**VS Code Quickstart**

**Getting Started with VS Code for web3 Development**

Welcome! This guide will walk you through setting up Visual Studio Code (VS Code) for our development journey, focusing on the tools and workflows essential for web3 development, particularly within the context of courses like the Vyper curriculum this lesson is excerpted from. We'll cover using the integrated terminal, installing necessary tools like Git, navigating your file system via the command line, and setting up a dedicated project folder. This setup is designed to work consistently whether you're using macOS, a Linux distribution, or Windows with the Windows Subsystem for Linux (WSL) enabled.

**Using the VS Code Integrated Terminal**

One of VS Code's most powerful features for developers is its built-in terminal. This allows you to run commands, scripts, install software, and interact with tools like Git without ever leaving your code editor, streamlining your workflow significantly.

**How to Open the Terminal:**

1. **Menu:** Go to the top menu bar and select Terminal -> New Terminal.
2. **Keyboard Shortcut:**
   * On macOS: Control +   (backtick) orCommand+   (backtick)
   * On Windows/Linux: Control +   ` (backtick)

The terminal panel will appear at the bottom of your VS Code window. It runs within a shell program (like Bash or Zsh). While different shells exist, the basic commands we'll use in this course will function identically across the common shells found on macOS, Linux, and WSL. You can check your current shell by typing echo $SHELL and pressing Enter.

**Ensuring Cross-Platform Compatibility**

A key advantage of this setup guide is its OS-agnostic nature for users on macOS, Linux, or Windows with WSL. By leveraging the Linux-based environment provided natively (on Mac/Linux) or through WSL (on Windows), the terminal commands and tool interactions described from this point forward will be the same across all these platforms. This simplifies learning and ensures consistency throughout the course. If you are on Windows, ensure you have WSL installed and are operating within the WSL environment in VS Code.

**Installing Git for Version Control**

Git is a fundamental tool for modern software development. It's a version control system that tracks changes to your code, allows collaboration, and is essential for managing projects. While you might not use it in the very first steps, it's required later in the course, so we recommend installing it now.

**Installation Steps:**

1. **Check if Git is Already Installed:** Open your VS Code terminal and type git --version. If it outputs a version number (e.g., git version 2.39.2), Git is installed, and you can skip the rest of this section.
2. **Install on Linux (Debian/Ubuntu):** If Git is not installed, run:

sudo apt update && sudo apt install git-all

1. **Install on Linux (Fedora/RHEL):** If Git is not installed, run:

sudo dnf install git-all

1. **Install on macOS:** Often, simply running git --version or just git in the terminal will trigger a prompt to install the Xcode Command Line Tools, which include Git. Follow the on-screen prompts. Alternatively, download the installer directly from the official Git website (git-scm.com/download/mac).
2. **Verify Installation:** After installation, run git --version again to confirm it was successful.

If you encounter installation issues, resources like Stack Overflow, the specific documentation for your OS, or the course's discussion forum are great places to seek help. Getting the environment set up can sometimes be tricky, so don't get discouraged!

**Mastering Basic Terminal Commands**

The terminal is your primary interface for interacting with your system's files and running development tools. Here are some essential commands:

* pwd (Print Working Directory): Shows the full path of the folder you are currently inside. Useful for orientation.

pwd

* ls (List): Lists the files and folders within the current directory.

ls

* cd (Change Directory): Navigates between folders.
  + cd ~ or just cd: Navigates to your home directory.
  + cd ..: Moves up one directory level (to the parent folder).
  + cd <folder\_name>: Moves into a specific folder within the current directory. (e.g., cd Documents)
* mkdir (Make Directory): Creates a new folder.

mkdir my\_new\_project

* touch (Touch): Creates a new empty file or updates the modification timestamp of an existing file.

touch my\_file.txt

* clear: Clears the terminal screen, removing previous output.

clear

* **Clear Screen Shortcut:**
  + macOS: Command + K
  + Windows/Linux: Control + K

Practice using these commands to become comfortable navigating your file system.

**Setting Up Your Project Workspace**

It's crucial to keep your projects organized. We recommend creating a dedicated main folder for all the material related to this course.

1. **Navigate to Your Preferred Location:** Use cd to go to the directory where you usually store your development projects (e.g., cd ~/Documents or cd ~/dev).
2. **Create the Course Folder:** Use mkdir to create a specific folder for this course. We'll use full-stack-web3-cu as an example:

mkdir full-stack-web3-cu

1. **Navigate Into the Project Folder:**

cd full-stack-web3-cu

1. **Open the Folder in VS Code:** Now that you are inside your project folder in the terminal, you can open this specific folder as a workspace in VS Code. This focuses the editor's file explorer and terminal context to this directory.
   * **Using the Terminal:** Type code . (the dot represents the current directory). This might require ensuring the code command is added to your system's PATH during VS Code installation.

code .

This will typically open a new VS Code window scoped to the full-stack-web3-cu folder.

* + **Using the UI:** Alternatively, in VS Code, go to File -> Open Folder... and navigate to select your full-stack-web3-cu folder.

1. **Create a Notes File:** It's helpful to keep notes within your project. Let's create a README.md file using the terminal (ensure your terminal is now operating within the full-stack-web3-cu directory):

touch README.md

You should see README.md appear in the File Explorer panel on the left side of VS Code. You can click it to start taking notes.

**Managing Terminal Sessions: Kill vs. Hide**

In the VS Code terminal panel, you'll notice icons near the top right. Understanding the difference between hiding and killing a terminal session is important:

* **Hiding (**X**icon or**Control/Command + `` **):** Clicking the X icon or using the toggle shortcut (Control+`` or Command+```) simply hides the terminal panel. The underlying terminal process continues to run, preserving its state and command history. You can bring it back with the shortcut or Terminal -> New Terminal (which often reopens the existing one).
* **Killing (Trash Can icon):** Clicking the trash can icon terminates the terminal session completely. The process is stopped, and its history is lost. You'll need to start a fresh session if you need a terminal again.

Generally, hiding is preferred if you plan to return to the same session. Killing is useful if a process is stuck or you want a completely clean slate.

**Optional: Boosting Productivity with AI Assistants**

VS Code's extensibility allows for powerful add-ons. AI coding assistants can significantly speed up development and learning. While optional, they are highly recommended.

* **GitHub Copilot & Copilot Chat:** These are popular extensions from GitHub/Microsoft.
  + **Copilot:** Provides real-time code suggestions and autocompletion as you type. Activate suggestions by typing, and accept them using the Tab key.
  + **Copilot Chat:** Offers a chat interface directly within VS Code, allowing you to ask coding questions, get explanations, debug code, and more, similar to using external tools like ChatGPT or Claude.
* **Installation:** Open the Extensions view in VS Code (usually the square icon on the left sidebar). Search for "GitHub Copilot" or "GitHub Copilot Chat" and click "Install". You may need to sign in with your GitHub account and potentially set up billing (Copilot is a paid service, though free trials or student packs might be available).
* **Alternatives:** Many other AI coding assistants exist, some with free tiers. Explore the VS Code Marketplace for options.

These tools can help you write boilerplate code faster, understand complex concepts, and learn best practices directly within your development environment.

You now have a solid VS Code setup configured for the course. You're familiar with the integrated terminal, essential commands, Git installation, project organization, and optional AI tools. This foundation will serve you well as we dive into more complex web3 development topics.