**Comparing windows, Mac and linux  
Directory Structure**

**Windows**

* Organizes files in a drive-letter-based hierarchy.
* Uses backslashes (\) to separate folders.
* The root of the system is typically C:\.

**macOS**

* Uses a Unix-style file system with a single root (/), like Linux.
* Separates directories using forward slashes (/).
* Files and folders are located under directories like /Users, /Applications, etc.

**Linux**

* Also based on Unix principles, with a single root /.
* Uses forward slashes (/) for file paths.
* Directories like /home, /etc, and /var hold system and user data.

**Naming Conventions**

**Case Sensitivity**

**Windows**:

* Not case-sensitive by default (file.txt and File.txt are the same).

**macOS**:

* Usually case-insensitive, but case-preserving. However, it can be configured to be case-sensitive.

**Linux**:

* Case-sensitive by default (file.txt and File.txt are different files).

**Naming Restrictions**

**Windows**:

* Cannot use characters like \ / : \* ? " < > |.
* Reserved filenames like CON, PRN, and AUX are not allowed.

**macOS**:

* Cannot use : in file names.

**Linux**:

* Fewer restrictions. Only the / character is not allowed in file names.

**File Systems**

**Windows**:

* Primarily uses NTFS (New Technology File System).
* Also supports FAT32 and exFAT, especially for USB drives.

**macOS**:

* Uses APFS (Apple File System) on newer devices.
* Previously used HFS+.
* Also supports FAT32 and exFAT for compatibility.

**Linux**:

* Uses ext4 as the most common default file system.
* Also supports Btrfs, XFS, and others.
* File Allocation Table (FAT)
* FAT32 is still used today for removable media (USBs, SD cards) because it is supported across all major OSes.
* exFAT is an improvement over FAT32 and is often used on larger external drives for compatibility.

**Executable File Types**

**Windows:**

* .exe – Standard compiled application file.
* .bat – Batch file used for scripting command-line instructions.
* .cmd – Similar to .bat but works better with modern command-line interpreters.
* .msi – Microsoft Installer package for application setup.
* .ps1 – PowerShell script file, used with Windows PowerShell.

**macOS:**

* .app – Application bundle containing all resources and binaries.
* .command – Shell script executable via double-click or Terminal.
* .sh – Standard shell script used in Unix systems.
* Executables can have no extension, especially for command-line tools.

**Linux:**

* Typically has no required extension.
* Common extensions include:
* .sh – Shell script.
* .run – Self-extracting Linux program installer.
* Executability depends on file permissions, not extensions.

**Script Execution and Shell Access**

**Windows:**

* Uses Command Prompt (cmd.exe) and PowerShell for script execution.
* Scripts like .bat, .cmd, and .ps1 are executed using these shells.
* PowerShell provides more powerful scripting capabilities compared to Command Prompt.

**macOS:**

* Uses the Terminal app to access the shell (typically zsh or bash).
* Shell scripts (.sh, .command) are executed with the bash or zsh interpreter.
* AppleScript is another option for automation.

**Linux :**

* Scripts are run through the Terminal, using shells like bash, zsh, or others.
* Shell scripts are executed by giving them execute permissions (chmod +x script.sh) and running ./script.sh.
* Supports a wide variety of scripting languages including Python, Perl, etc.