus, layouts, and designs for websites.	<ul style="list-style-type: none">• Back-end developers focus on the server side of websites. They use technical skills to perform the behind-the-scenes work that creates a website's structure and overall functionality, allowing a site's front end to exist. These professionals create a site's operations, databases, and application programming interface (API).

Q.3 What is Node.js?

- Node.js was first introduced in 2009 developed by Ryan Dahl and is a runtime environment for JavaScript built on Google's v8 engine whose main purpose is to run JavaScript on the server and hence JavaScript can be executed outside of the browser. The nicest part about Node.js is that it never blocks I/O, is event-driven, and can be used to create highly scalable apps. In Node.js everything is a module and using these modules developers make use of Node.js in creating web APIs, Rest API servers, command-line applications, and real-time chat applications.

Q.4 Write down advantages of Node.js?

- Fast Execution
- I/O is Asynchronous and Event Driven
- Single threaded
- Highly Scalable
- No buffering
- Open source
- Rich library
- Portable
- Large Community Support

Q.5 How does Node.js differ with other languages?

- **QV8 is a set of libraries built on top of an extremely powerful virtual machine.**
 - Node.js is a bundled version of Google's V8 JavaScript engine, the libuv platform abstraction layer, and a core library written mostly in JavaScript." It's also worth noting that Ryan Dahl, the Node.js founder, was attempting to create real-time websites with push functionality. He provides developers a tool to work in the non-blocking, event-driven I/O paradigm with Node.js.
- **Event-driven asynchronous callbacks.**
 - To understand why Node.js applications have to be written this way, we need to understand how Node.js executes code. Node's approach isn't unique, but the underlying execution model is different from runtime environments like Python, Ruby, PHP, or Java.
- **NPM: The Node Package Manager**
 - The idea of NPM modules is quite similar to that of Ruby Gems: a set of publicly available, reusable components, available through easy installation via an online repository, with version and dependency management.

- A full list of packaged modules can be found on the NPM website <https://npmjs.org/> or accessed using the NPM CLI tool that automatically gets installed with Node.js. The module ecosystem is open to all, and anyone can publish their module that will be listed in the NPM repository.

Q.6 Who uses Node.js?

In fact, Node.js is so popular; quite a few major business enterprises are well-acquainted with the software. Companies that use Node.js include the following:

- LinkedIn
- Netflix
- Uber
- Trello
- PayPal
- NASA
- eBay
- Medium
- Groupon
- Walmart
- Mozilla
- GoDaddy

Q.7 Which are the applications of Node.js?

1. Browser Games

- Browser-based games are probably the most exciting application of real-time web. No longer have users needed to open up Flash, Java, or other programs such as Shockwave applet to play games. They do it right in their browsers. Node.js used in conjunction with Socket.io and HTML 5, rich and impressive real-time browser games can be designed. Other games such as multiplayer games as well can be created using the same technology.

2. Chat Rooms

- These days it is true that most people use phone-based apps or social media platforms to chat with friends. Uses of Node.js can create chat rooms very easily. Independent chat rooms probably not very popular on today's Internet, but this is used a lot in online games which makes more fun when there is a chat component to them.

3. Collecting Data

- Massive amounts of data can be collected and made more efficient with the uses of Node.js. large quantities of data into a database usually create a bottleneck, because database access is a blocked operation. Node.js, on the other hand, receives this data, then send it to the backend in a piecewise manner. This ensures that data gets stored without breaking any system.

4. Streaming

- Video and audio media content files are uploaded to streaming video such as YouTube or Facebook and the same has to be processed to make it available for viewing or heard by users across the world. The processing of these files happens after they have been uploaded. Using applications of Node.js, these videos can be processed as they are being uploaded. This saves a lot of time and makes media widely accessible at fast speed. Other than video and audio, other data types such as input and output in real-time also require real-time processing to work on projects with colleagues spread halfway around the world.

5. Mean Stack

- The MEAN Stack is all coded in JavaScript and hence it is worth looking at. Mean stack makes development simple and easy both for a developer standpoint and for the devices too. MEAN can be integrated with other modules too. Example: – Mongoose is used with

Node.js that made integration with MongoDB much easier. Actually, N in Node.js stands for N in MEAN Stack.

6. Real-time Applications

- Uses of Node.js is a great choice for those kinds of applications that process a high volume of short messages and require low latency. Such systems can be easily developed with Node.js. Uses of Node.js can also be a good choice for real-time collaborative drawing and editing-type apps. Example: – Trello, DropBox.

