

JAVASCRIPT

24 June 2024

JavaScript is a **programming language used for creating dynamic content on websites**. It is a **lightweight, cross-platform** and **single-threaded** programming language. JavaScript is an **interpreted** language that executes code line by line providing more flexibility. It is a commonly used programming language to **create dynamic and interactive elements in web applications**. It is easy to learn.

Why to learn JavaScript?

JavaScript is an essential programming language for web developers to learn as it grows rapidly. JavaScript is responsible for behaviour of the web pages. By learning it, you will develop skills in web development.

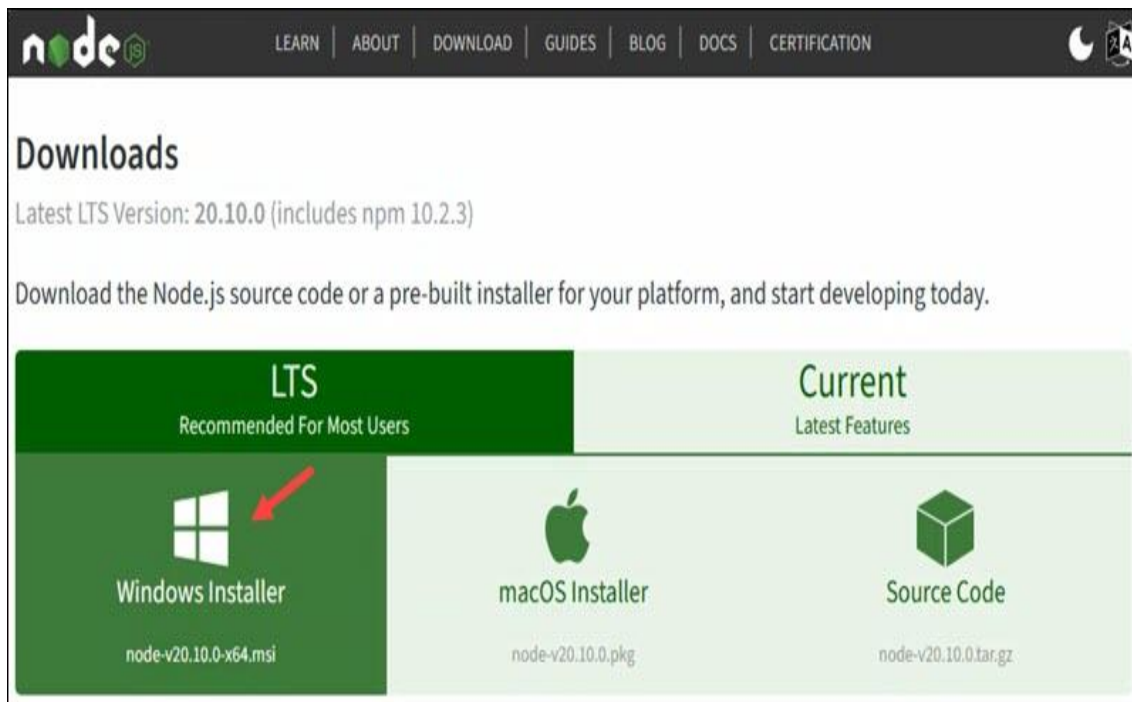
Key Features of JavaScript:

1. **Versatility:** JavaScript can be used to develop websites, games, mobile apps, and more.
2. **Client and Server-Side:** With frameworks like [Node.js](#) and [Express.js](#), JavaScript is now used for building server-side applications.
3. **End-to-End Solutions:** JavaScript enables developers to create complete solutions for various problems.
4. **Constant Evolution:** JavaScript continually evolves with new features and standards.
5. **Vibrant Community:** A large community of users and mentors actively contributes to JavaScript's growth.

Download Node.js

Step1: Download Node.js Installer

In a web browser, navigate to the [Node.js Downloads](#) page. Click the **Windows Installer** button to download the latest stable version with long-term support (LTS). The installer also includes the NPM [package manager](#).



