# Shells Overview

A shell is a program that acts as an interface between the user and the operating system kernel. Shells generally fall into two categories:

- Command Line Interface (CLI)
- Graphical User Interface (GUI)

## CMD (Command Prompt)

Platform: Windows (default shell)

Use Cases:

· - Performing basic file operations like copy, move, and delete

· - Running batch scripts (text files with a list of commands)

· - Launching applications

· - Troubleshooting basic system issues

Commands:

· - dir

· - cd

· - del

· - copy

· - move

Power: Low — Limited capabilities, lacks support for piping and complex logic or variable manipulation.

## Bash (Bourne Again Shell)

Platform: Default on Linux and macOS

Use Cases:

· - System automation and scripting on Linux/macOS

· - Writing .sh bash scripts

· - Managing processes, pipelines, and performing advanced text processing

· - Widely used in DevOps and server environments

Commands:

· - Supports redirection

· - piping

· - functions

· - arrays

· - loops

Power: High — Extremely flexible; supports advanced scripting, automation, and environment configuration.

## PowerShell

Platform: Windows (advanced shell), also available on Linux/macOS

Use Cases:

· - System administration and automation on Windows

· - Manipulating structured data using .NET objects

· - Managing services, registry, users, and tasks with powerful object-oriented scripting

Commands:

· - Verb-Noun syntax (e.g., Get-Process, Set-Item)

· - Handles objects

Power: Very High — Designed for automation and complex system tasks; provides deep access to Windows internals and .NET.

## Anaconda Prompt

Platform: Cross-platform (comes with Anaconda distribution)

Use Cases:

· - Managing Python environments and packages using conda

· - Running tools like Jupyter Notebooks, Spyder, etc.

· - Useful in data science and machine learning workflows

Commands:

· - conda commands

· - All standard shell commands

Power: Medium — Especially powerful within the Python data science ecosystem.