7. Fast and scalable Applications

- Ruby on Rails probably is not sufficient in speed, if there are a ton of requests. Uses of Node.js prove to be useful in such situations when faster and a more scalable program is needed. Node has the ability to process many requests, that too at low response times, and it shares things such as validation code between client machine and server. This makes applications of Node.js a great fit for modern web applications. Carrying out lots of processing on the client's side is possible with node.js. Node.js is a popular choice for single-page application sites and mobile apps, where rendering is done on the client's side, and a JSON API is provided by a backend.

8. Processing

- Node.js also very useful, when high volumes of IO-bound requests are to be processed. If a lot of CPU processing requests are required to serve the request, it will not be efficient. If it's primarily and considers only shuffling data around, then node.js will be quite efficient. A single instance serves a lot more requests than too with the same hardware. Node.js compared to the usual big and thick applications is better such as rails.

9. API on top of an object DB

- Although Node.js really shines with real-time applications, it's quite a natural fit for exposing the data from object DBs (e.g. MongoDB). JSON stored data allow Node.js to function even when there is no impedance mismatch and there is data conversion

10. Queued Input

- If a huge amount of simultaneous data is received, a database does become a bottleneck as explained above. But, applications of Node.js can handle the database concurrent connections. The system maintains its responsiveness even under a heavy load and that is particularly useful

Q.8 How to download & install Node.js

Step1: Open link and download Node.js application according to your system like Windows, MAC, Linux etc. [Download Here](#)

Downloads
Latest LTS Version: **16.16.0** (includes npm 8.11.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users		Current Latest Features
Windows Installer node-v16.16.0-x64.msi	macOS Installer node-v16.16.0.pkg	Source Code node-v16.16.0.tar.gz

Platform	32-bit	64-bit
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v16.16.0.tar.gz	

Step2): After Downloading was completed just install it in your system.

Run the installation

Double click on the downloaded .msi file to start the installation.

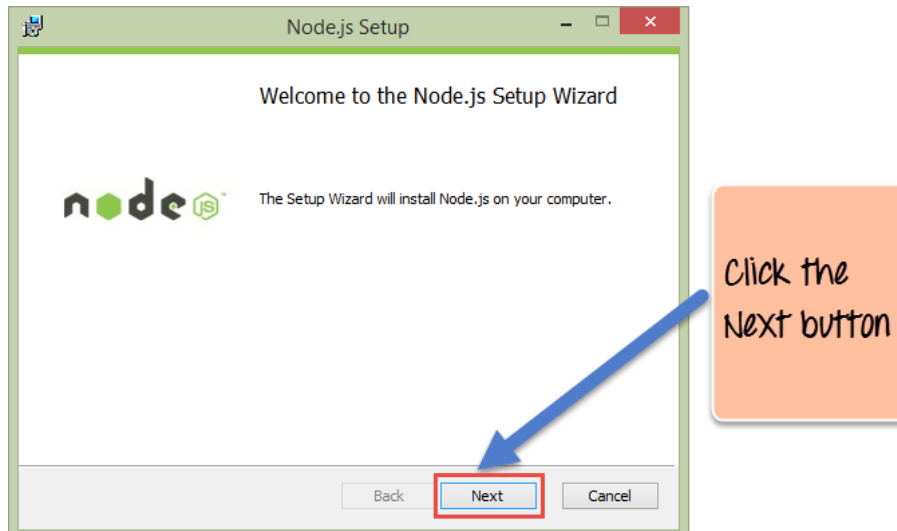
Click the Run button on the first screen to begin the installation.