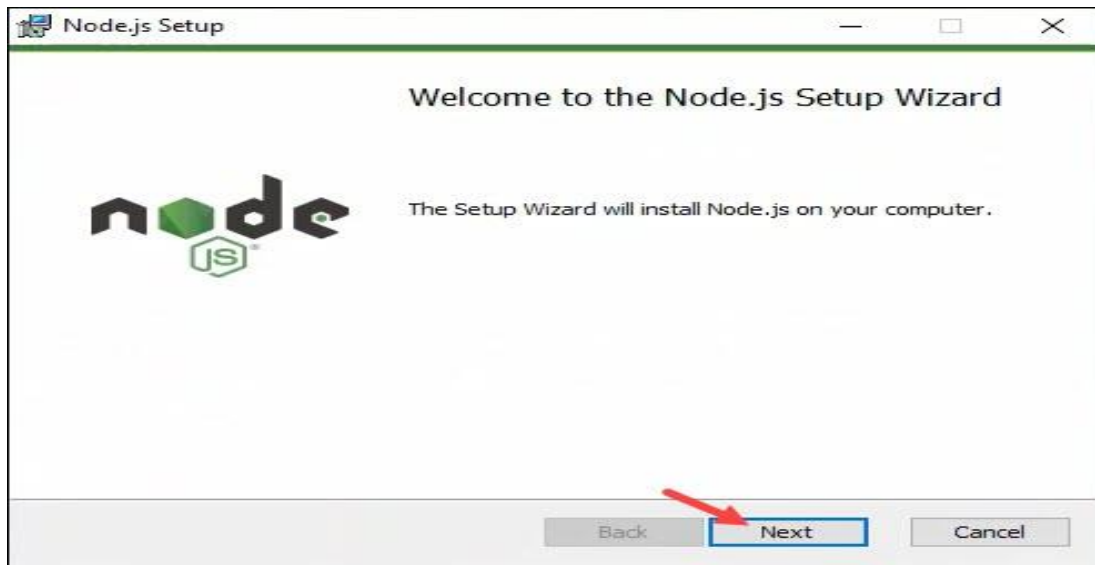
The [file](#) is saved in the **Downloads** folder by default.

Other versions of Node.js and NPM are available, so choose the appropriate one for your system and use case. Use the top tabs to switch from the LTS to the current version to test the newest features. If you are new to Node.js or don't need a specific version, choose LTS since it is tested and stable.

Step 2: Install Node.js and NPM

After downloading the installer, follow the steps below:

1. Launch the installer by double-clicking the downloaded file.
2. The Node.js Setup Wizard starts with the welcome screen.



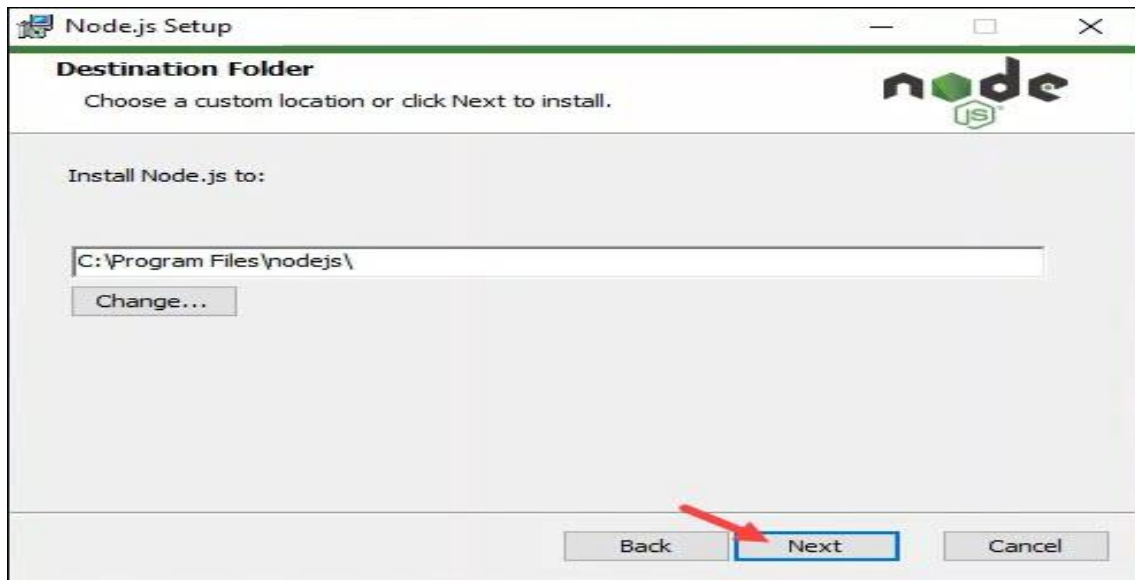
Click **Next** to proceed.

