

Skill Labs

BlockChain Track

Installations

Venue

ECE Seminar Hall ▾

Session Time

02:30PM Onwards ▾

Subject

Blockchain Technology ▾

Net Installation Checklist :

- ☐ VS Code
- ☐ Python 3.13 + packages (pyteal, py-algorand-sdk)
- ☐ WSL (Ubuntu) **[NOTE: MacOS/Linux Distro Users need not install WSL]**
- ☐ Node.js + npm

- ☐ Remix IDE
- ☐ Wallets (Phantom + MetaMask)
- ☐ Rust (rustup)
- ☐ Solana CLI
- ☐ Ethereum (Geth)

Dec 29, 2025

Installation #0: Visual Studio Code

- ☐ Installation Link: [*VS Code Installer*](#)
- ☐ Download the Installer, Run the Installation.

Install with:

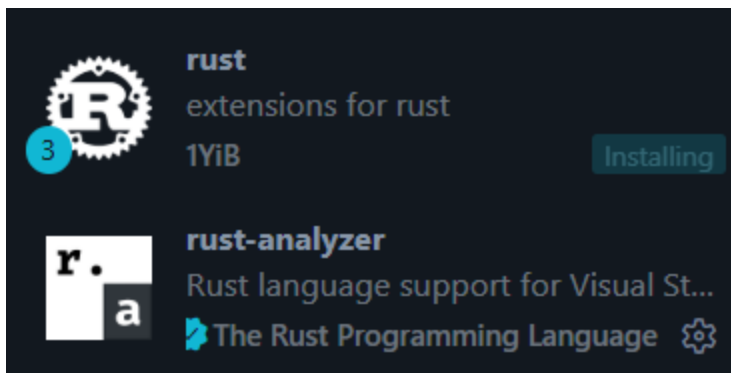
- ☐ Add to PATH
- ☐ Register Code as editor
- ☐ Open with Code

Open VS Code → Extensions tab → install:

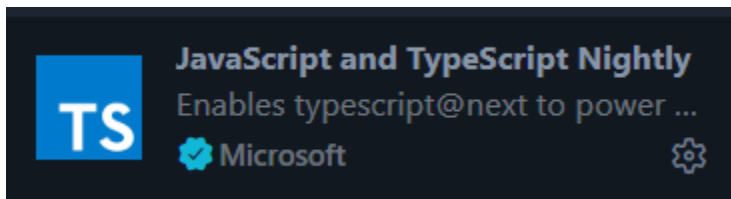
- ☐ Python



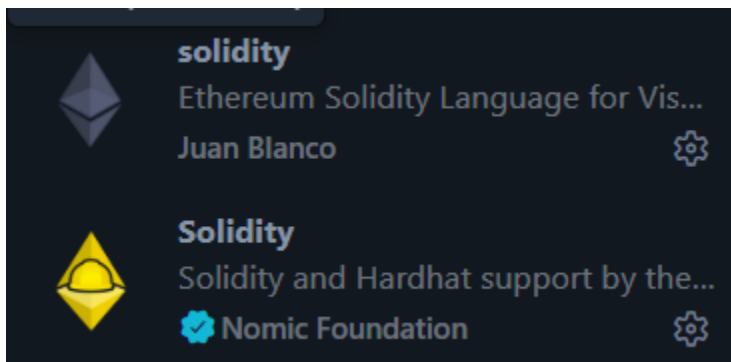
- ☐ Rust Analyzer



☐ Typescript



☐ Solidity



☐ WSL



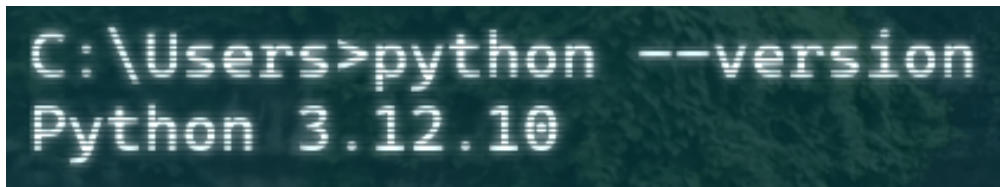
Installation #1: Python

[<=v3.13 + 2 Packages]

- ☐ Installation Link: [Python v3.13 Installer](#)
- ☐ Download the Installer, Run the Installation.

