

PowerShell Basics

Core Commands (Cmdlets)

<code>Get-Command</code>	Lists all available PowerShell commands (cmdlets).
<code>Get-Help <command></code>	Displays help information for a specific command.
<code>Set-ExecutionPolicy</code>	Configures the PowerShell execution policy (e.g., <code>RemoteSigned</code>).
<code>Get-Process</code>	Retrieves information about running processes.
<code>Stop-Process</code>	Stops a running process (e.g., <code>Stop-Process -Id <PID></code>).
<code>Get-Service</code>	Retrieves information about Windows services.
<code>Start-Service</code>	Starts a Windows service.
<code>Stop-Service</code>	Stops a Windows service.
<code>Restart-Service</code>	Restarts a Windows service.

Variables and Operators

<code>\$</code>	Indicates a variable (e.g., <code>\$name = "John"</code>).
<code>=</code>	Assignment operator (e.g., <code>\$x = 10</code>).
<code>-eq</code> , <code>-ne</code> , <code>-gt</code> , <code>-lt</code>	Equality, inequality, greater than, and less than operators.
<code>-and</code> , <code>-or</code> , <code>not</code>	Logical AND, OR, and NOT operators.
<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code>	Arithmetic operators (addition, subtraction, multiplication, division).
<code>+=</code> , <code>-=</code>	Shorthand for incrementing/decrementing the value of a variable.

Basic Syntax & Structures

<code>if (\$condition) { ... } elseif (\$condition) { ... } else { ... }</code>	- Conditional statement.
<code>for (\$i = 0; \$i -lt 10; \$i++) { ... }</code>	- For loop.
<code>foreach (\$item in \$collection) { ... }</code>	- Foreach loop.
<code>while (\$condition) { ... }</code>	- While loop.
<code>function My-Function { ... }</code>	- Function definition.

PowerShell Advanced

Working with Objects

<code>Get-Member</code>	Displays the properties and methods of an object.
<code>Select-Object</code>	Selects specific properties of an object (e.g., <code>Get-Process Select-Object Name, Id</code>).
<code>Where-Object</code>	Filters objects based on a condition (e.g., <code>Get-Process Where-Object {\$_.CPU -gt 1}</code>).
<code>ForEach-Object</code>	Performs an action on each object in a collection (e.g., <code>Get-Process ForEach-Object {Write-Host \$_.Name}</code>).
<code>Export-Csv</code>	Exports objects to a CSV file (e.g., <code>Get-Process Export-Csv -Path processes.csv -NoTypeInformation</code>).
<code>Import-Csv</code>	Imports objects from a CSV file (e.g., <code>\$data = Import-Csv -Path data.csv</code>).

Modules and Script Execution

<code>Import-Module</code>	Imports a PowerShell module (e.g., <code>Import-Module ActiveDirectory</code>).
<code>Get-Module</code>	Lists imported modules.
<code>. <script_path></code>	Executes a PowerShell script in the current scope.
<code>& <script_path></code>	Executes a PowerShell script in a new scope.
<code>. <script_path></code>	Executes a PowerShell script.
<code>Publish-Module</code>	Publishes a module to a repository.

Error Handling

<code>try { ... } catch { ... } finally { ... }</code>	- Try-Catch-Finally block for error handling.
<code>\$ErrorActionPreference = 'Stop'</code>	- Sets the error action preference to stop execution on error.
<code>Write-Error "Error message"</code>	- Writes an error message.

Batch Scripting Basics

Basic Commands

<code>echo</code>	Displays text on the console.
<code>@echo off</code>	Turns command echoing off (usually at the beginning of the script).
<code>pause</code>	Pauses script execution until a key is pressed.
<code>cls</code>	Clears the console screen.
<code>title</code>	Sets the title of the console window.
<code>exit</code>	Exits the script.
<code>rem</code> or <code>::</code>	Comment a line (rem is compatible with older versions).

Batch Scripting Advanced

Conditional Statements and Loops

<code>if <condition> (...) else (...)</code> - Conditional execution. Example: <code>if "ERRORLEVEL" == "0" (</code> <code>echo Success</code> <code>) else (</code> <code>echo Failure</code> <code>)````</code>
<code>for %i in (<set>) do (...)</code> - Looping through a set of items. Example: <code>for %i in (*.txt) do (</code> <code>echo Processing %i</code> <code>)````</code>
<code>goto <label></code> - Jump to a specific label in the script. Example: <code>:start</code> <code>echo Starting...</code> <code>goto end</code> <code>:end</code> <code>echo Ending...</code>

Variables and Operators

<code>set</code> <code><variable>=</code> <code><value></code>	Sets a variable (e.g., <code>set name=John</code>).
<code>%</code> <code><variable>%</code>	Accesses a variable's value (e.g., <code>echo %name%</code>).
<code>==</code> , <code>EQU</code> , <code>NEQ</code> , <code>GTR</code> , <code>LSS</code>	Equality, not equal, greater than, and less than operators (string comparison).
<code>if</code> <code><condition></code> <code>(...) else</code> <code>(...)</code>	Conditional statement.
<code>for %i in</code> <code>(<set>) do (</code> <code>...)</code>	For loop (e.g., <code>for %i in (a b c) do echo %i</code>). Use <code>%i</code> in command line
<code>goto</code> <code><label></code>	Jumps to a labeled section of the script.

Working with Strings

<code>%variable:~start,length%</code>	Substring extraction from a variable (e.g., <code>%string:~0,5%</code>).
<code>set</code> <code>variable=%variable:old=new%</code>	String replacement (e.g., replace 'old' with 'new' in %variable%).
<code>set</code> <code>variable=%variable:</code> <code><search>=%</code>	Delete search string

Calling External Programs

<code>start <program></code> - Starts an external program. Example: <code>start notepad.exe file.txt````</code>
<code>call <batch_file></code> - Calls another batch file. Example: <code>call other_script.bat````</code>

File and Directory Operations

<code>dir</code>	Lists files and directories in the current directory.
<code>cd</code>	Changes the current directory (e.g., <code>cd C:\Windows</code>).
<code>mkdir</code> or <code>md</code>	Creates a new directory (e.g., <code>mkdir NewDirectory</code>).
<code>rmdir</code> or <code>rd</code>	Removes a directory (e.g., <code>rmdir NewDirectory</code>).
<code>copy</code>	Copies files (e.g., <code>copy file.txt C:\Backup</code>).
<code>del</code>	Deletes files (e.g., <code>del file.txt</code>).