4. Review the end-user license agreement and click the checkbox to accept the terms and conditions.



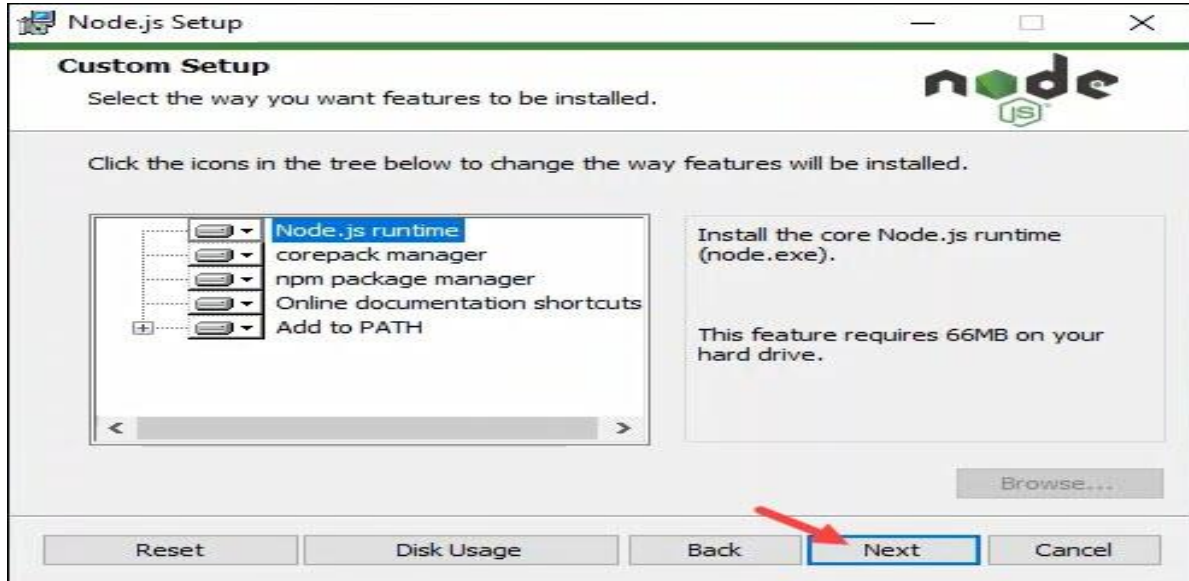
Click **Next** to continue.

5. The installer asks to choose the installation location.



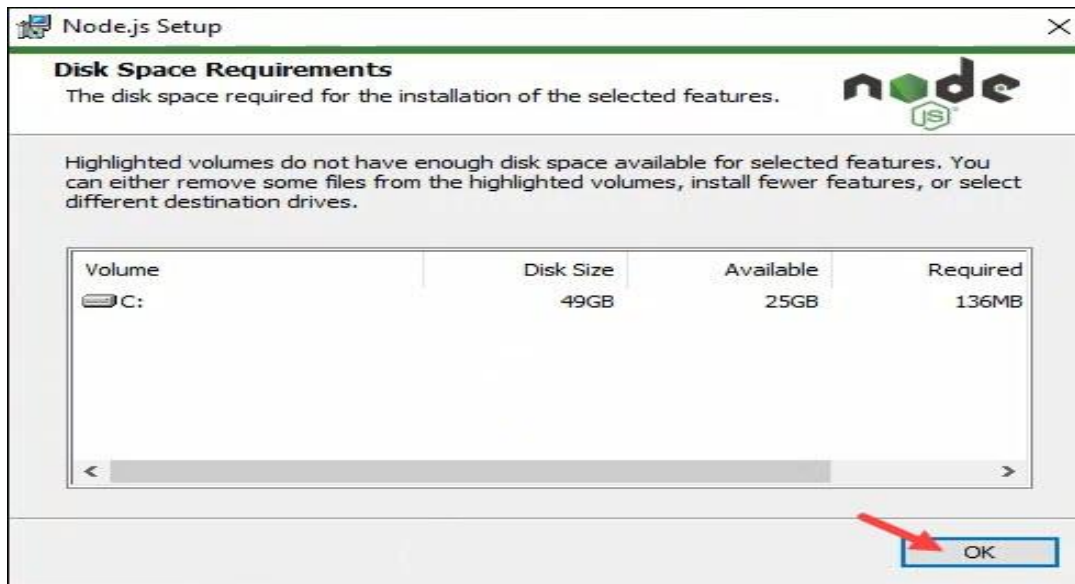
Leave the default location for a standard installation and click **Next** to proceed.

