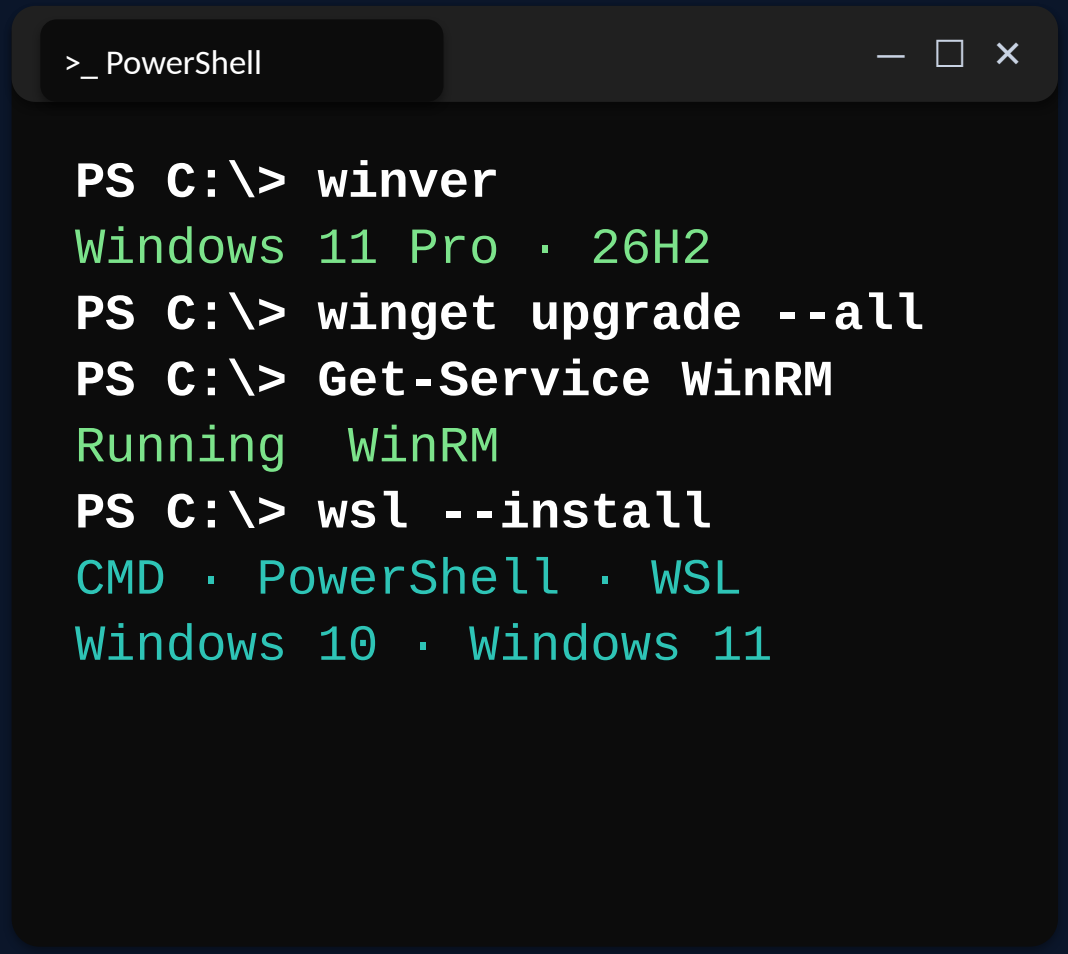


LESSON 6 OF 12

Windows 10 & 11

Desktop · Workstation · Laptop

Developer Setup: Compilers, Interpreters, VS Code & WSL

Full Course · Windows 11 26H2 · Windows 10 22H2

What You Will Learn in Lesson 6

- The modern Windows developer toolbox
- Languages: Python, C and C++, .NET, Java, Node.js, Go, Rust
- Python the right way: py launcher and virtual environments
- Compiling C, C++ and C# on Windows
- WSL 2: real Linux inside Windows
- VS Code and Git essentials
- Dev Drive, developer settings and Windows Sandbox

The Windows Developer Toolbox

Tool	Why you want it
Windows Terminal + PowerShell 7	A modern command line (Lesson 4)
VS Code	Free editor for every language; extensions; remote and WSL support
Git	Version control; GitHub, GitLab, Azure DevOps
WSL 2	Ubuntu, Debian, Kali and more, running a real Linux kernel
Visual Studio / Build Tools	MSVC C and C++ compiler, .NET, Windows apps
Docker Desktop or Podman	Containers (uses WSL 2 underneath)

Languages on Windows

Language	Install with winget (example ID)	Check
Python 3.15	Python.Python.3.15	py --version
Node.js LTS	OpenJS.NodeJS.LTS	node -v ; npm -v
Java (OpenJDK)	Microsoft.OpenJDK.21	java --version
.NET SDK 10	Microsoft.DotNet.SDK.10	dotnet --info
C / C++ (MSVC)	Microsoft.VisualStudio.2022 .BuildTools	cl (Developer PowerShell)
Go / Rust	GoLang.Go / Rustlang.Rustup	go version ; rustc -V

Package IDs and newest versions change — confirm with winget search before installing.

