

Installing Node.js

We will learn some Node.js (often just called "Node"), and we will also be using NPM (Node Package Manager) to install JavaScript packages. NPM is separate from Node but is bundled with Node, so the easiest way to use NPM is to install Node. Some tools we use for development also require Node to be installed.

There are many versions of Node and over time you will find yourself developing on different projects that are all running different Node versions. To solve this problem, there is a tool called NVM (don't get this confused with NPM). NVM means "Node Version Manager" and that's what it does. So rather than install NodeJS directly, you should install NVM and let it manage Node for you.

If you have installed Node directly on your computer in the past, I recommend that you uninstall it and switch to using NVM now. If you type the command `which node` and don't see `nvm` in the path that is printed, then you probably installed Node directly.

Installing NVM

To confuse matters a bit, there are two different NVMs. They have different maintainers but are very similar. Choose which one to use based on the operating system you are using:

- NVM for Windows: <https://github.com/coreybutler/nvm-windows#installation-upgrades>
- NVM for Linux or Mac OSX: <https://github.com/nvm-sh/nvm#installing-and-updating>

For Windows, there are also more detailed instructions here:

<https://docs.microsoft.com/enus/windows/nodejs/setup-on-windows>

Using NVM

You use NVM by typing commands in a terminal window.

Change the Current Node Version

Command: `nvm use <version or version alias>`

Examples:

```
nvm use 12.18.3
nvm use stable
nvm use default
```

If that version is installed, it will now be the version that is used whenever you type a command that begins with `node`, and the version of NPM that ships with that version of Node will be used whenever you type a command that begins with `npm`.

NOTE: Changing the current Node.js version affects your entire computer. It is not specific to the directory you are in when you type the `nvm use` command.

If that version is not installed, it will tell you to run `nvm install` and you should do so.

Install a Node Version

Command: `nvm install <version or version alias>`

Examples:

```
nvm install 12.18.3
nvm install stable
nvm install default
```

The specified version will be installed AND made current.

Using .nvmrc

When you create a project, it can be good to mark the version of Node that you are expecting people to run it with. To do that, we use a file named `.nvmrc` at the root of any folder.

All you need to type in the file is the version you are expecting.

With that file in place, anyone who downloads your project folder can simply type `nvm use` without specifying a version, and it will grab the correct version from the file.

Confirming Which Node.js Version is in Use

```
node -v  
npm -v
```

Updating NPM

NPM is actually just a global NPM package. Global NPM packages are installed with `npm install -g`. So you can update NPM using `npm install -g npm`. (Yes, it uses itself to install a new version of itself.)

IMPORTANT: All global NPM packages *are specific to the current version of Node.js*. This means that if you `npm install -g` some package and then `nvm use` a different Node.js version, the package you installed may no longer be available or may be at a different version.