6. Select components to include or remove from the installation. The default options install Node.js, NPM, corepack, online documentation shortcuts, and add the programs to PATH.



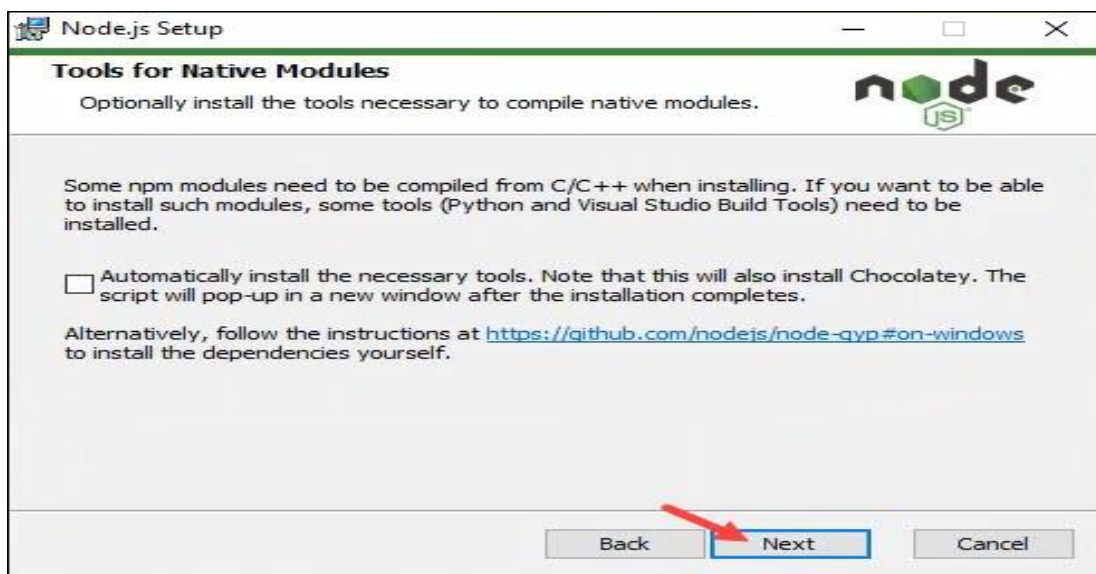
Customize the setup or click **Next** to accept the default values.

7. The following section shows the total required space for the installation and the available space on disk.



Click **OK** to proceed. Select a different disk or install fewer features if the installation does not allow proceeding.

