PowerShell Tutorial - Beginner to Intermediate

Author: Aditya Saxena

Date: June 2025

====================================================================

SECTION 1: INTRODUCTION TO POWERSHELL

====================================================================

PowerShell is a cross-platform automation shell and scripting language

built on .NET. It is designed for system administration, cloud automation,

and configuration management.

- Works on Windows, Linux, macOS

- Supports object-based pipelines (unlike Bash's text-based)

To start:

- Windows: Open PowerShell or Windows Terminal

- Linux/macOS: Use 'pwsh' from terminal

Script files use the .ps1 extension

====================================================================

SECTION 2: BASIC COMMAND STRUCTURE

====================================================================

PowerShell commands (cmdlets) use Verb-Noun structure:

Examples:

Get-Process # View running processes

Get-Service # List services

Start-Service # Start a service

Stop-Service # Stop a service

Useful meta commands:

Get-Command # Lists all available cmdlets

Get-Help <cmdlet> # Shows usage info

Get-Alias # View command aliases (e.g., ls, dir)

====================================================================

SECTION 3: VARIABLES AND DATA TYPES

====================================================================

Variables begin with $

Examples:

$name = "Aditya"

$age = 30

$list = @(1, 2, 3)

Common types:

[int], [string], [bool], [array], [hashtable]

Casting example:

[int]$x = 42

[string]$s = "Hello"

To display output:

Write-Output $name

Write-Host "Your age is $age"

====================================================================

SECTION 4: CONDITIONAL LOGIC

====================================================================

if ($value -gt 10) {

Write-Output "Greater than 10"

} elseif ($value -eq 10) {

Write-Output "Equals 10"

} else {

Write-Output "Less than 10"

}

Operators:

-eq, -ne, -gt, -lt, -ge, -le

Logical:

-and, -or, -not

====================================================================

SECTION 5: LOOPS

====================================================================

For loop:

for ($i = 0; $i -lt 5; $i++) {

Write-Output $i

}

While loop:

$n = 1

while ($n -le 3) {

Write-Output $n

$n++

}

Foreach loop:

$fruits = @("apple", "banana")

foreach ($fruit in $fruits) {

Write-Output $fruit

}

====================================================================

SECTION 6: FUNCTIONS

====================================================================

function Greet-User {

param([string]$Name)

Write-Output "Hello, $Name!"

}

Greet-User -Name "Aditya"

====================================================================

SECTION 7: FILE OPERATIONS

====================================================================

Get-Content "file.txt" # Read a file

"Hello" | Out-File "file.txt" # Write to file

"Append" | Add-Content "file.txt" # Append to file

Test-Path "file.txt" # Check if file exists

====================================================================

SECTION 8: MODULES AND IMPORTING

====================================================================

Modules = cmdlet packages

Get-Module -ListAvailable # Show installed

Install-Module -Name Az -Force # Install from Gallery

Import-Module Az # Load into session

====================================================================

SECTION 9: REMOTE EXECUTION

====================================================================

Enable-PSRemoting -Force

Invoke-Command -ComputerName SERVER -ScriptBlock { Get-Process }

Enter-PSSession -ComputerName SERVER

Exit-PSSession

====================================================================

SECTION 10: ERROR HANDLING

====================================================================

try {

Get-Item "no-file.txt"

} catch {

Write-Output "Error caught"

}

$ErrorActionPreference = "Stop" # Stop on error

====================================================================

SECTION 11: SCRIPT EXECUTION POLICY

====================================================================

Get-ExecutionPolicy

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

To run script:

.\myscript.ps1

====================================================================

SECTION 12: BEST PRACTICES

====================================================================

- Always validate inputs using param()

- Use Write-Verbose and Write-Debug for logging

- Never hardcode credentials

- Use Get-Credential or managed identity for secrets

- Modularize code into functions

- Test scripts using -WhatIf or -Confirm

====================================================================

SECTION 13: LEARNING RESOURCES

====================================================================

Docs : https://learn.microsoft.com/powershell/

Gallery : https://www.powershellgallery.com

Books : "Learn PowerShell in a Month of Lunches"

AzureShell: https://shell.azure.com

====================================================================

END OF POWERSHELL TUTORIAL

====================================================================