

PowerShell ISE: 12 Handy Script Examples (with Explanations)

Use in the PowerShell ISE or VS Code PowerShell terminal. Save as .ps1 files to reuse.

This one-pager contains 12 short PowerShell examples you can paste into the PowerShell ISE. Each example includes a quick explanation of what it does and why the key parameters matter. Tip: Run ISE as Administrator when doing system-level tasks (services, users).

1) Hello World

Explanation:

Writes a string to the pipeline. In ISE, this shows in the Output pane. Write-Output is preferred over echo/Write-Host when you may want to pipe or capture the result.

[Code](#)

```
Write-Output "Hello, World from PowerShell!"
```

-

2) List Files in a Folder

Explanation:

Get-ChildItem (alias: dir/ls) enumerates files and folders. Format-Table selects which properties to display. \$env:USERNAME pulls your login name from the environment.

[Code](#)

```
Get-ChildItem -Path "C:\Users\$env:USERNAME\Documents" | Format-Table Name, Length, LastWriteTime
```

-

3) Get System Information

Explanation:

Get-ComputerInfo returns a large set of system details. Select-Object narrows to key fields like computer name, OS, and architecture.

[Code](#)

```
Get-ComputerInfo | Select-Object CsName, OsName, OsArchitecture, WindowsVersion
```

-

4) Export Running Processes to CSV

Explanation:

Gets current processes, picks a few properties, and writes them to a CSV file for Excel or analysis. -NoTypeInfo omits the type header row.

[Code](#)

```
Get-Process | Select-Object Name, Id, CPU | Export-Csv -Path "C:\Temp\processes.csv" -NoTypeInfo
```

-

5) Disk Space Report (Drives)

Explanation:

Lists PowerShell drives from the FileSystem provider with used and free bytes. Useful for quick capacity checks across C:, D:, etc.

[Code](#)

```
Get-PSDrive -PSProvider FileSystem | Select-Object Name, Used, Free
```

PowerShell ISE: 12 Handy Script Examples (with Explanations)

Use in the PowerShell ISE or VS Code PowerShell terminal. Save as .ps1 files to reuse.

6) Kill a Process by Name

Explanation:

Forcefully stops all processes that match the name. Use cautiously; consider -Id for targeting a single PID. Requires appropriate privileges.

```
Stop-Process -Name "notepad" -Force
```

Code.

7) Check If a Service Is Running

Explanation:

Queries the Windows Print Spooler service (Spooler). Status shows Running/Stopped; DisplayName is the friendly name.

```
Get-Service -Name "Spooler" | Select-Object Status, DisplayName
```

Code.

8) Restart a Service

Explanation:

Stops and starts the service. Useful after printer driver or queue issues. May require admin rights depending on policy.

```
Restart-Service -Name "Spooler" -Force
```

Code.

9) Simple Backup (Copy Folder)

Explanation:

Copies a directory tree from Documents to D:\Backup. -Recurse walks subfolders; -Force overwrites existing files if needed.

```
$source = "C:\Users\$env:USERNAME\Documents"
```

Code.

\$des

10) Create a New Local User

Explanation:

Creates a local account on Windows 10/11. The password is converted to a SecureString at runtime. Run in an elevated session (Run as Administrator).

```
New-LocalUser -Name "TestUser" -Password (ConvertTo-SecureString "P@ssword123"
-AsPlainText -Force) -FullName "Test User" -Description "Temporary account"
```

Code.

PowerShell ISE: 12 Handy Script Examples (with Explanations)

Use in the PowerShell ISE or VS Code PowerShell terminal. Save as .ps1 files to reuse.

11) Ping Multiple Hosts

Explanation:

Loops over a list of hostnames and sends two ICMP pings to each. Select-Object keeps only the useful columns. Test-NetConnection is another modern option.

```
$computers = @("google.com", "github.com", "microsoft.com")
```

Code.

forea

-

12) Simple GUI Popup

Explanation:

Loads WPF types and shows a basic message box with a title. Handy for quick notifications or confirming script steps.

```
Add-Type -AssemblyName PresentationFramework
```

Code.

[Sys

Examp

-