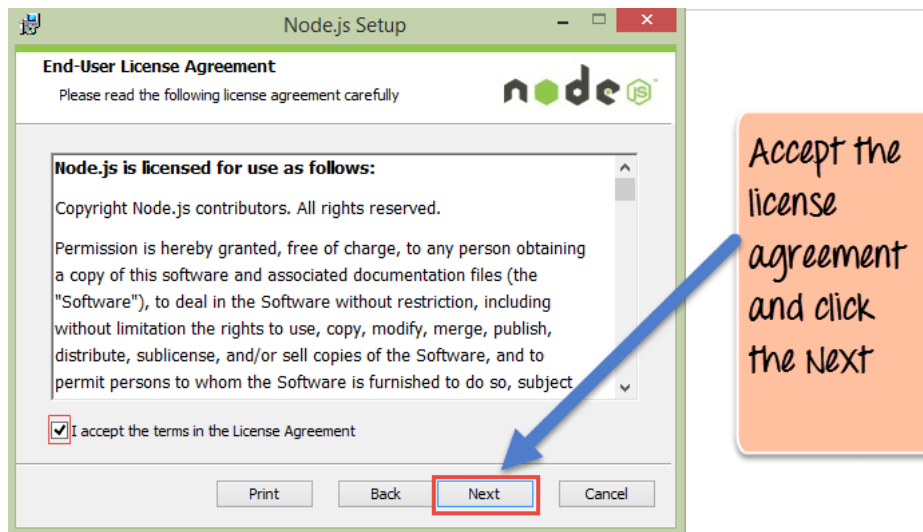
Click the
Run button

Step 3): Continue with the installation steps

In the next screen, click the “Next” button to continue with the installation

**Step 4):** Accept the terms and conditions

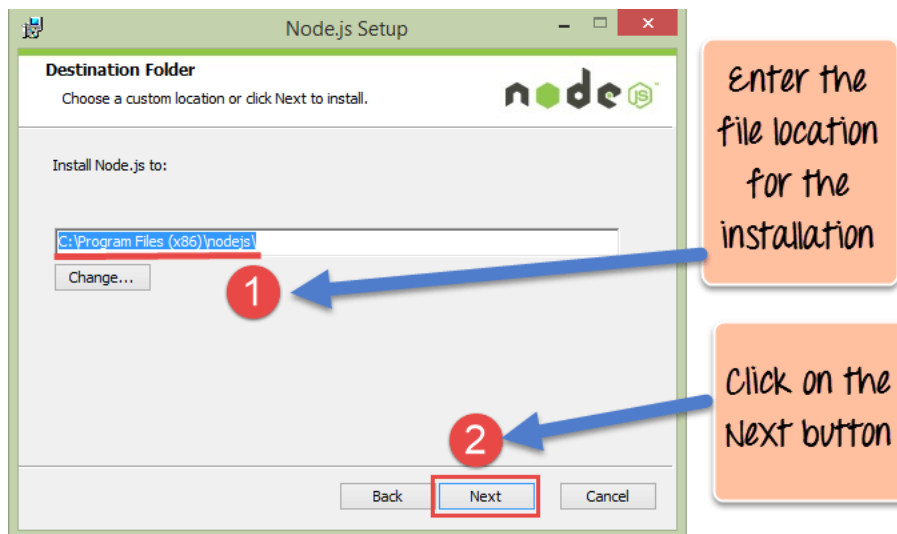
In the next screen, Accept the license agreement and click on the Next button.



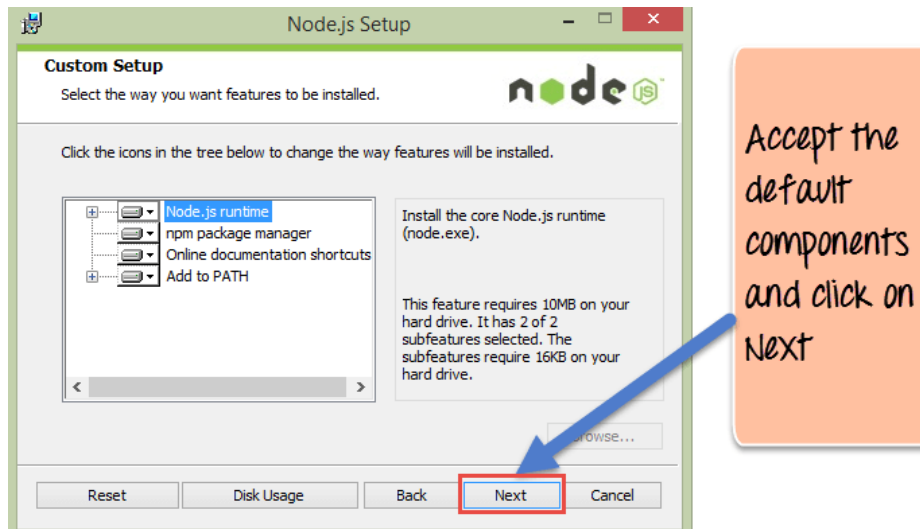
Step 5): Set up the path

In the next screen, choose the location where Node.js needs to be installed and then click on the Next button.

1. First, enter the file location for the installation of Node.js. This is where the files for Node.js will be stored after the installation.
2. Click on the Next button to proceed ahead with the installation.