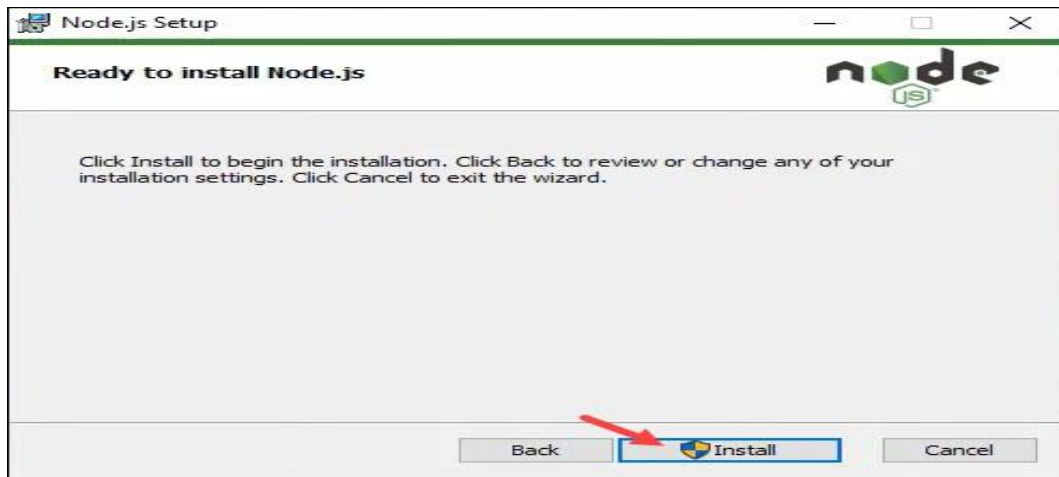
8. Choose whether to install additional dependencies for compiling native modules. Some NPM modules compile from [C/C++](#) and require additional tools to function correctly ([Python](#), Visual Studio Build Tools, and Chocolatey).



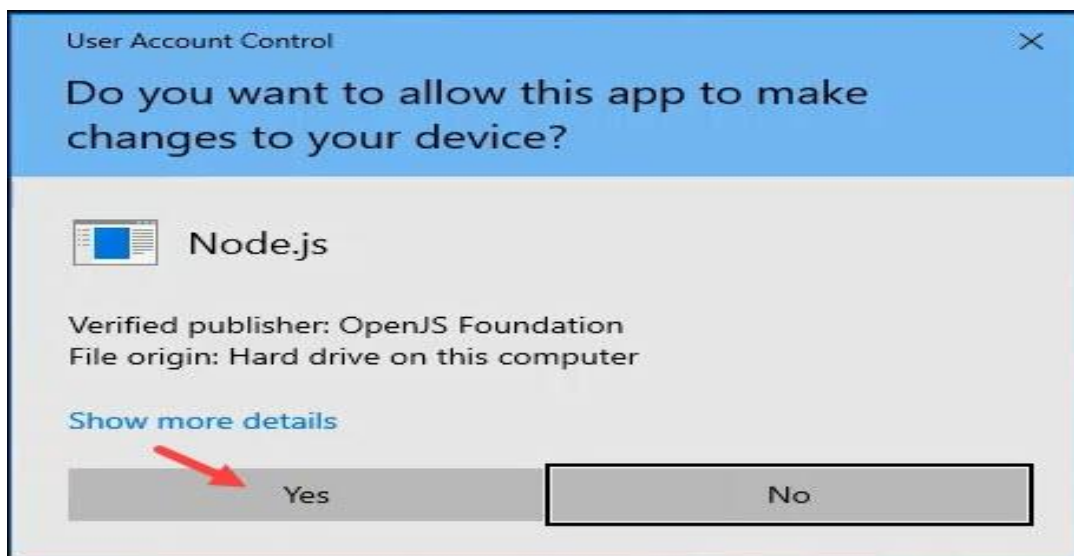
If you use such modules, select the checkbox and click **Next**. The selection of this option starts an installation script after the Node.js installation is complete.

For a simple installation, skip this step and click **Next** to proceed.

