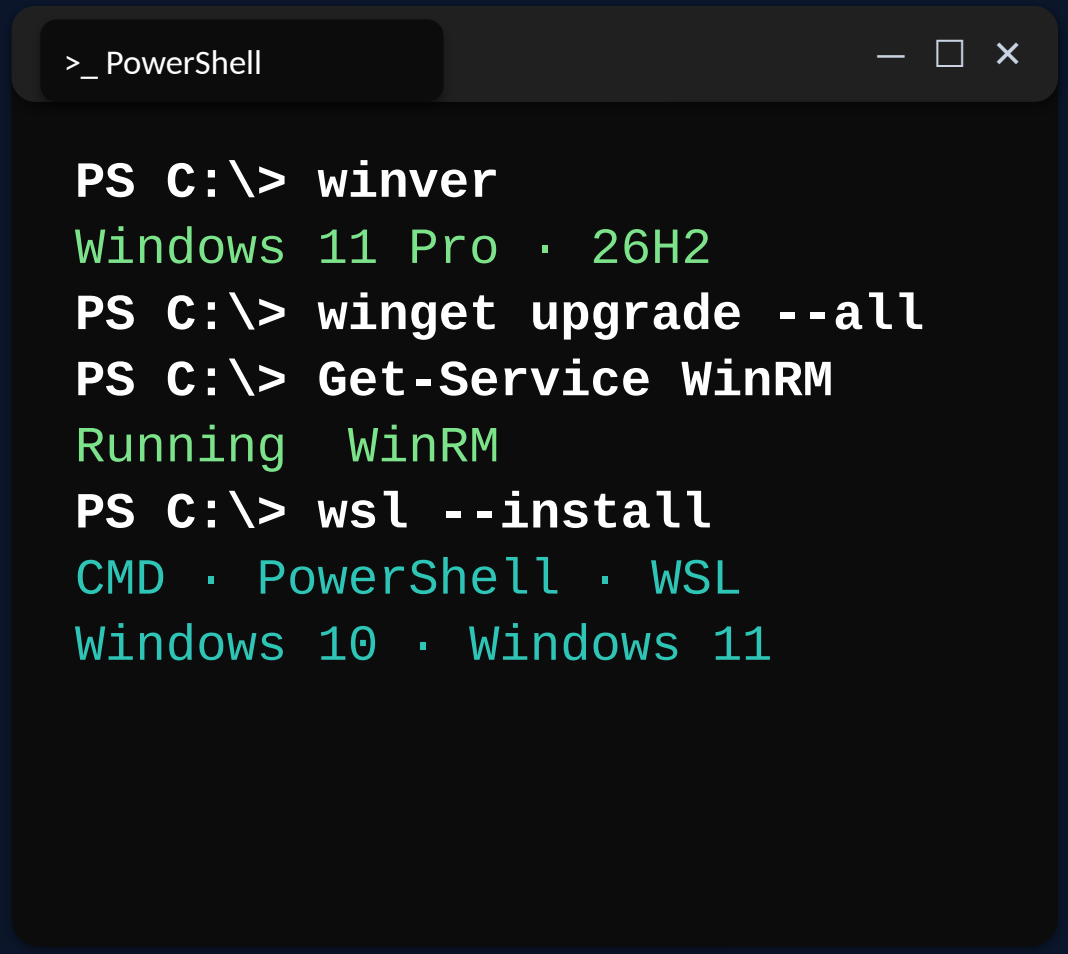


LESSON 5 OF 12

Windows 10 & 11

Desktop · Workstation · Laptop

Installing Software: winget, Microsoft Store, MSI & More

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

What You Will Learn in Lesson 5

- The ways software reaches a Windows PC
- Safety rules: official sources, signatures and SmartScreen
- winget: search, install, upgrade, uninstall, export and import
- MSI and EXE installers, including silent installs
- Where programs live, and the PATH variable
- Community managers: Chocolatey and Scoop
- Windows optional features and capabilities

How Software Reaches Windows

Method	What it is	Updates
Microsoft Store	Curated apps, MSIX packages, sandboxed	Automatic
winget	Windows Package Manager, built in (App Installer)	winget upgrade
MSI installer	Windows Installer package; good for silent installs	Vendor or IT tools
EXE installer	Vendor's own setup program	Vendor updater
Portable / ZIP	Unzip and run; no installation	Manual
Chocolatey / Scoop	Community package managers	choco / scoop update