Python the Right Way

```
PS C:\> py --list                    # every Python installed
PS C:\> py -3.15 --version
PS C:\> mkdir C:\Code\hello-py ; cd C:\Code\hello-py
PS C:\Code\hello-py> py -m venv .venv          # a private environment
PS C:\Code\hello-py> .\.venv\Scripts\Activate.ps1
(.venv) PS C:\Code\hello-py> python -m pip install requests pandas
(.venv) PS C:\Code\hello-py> python -c "import pandas; print(pandas.__version__)"
(.venv) PS C:\Code\hello-py> pip freeze > requirements.txt
(.venv) PS C:\Code\hello-py> deactivate
```

Compile C, C++ and C#

```
# Start > 'Developer PowerShell for VS'  (sets up the MSVC compiler)
PS C:\Code> cl /W4 /EHsc hello.cpp          # -> hello.exe
PS C:\Code> .\hello.exe
Hello from MSVC!

# Alternative: GCC through MSYS2 (UCRT64 shell) or inside WSL
$ gcc -Wall -O2 hello.c -o hello

PS C:\Code> dotnet new console -n HelloCS ; cd HelloCS
PS C:\Code\HelloCS> dotnet run
```

WSL 2: Real Linux Inside Windows

Command	What it does
wsl --install	Install WSL with Ubuntu (restart once)
wsl --list --online	Distributions you can install
wsl --install -d Debian	Add another distribution
wsl -l -v	Installed distros and WSL version
wsl --update / wsl --shutdown	Update WSL / stop all distros
\\wsl.localhost\Ubuntu and /mnt/c	Linux files from Windows / Windows files from Linux

A Quick WSL Session

```
PS C:\> wsl --install                          # first time only, then restart
PS C:\> wsl                                    # open your default distro
$ uname -a ; cat /etc/os-release | head -3
$ sudo apt update && sudo apt install -y build-essential python3-venv git
$ explorer.exe .                               # open this Linux folder in File Explorer
$ code .                                       # open it in VS Code (WSL extension)
$ exit
PS C:\> wsl -l -v
  NAME      STATE     VERSION
* Ubuntu    Stopped   2
```

VS Code Essentials

- Extensions: Python, C/C++, C#, PowerShell, WSL, Remote - SSH, GitLens
- Ctrl + ` opens the integrated terminal (PowerShell or WSL bash)
- Ctrl + Shift + P: the Command Palette for every action
- Built-in Git: stage, commit, push from the Source Control panel
- Debugging: breakpoints and variables for Python, C++, C#, Node.js
- Settings Sync with your GitHub or Microsoft account

Git Setup on Windows

```
PS C:\> git config --global user.name  "Your Name"
PS C:\> git config --global user.email "you@example.com"
PS C:\> git config --global core.autocrlf true     # CRLF on Windows, LF in the repo
PS C:\> git config --global init.defaultBranch main

PS C:\> ssh-keygen -t ed25519 -C "you@example.com"   # key for GitHub
PS C:\> Get-Content $HOME\.ssh\id_ed25519.pub        # paste into GitHub > SSH keys

PS C:\Code\hello-py> git init ; git add . ; git commit -m "First commit"
```

Dev Drive, Developer Settings and Sandbox

Feature	What it gives you
Dev Drive	ReFS volume tuned for code; Defender performance mode = faster builds
Create it	Settings > System > Storage > Disks & volumes > Create Dev Drive
Developer settings	Settings > System > Advanced: long paths, sudo, File Explorer + Git
Windows Sandbox	Test unknown tools and installers in a throwaway desktop
Hyper-V	Full VMs: Windows Server, Linux servers, test networks

Lab: Your Developer Workstation

```
PS C:\> winget install -e --id Python.Python.3.15
PS C:\> winget install -e --id OpenJS.NodeJS.LTS
PS C:\> winget install -e --id Microsoft.DotNet.SDK.10
PS C:\> wsl --install                       # restart, then create your Linux user

PS C:\> py -m venv C:\Code\lab\.venv ; C:\Code\lab\.venv\Scripts\Activate.ps1
(.venv) PS C:\> python -c "print('Python OK')" ; node -e "console.log('Node OK')"
PS C:\> dotnet new console -n Lab6 ; dotnet run --project Lab6
PS C:\> wsl -e bash -c "gcc --version | head -1"
```

Summary & Homework

- One winget command per language; confirm IDs with winget search
- Python: py launcher + a virtual environment per project
- MSVC via Developer PowerShell; GCC via MSYS2 or WSL
- WSL 2 = real Linux; keep Linux projects in the Linux file system
- Homework: push a small project to GitHub using an SSH key
- Homework: create a Dev Drive (or a VHD) and build a project on it

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

Slides for this lesson: MyWebUniversity.com/presentations

YouTube: @MyWebUniversity · @MyWebUniversity-Dari

Wahidullah Lutfy — Founder and Director of IT of MyWebUniversity, and CS teacher

Book: The Direct Path to Linux Ubuntu (Amazon · Barnes & Noble)

Learn it, then pass it on.