7. Click the **Install** button to start the installation.

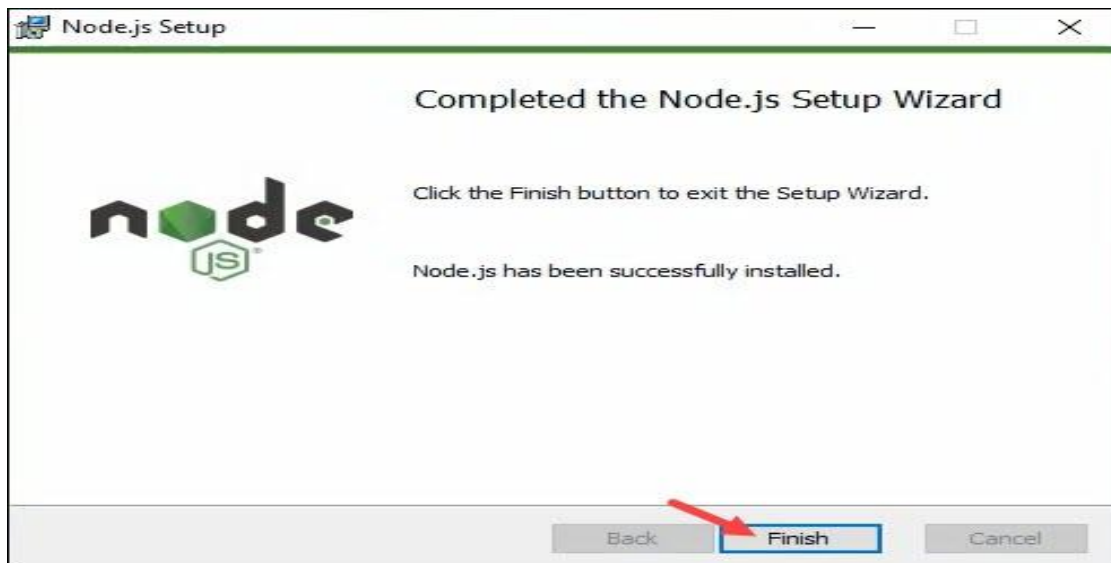


8. The installer prompts for administrator confirmation to make changes to the device.



Enter the administrator password if prompted and click **Yes** to continue.

9. The installation takes some time. When it is complete, the final screen shows a success message.



Click **Finish** to complete the installation and close the installer.

PROGRAM:

```
console.log("Hello everyone");
```

Output: Hello everyone

Javascript operation:

JS String: JavaScript String is a sequence of characters, typically used to represent text. It is enclosed in single or double quotes and supports various methods for text manipulation.

Syntax: let str= " ";

Program:

```
Let num="123";
```

```
Console.log("typeof num");
```

OUTPUT:

'123'

JS number: In JavaScript, numbers play an important role, and understanding their behavior is essential for effective programming. Let's explore the various aspects of numeric types in JavaScript.

1. The Only Numeric Type

As we know JavaScript has only one numeric type: the **double-precision 64-bit binary format IEEE 754** means that it doesn't differentiate between integers and floating-point numbers explicitly. Instead, it uses a unified approach for all numeric values.

- Integers and floating-point numbers are both represented using this format.
- The numeric precision is **53 bits**, allowing for an accurate representation of integer values ranging from $-2^{53} + 1$ to $2^{53} - 1$.

2. Scientific Notation

JavaScript allows writing extra-large or extra-small numbers using scientific (exponent) notation.

3. Integer Precision:

Integers (numbers without a period or exponent notation) are accurate up to 15 digits.

4. Floating Point Precision:

Floating point arithmetic is **not always 100% accurate** due to binary representation limitations.

JS Boolean:

Boolean is a datatype that returns either of two values i.e. true or false. In JavaScript, Boolean is used as a function to get the value of a variable, object, conditions, expressions, etc. in terms of true or false.

Note: A variable or object which has a value is treated as a **true** boolean value. '0', 'NaN', empty string, 'undefined', and 'null' is treated as **false** boolean values.

Program:

```
Let boy=true;
```

```
Console.log(boy);
```

OUTPUT:

1

JS array:

An array in JavaScript is a data structure used to store multiple values in a single variable. It can hold various data types and allows for dynamic resizing. Elements are accessed by their index, starting from 0.

Program:

```
Let arr=["123",123,12.43, true];  
Console.log("arr");
```

OUTPUT:

```
[‘123’,123, 12.43,1]
```

JS object:

Program:

```
Let ob= { My self :”Aman”,  
          Age:”20”,  
          Girl: true,  
          Money: “1500”,}  
Console.log("ob");
```

OUTPUT:

```
My self :’Aman’, Age:’20’, Girl: true, Money: ‘1500’
```