During installation:

- **Add Python to PATH** (kinda IMPORTANT)
- Install for all users
- ☐ Run **\$python --version**



```
C:\Users>python --version
Python 3.12.10
```

Works :)

- ☐ If version is greater than 3.13, Uninstall & Re-install. [Or Opt.2 : Change the config file]
- OR: Create a virtual envi file {`.myenv`}
- ☐ Run **\$pip install pyteal py-algorand-sdk**

Installation #2: WSL

[NOTE: MacOS/Linux Distro Users need not install WSL]

(Windows Subsystem for Linux) [Ubuntu Linux Environment]

Installation Link: *Built-in Windows Feature*

- ☐ Open **PowerShell (Run as Administrator)**
Run: **\$wsl --install**
- ☐ Restart the system.

Ubuntu will open automatically.

Set:

☐ Username

☐ Password

Run: `$uname -a`

For Verification..

Installation #3: Node.js + npm

[JavaScript Runtime for Ethereum Tooling - Windows {Not necessarily on wsl}]

Installation Link: [Node Installer Downloads](#)

From: <https://nodejs.org>

Download **LTS version**.

Run the installer (default options).

Verify:

`node --version`

`npm --version`

If unsuccessful,

Run:

```
sudo apt install npm
```

```
sudo apt install node.js
```

Installation #4: Remix IDE

[Solidity Smart Contract IDE]

Access Link: <https://remix.ethereum.org>

No installation required (browser-based).

OR [Already installed]

VS Code Extension:

Install **Remix IDE** extension from VS Code Marketplace.

Installation #5: Wallet Setup

[Blockchain Accounts]

MetaMask (Ethereum Wallet)

Installation Link: <https://metamask.io>

Install browser extension.

Create wallet.

Save **Secret Recovery Phrase (12 words)** offline.

Phantom (Solana Wallet)

Installation Link: <https://phantom.app>

Install browser extension.

Create wallet.

Save **Seed Phrase** offline.

Installation #6: Rust (via rustup)

[Required for Solana Smart Contracts]

Run **inside WSL (Ubuntu)**

```
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Choose:

```
1) Proceed with installation
```

Reload shell:

```
source ~/.cargo/env
```

Verify:

```
rustc --version
```

```
cargo --version
```

Installation #7: Solana CLI

[Solana Developer Tools]

Run **inside WSL**

```
sh -c "$(curl -sSfL https://release.anza.xyz/stable/install)"
```

Add to PATH:

```
export
```

```
PATH="/Users/test/.local/share/solana/install/active_release/bin:$PATH"
```

If you are using **Linux** or **WSL**, you must add the Solana CLI binary to your PATH so that the command is available in your terminal. To do so, follow the steps below:

a. Check which shell you are using:

```
echo $SHELL
```

- If the output contains /bash, use .bashrc.
- If the output contains /zsh, use .zshrc.

b. Run the appropriate command, based on your shell.

For Bash (bashrc):

```
echo 'export  
PATH="$HOME/.local/share/solana/install/active_release/bin:$PATH"' >>  
~/.bashrc
```

```
source ~/.bashrc
```

For Zsh (zshrc):

```
echo 'export  
PATH="$HOME/.local/share/solana/install/active_release/bin:$PATH"' >>  
~/.zshrc
```

```
source ~/.zshrc
```

4. Restart your terminal or run the following command to refresh the terminal session:

```
source ~/.bashrc # If using Bash
```

```
source ~/.zshrc # If using Zsh
```

5. Verify that the installation succeeded by checking the Solana CLI version:

```
solana --version
```

You will see output like the following:

```
solana-cli 2.2.12 (src:0315eb6a; feat:1522022101, client:Agave)
```

Verify:

```
solana --version
```

(FURTHER)!!

Create wallet:

```
solana-keygen new
```

Set Devnet:

```
solana config set --url https://api.devnet.solana.com
```

Installation #8: *Ethereum (Geth)*

[Local Ethereum Node]

Run **inside WSL**:

```
sudo add-apt-repository -y ppa:ethereum/ethereum  
sudo apt update  
sudo apt install ethereum
```

Verify:

```
geth version
```

Run dev chain:

```
geth --dev
```

Final Test #9:

VS Code ↔ WSL Integration

[Recommended Setup]

From WSL:

```
$cd /mnt/c/Users/<username>/blockchain  
###blockchain can be your folder name
```

```
$code .
```

VS Code opens connected to WSL.

[Basically, we are using Linux tooling onto our Windows files in order to still code with them on VS Code]

Problem Statements DocLink:

[Blockchain Track PS 2025-26](#)