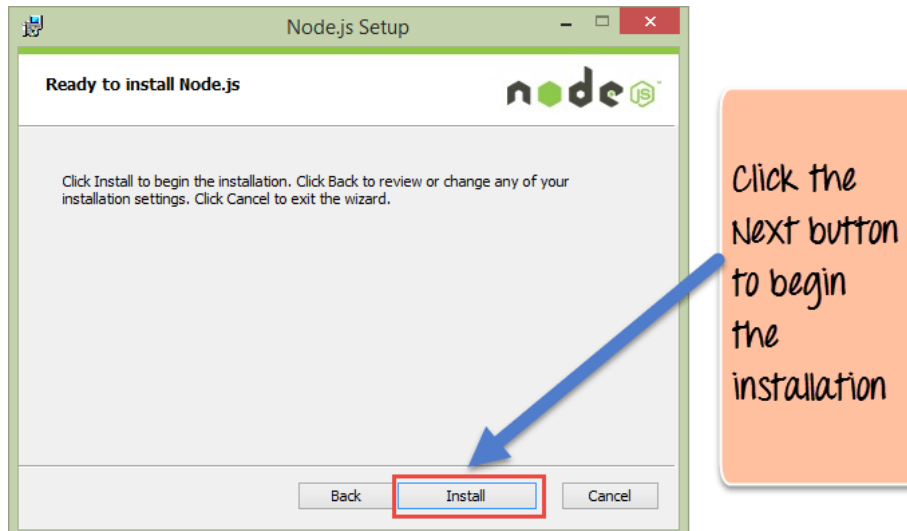
**Step 6): Select the default components to be installed**

Accept the default components and click on the Next button.

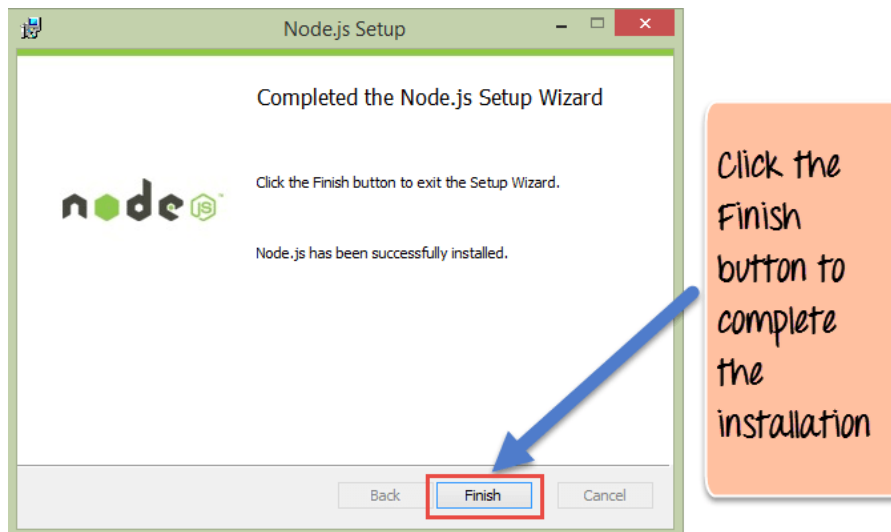


Step 7): Start the installation

In the next screen, click the Install button to start installing Node.js on Windows.

**Step 8): Complete the installation**

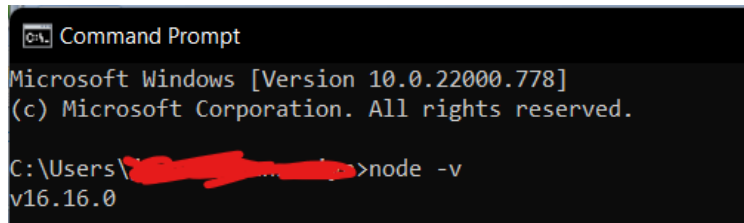
Click the Finish button to complete the installation.



Q.9 How to check installed version of Node.js?

On the command prompt, use the following command to validate the version of Node.js that is installed.

node -v



```
Command Prompt
Microsoft Windows [Version 10.0.22000.778]
(c) Microsoft Corporation. All rights reserved.

C:\Users\>node -v
v16.16.0
```

Q.10 How to write and execute Node.js program to print “Hello World” in console?

- 1: First we create file with extension **.js**.
- 2: Open those file and write **console.log(“Hello World”)** then save.
- 3: Open command prompt and go to folder where file is saved.
- 4: Type command to run Node.js file **‘node [filename].js’** and press Enter.



```
C:\Windows\System32\cmd.exe
E:\B_Tech\SEM_5\SP\Practical\Practical 1>node p1.js
Hello World
E:\B_Tech\SEM_5\SP\Practical\Practical 1>
```