Safety Rules for Installing Software

- Download only from the vendor's official site or a trusted manager
- Watch for fake 'Download' buttons and look-alike websites
- Check the digital signature: right-click > Properties > Digital Signatures
- Respect SmartScreen and UAC warnings — read before you click
- Use a standard account day to day; elevate only to install
- Uninstall what you no longer use, and keep the rest updated

winget Essentials

```
PS C:\> winget search "visual studio code"
PS C:\> winget show --id Microsoft.VisualStudioCode       # publisher, version, URL
PS C:\> winget install --id Microsoft.VisualStudioCode -e
PS C:\> winget install --id Git.Git -e --silent
PS C:\> winget list                       # everything installed (even non-winget)
PS C:\> winget upgrade                    # what has updates?
PS C:\> winget upgrade --all              # update everything
PS C:\> winget uninstall --id Git.Git -e
```

Rebuild a PC in Minutes: Export and Import

```
PS C:\> winget export -o C:\Lab\my-apps.json      # save your app list
PS C:\> Get-Content C:\Lab\my-apps.json | Select-Object -First 12

# On a new or freshly reset PC:
PS C:\> winget import -i C:\Lab\my-apps.json --accept-package-agreements

# Advanced: describe a whole machine in a YAML file (WinGet Configuration)
PS C:\> winget configure -f C:\Lab\dev-machine.winget
```

MSI and EXE Installers

Task	Command
Install silently, with a log	msiexec /i app.msi /qn /l*v C:\Logs\app.log
Uninstall an MSI	msiexec /x app.msi /qn (or the product code {GUID})
Repair an MSI	msiexec /fa app.msi
EXE silent switches	Vary by vendor: /S, /silent, /quiet — try setup.exe /?
Exit codes	0 = success; 3010 = success, restart required

Where Programs Live

Location	What goes there
C:\Program Files	64-bit applications for all users
C:\Program Files (x86)	32-bit applications
%LOCALAPPDATA%\Programs	Per-user installs (no admin needed), e.g. VS Code user setup
C:\Program Files\WindowsApps	Microsoft Store (MSIX) apps — protected
%APPDATA%, %LOCALAPPDATA%	Each user's settings and caches
Registry ...\Uninstall	The list behind 'Installed apps'

Find Installed Software and Fix PATH

```
PS C:\> $keys = 'HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*',
                'HKCU:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*'
PS C:\> Get-ItemProperty $keys | Where-Object DisplayName |
          Sort-Object DisplayName | Select-Object DisplayName, DisplayVersion

PS C:\> $env:Path -split ';'              # folders searched for commands
PS C:\> where.exe python                  # which python runs?
PS C:\> Get-Command code | Select-Object Source
# Edit PATH: sysdm.cpl > Advanced > Environment Variables (then open a new terminal)
```

winget vs Chocolatey vs Scoop

	winget	Chocolatey	Scoop
Made by	Microsoft (built in)	Community / company	Community
Admin rights	Often (per app)	Usually	No (user folder)
Package count	Very large	Very large	Large (buckets)
Best for	Everyone, IT departments	Long-time automation scripts	Developer CLI tools
Install it	Already there	Install script from chocolatey.org	Install script from scoop.sh

Optional Features and Capabilities

```
# GUI: Settings > System > Optional features
PS C:\> Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
PS C:\> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

PS C:\> Get-WindowsOptionalFeature -Online | Where-Object FeatureName -like '*NetFx3*'
PS C:\> Enable-WindowsOptionalFeature -Online -FeatureName NetFx3   # .NET 3.5

C:\> DISM /Online /Get-Features /Format:Table
```

Lab: A Developer PC in Five Commands

```
PS C:\> winget install --id Microsoft.PowerShell -e
PS C:\> winget install --id Microsoft.VisualStudioCode -e
PS C:\> winget install --id Git.Git -e
PS C:\> winget install --id Python.Python.3.15 -e
PS C:\> winget export -o $HOME\Documents\my-apps.json

# Open a NEW terminal, then check:
PS C:\> git --version ; python --version ; code --version
```

Package IDs change over time — confirm them with winget search.

Summary & Homework

- Prefer winget or the Microsoft Store; always use official sources
- winget: search, install --id -e, upgrade --all, export / import
- Silent MSI: msiexec /i app.msi /qn /l*v log.txt
- Use where.exe and $env:Path to solve 'command not found'
- Homework: export your app list and keep it with your backup
- Homework: run winget upgrade --all once a week

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.