**NodeJS Install**

**Installing NodeJS and Setting Up Your Development Environment**

Welcome back! Before we dive deeper into the course material, we need to ensure your development environment is set up correctly. A crucial piece of this setup is installing NodeJS. This lesson will guide you through understanding what NodeJS is, why you need it, and how to install it using the recommended tools.

Remember the exercise of writing down your motivations for taking this course? Actively engaging with tasks like that often correlates with greater success, both in this course and in your broader journey within the industry. Keep that momentum going as we tackle this essential setup step.

**Leveraging AI for JavaScript Learning**

Throughout this course, we will be working extensively with JavaScript. If you're new to JavaScript or encounter concepts you don't fully grasp, remember that Artificial Intelligence (AI) tools are exceptionally proficient with this language.

Don't hesitate to use AI assistants like DeepSeek (chat.deepseek.com), Claude, or others to ask questions, understand code snippets, or even generate basic examples. For instance, asking an AI to "write a basic javascript codebase that will tell me the square root of a number" yields quick and useful results.

Beyond AI, utilize all available resources: Google, Reddit, the course discussion forums, and any other help channels provided. **Strive to understand every single line of code presented.** If something is unclear, ask questions until it makes sense. This foundational understanding is critical.

**What is NodeJS?**

We will heavily rely on a technology called NodeJS, which you can find at nodejs.org. NodeJS is formally defined as a "JavaScript runtime environment." While this terminology might seem a bit confusing initially, the core concept is straightforward.

The JavaScript code that runs directly within your web browser (like Chrome, Firefox, etc.) operates in the browser's built-in environment. It "just works" when embedded in a webpage. However, if you want to run JavaScript code directly on your computer – perhaps as a standalone script or to power a server – you need a different environment capable of executing that code outside the browser context.

This is where NodeJS comes in. It provides the necessary "runtime environment" to execute JavaScript files locally or on a server. Therefore, installing NodeJS is a mandatory prerequisite for proceeding with the course modules that involve running JavaScript code outside of a web browser.

**Installing NodeJS via NVM and PNPM**

Installing development tools can sometimes be the most challenging part of getting started. If installing NodeJS or other tools like VS Code takes time – even a full day – don't get discouraged. Patience is key during this setup phase.

We will perform the installation using the terminal, which is conveniently integrated into Visual Studio Code (VS Code).

1. **Navigate to the Official Source:** Open your web browser and go to the official NodeJS website's download page: nodejs.org/en/download.
2. **Select Your Operating System:** Choose the appropriate operating system (macOS, Windows, Linux).
   * **Important Windows Note:** If you are using Windows, it is highly recommended that you use the Windows Subsystem for Linux (WSL). If you are using WSL, select the **Linux** installation instructions on the NodeJS website.
   * For macOS or native Linux environments, select the corresponding instructions.
3. **Choose the Installation Method:** The website will present different ways to install NodeJS. We strongly recommend using **NVM (Node Version Manager)**. NVM is a tool that makes it easy to install and switch between different versions of NodeJS, which is incredibly helpful in development. Select the NVM option.
4. **Select the Package Manager:** NodeJS comes with a default package manager called npm. However, for this course, we recommend using **PNPM**. On the installation instructions page, ensure you change the package manager selection from the default (npm) to pnpm.
5. **Execute the Installation Script:** Based on your selections (OS, NVM, PNPM), the website will generate a command-line script. Carefully copy this entire script.
6. **Run in VS Code Terminal:** Go back to VS Code, open the integrated terminal (usually via the Terminal menu -> New Terminal), paste the copied script into the terminal, and press Enter/Return to execute it. This script will typically download and run the NVM installer, then use NVM to install the recommended NodeJS version, and finally configure PNPM.

Follow the specific instructions provided by the script. This might involve one or more commands to copy and paste.

**Verifying Your Installation**

After the installation script finishes, there's a crucial step:

1. **Restart Your Terminal:** Close your current terminal session in VS Code (you can often click the trash can icon associated with the terminal panel) and then open a new one (Terminal -> New Terminal). This ensures that your system recognizes the newly installed software and any changes made to your system's environment paths.
2. **Check Node Version:** In the *new* terminal window, type the following command and press Enter:

node --version

You should see a version number printed, such as v22.x.x (the exact version depends on the LTS version recommended by nodejs.org at the time of installation).

1. **Check PNPM Version:** Next, type the following command and press Enter:

pnpm --version

You should see another version number printed, for example, 10.x.x.

If both commands successfully output version numbers, congratulations! You have successfully installed NodeJS and PNPM using NVM.

**Additional PNPM Resource**

If you encounter issues specifically with PNPM or want to learn more about its installation options, you can refer to its official documentation: pnpm.io/installation.

With NodeJS installed, your development environment is one step closer to being ready.