Lesson Weekend

Intermediate JavaScript (/intermediate-javascript) / Object-Oriented JavaScript (/intermediate-javascript/object-oriented-javascript) / Installing and Using Node Version Manager

Text

Node.js is a **runtime** for JavaScript, which is an environment that lets us run JS code outside of a web browser, typically on a server. We'll be using Node to run various JavaScript code outside of our browsers, and to make use of its JavaScript package manager called npm, or Node Package Manager, to quickly install and use JavaScript libraries in the projects we build.

We're not quite ready to jump into the details of Node and npm — we'll do that in an upcoming course section. Instead, in this lesson we'll learn how to install and use Node Version Manager, which will prepare us for upcoming coursework.

Installing and Using Node Version Manager

Node Version Manager (NVM) is a tool that manages Node versions: it allows you to install multiple versions of Node.js (https://nodejs.org/en/docs) and switch between them. A tool like

NVM is helpful because you may need a different version of node, depending on the project you are working on. For example, your internship or job may require you to use a different version of Node than the one we use in the LearnHowToProgram.com curriculum. In cases like these, NVM offers you the ability to easily install different versions and switch between them from the command line.

Uninstalling Node

Before getting started you will need to completely uninstall Node for your device. If you haven't installed node yet, then skip to the next section. If you are not sure whether Node is installed on your computer, you can easily check by entering the following into the command line:

```
$ node -v
v18.16.0
```

The command `node -v` checks the version of Node that's installed on your computer. If you see a version number like `v18.16.0` returned, then node is installed on your computer. Note that your version of node may be different from the example above. If you get a message about "node: command not found", it means your computer does not recognize the `node` command because node is not installed (or possibly there was an error in the installation process).

To uninstall node on a Windows machine, follow these steps:

- Open the *Control Panel*.
- Find and select *Programs* or *Programs and Features*.
- Find and select the option to *uninstall a program*.
- Within the list of programs search for **Node.js**, and double click it to uninstall.
- Proceed with the uninstallation process.

- To double check that node is no longer installed on your machine, exit out of GitBash and reopen it. Then enter `node -v` and verify that you receive a message similar to “node: command not found”.

To uninstall node on a Mac machine, we’ll use Homebrew, the same package manager tool for MacOS that we used to initially install Node.js in the LearnHowToProgram.com curriculum. If you installed Node prior to starting at Epicodus, research the best method to uninstall Node for your personal environment. To uninstall Node on a Mac using Homebrew, follow these steps:

- Open your Terminal application.
- Enter `brew uninstall --force node`.
- To double check that node is no longer installed on your machine, completely shut down the Terminal application (including all windows and the application itself so that it isn’t still operating in the background) and reopen it. Then enter `node -v` and verify that you receive a message similar to “command not found: node”.

Installing `nvm-windows` and Node.js on a Windows

For Windows users, download nvm-windows by following these instructions:

- Visit the site https://github.com/coreybutler/nvm-windows/releases (https://github.com/coreybutler/nvm-windows/releases).
- Under the latest released version (at the top of the page), scroll down to *Assets* and click on `nvm-setup.exe` to download the setup wizard.
- Once `nvm-setup.exe` is downloaded, open the file and click through the setup wizard.
- Once you complete the setup wizard, type `nvm` into your GitBash terminal to verify that nvm-windows has been installed. (You may need to restart your GitBash terminal or computer).

- Now we are ready to install different versions of Node by using the script `nvm install <version>`, where `<version>` is the version number of Node that we want our computer to use. We'll use this command now to install the recommended version number of Node.js for the LearnHowToProgram.com curriculum. In your GitBash terminal, enter in the following command: `nvm install 16.13.1`.
- After Node version 16.13.1 is installed, we need to instruct nvm to use that version with the command `nvm use <version>`. Within your GitBash terminal, enter in the following command now: `nvm use 16.13.1`.

Here are other useful commands to use to manage your Node versions with nvm-windows:

- You can see a list of all your installed versions by using the command `nvm list`.
- You can switch between different Node.js versions by using `nvm use <version>`, where `<version>` is the version number of Node that you wish to use.
- Check out other nvm commands by entering `nvm` into the command line.

Finally, check out these resources for more information about Node versions and nvm-windows:

- See all of the versions of Node that you can install via nvm-windows, check out the Node documentation: https://nodejs.org/en/docs (https://nodejs.org/en/docs). It's recommended to use Node versions that are listed as `lts` for "long term support", which are the stable versions of node that are supported for a long time — but not forever! To see a list of currently supported Node versions and their end of life dates, visit the node releases page (https://nodejs.dev/en/about/releases/).
- Check out the nvm-windows documentation via their GitHub repo README (https://github.com/coreybutler/nvm-windows).

Installing `nvm` and Node.js on Mac and Linux

For Mac and Linux users you will be using a tool called nvm, located at https://github.com/nvm-sh/nvm (https://github.com/nvm-sh/nvm).

To install or update nvm, we'll use a script. Open your Terminal application and enter in the following command and hit enter:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
```

The command `curl` invokes the tool cURL, which is a command line tool that lets us make requests to servers through the command line. We won't be using cURL regularly in this program, and it is an advanced topic right now, since we have not yet learned about making requests to servers. If you are eager to learn more now, start with this article from IBM (https://developer.ibm.com/articles/what-is-curl-command/).

After running the nvm installation script, you'll need to verify that nvm is correctly installed and possibly troubleshoot any issues. Follow these steps:

- Take a screenshot of the installation script in your Terminal in case you need to troubleshoot your installation. Then, completely exit out of all Terminal application windows and completely close down the Terminal application.
- Reopen Terminal, and enter in the following command: `command -v nvm`. If you should see `nvm` returned to you in the command line, this confirms that nvm is installed correctly.
- If you do not see `nvm` as a response, but instead a message like "nvm: command not found" visit this section of the nvm docs (https://github.com/nvm-sh/nvm#install--update-script) and look for either "Troubleshooting on Mac" or "Troubleshooting on Linux" for troubleshooting steps to follow. Make sure to

record any troubleshooting you do so you can share it with your instructor if you cannot resolve your installation issue.

Next, we'll install the version of Node that we'll use in the program. We'll also learn about other useful commands we can use with nvm:

- To install a specific version of node use the command `nvm install <version>` where `<version>` is the version of Node you want to install. We'll use this command now: open terminal and enter `nvm install 16.13.1`. If this is the first version of Node that you are installing, nvm knows to automatically use it as the default version.
- We can verify that Node was successfully installed by running `node -v` in the command line. We should see `v16.13.1` returned to us.
- You can list the version(s) of Node you have installed on your computer with the `nvm ls` command. This command will list your installed versions of node at the top of the list, including the default version that's currently being used, as well as a list of Node versions that are not installed on your machine, but are listed as `lts` for "long term support", which are the stable versions of node that are supported for a long time — but not forever! To see a list of currently supported Node versions and their end of life dates, visit the node releases page (https://nodejs.dev/en/about/releases/).
- You can list all available versions of Node using `nvm ls-remote`. This will be a very large list.
- To switch the default version of Node that your computer is using, use the command `nvm use <version>`, where `<version>` is the version number of Node that you want to set as default.
- To learn about other usage commands, visit the nvm documentation (https://github.com/nvm-sh/nvm#usage).

If you run into any issues or questions, reach out to your instructor for help and further guidance.

Previous (/intermediate-javascript/object-oriented-javascript/object-oriented-javascript-objectives)

Next (/intermediate-javascript/object-oriented-javascript/homework-and-class-structure)

Lesson 2 of 33
Last updated March 23, 2023

disable dark mode

(http://www.epicodus.com)