



PowerShell Genie Complete Cheat Sheet

Commands, pipeline patterns, object thinking, visual sketches, and task references in one PDF.

1	Discover before guessing Use Get-Command and Get-Help to inspect the command that is actually available in your session.
2	Keep objects structured Filter, sort, select, measure, and export object properties. Do not parse formatted console text when object properties exist.
3	Filter before formatting Where-Object chooses objects. Select-Object chooses properties or limits rows. Format-* is for final display.
4	Sort before top-N If the question asks for the largest, smallest, busiest, or newest objects, sort first and then select the first rows.

Read this before running examples

Examples in this reference are learning patterns. Results depend on your operating system, installed modules, permissions, and local data. Get-Service is Windows-specific. CPU on Get-Process is accumulated processor time in seconds, not live CPU percentage. WorkingSet64 is memory in bytes. Validate commands in a safe environment before using them in production.

Core mental model

Verb-Noun describes action and target. The pipeline passes .NET objects. Each object keeps named properties such as Name, Status, CPU, WorkingSet64, Id, and StartType until you deliberately select or format them.

Essential PowerShell command patterns

Use these as starting patterns. Read the note in each heading, then adapt properties, names, paths, and limits to your own environment.

Discover Commands

```
Get-Command -Noun Process
Get-Help Get-Process -Examples
```

Inspect Objects - Console Columns Are Only A View

```
Get-Process | Get-Member
Get-Process | Select-Object -First 3 -Property Name, Id, CPU
```

Filter - Keep Objects Whose Properties Match

```
Get-Service | Where-Object { $_.Status -eq 'Running' }
```

Sort, Then Limit - Cpu Here Is Cumulative Seconds, Not Cpu Percentage

```
Get-Process | Sort-Object CPU -Descending | Select-Object -First 3 Name, CPU
```

Measure - Working-Set Memory In Bytes

```
Get-Process | Measure-Object -Property WorkingSet64 -Sum
```

Group - Count Services In Each Status (Windows)

```
Get-Service | Group-Object Status | Select-Object Name, Count
```

Files - Inspect The Current Directory

```
Get-ChildItem -File | Select-Object Name, Length, LastWriteTime
```

Strings - Double Quotes Expand Variables; Single Quotes Preserve Literal Text

```
$topic = 'PowerShell'
"Learning $topic"
'Learning $topic'
```

Export Objects - Do Not Put Format-Table Before Export-Csv

```
Get-Process | Select-Object Name, Id, CPU | Export-Csv -LiteralPath ./processes.csv -NoTypeInformation
```

Handle Errors - Catch Sees Terminating Errors

```
try {
    Get-Item -LiteralPath './missing-file.txt' -ErrorAction Stop
} catch {
    $_.Exception.Message
}
```

Remember

- Verb-Noun: action + target.
- \$_: the current object inside a pipeline script block.
- Get-Member: inspect properties and methods.
- Where-Object: choose rows. Select-Object: choose properties or limit rows.
- Sort before selecting a top-N result.
- Format only for final display; keep objects for export or further processing.

Practise With Sample Data

<https://powershellgenie.com/mission-lab/>

<https://powershellgenie.com/powershell-practice/>

Task-to-command reference

Use the task column when you know what you want to do but not the command name. The online reference has syntax, examples, related commands, and module notes.

Task	Command	Reference
Discover syntax	Get-Help	powershellgenie.com/powershell-commands/get-help/
Find commands	Get-Command	powershellgenie.com/powershell-commands/get-command/
Inspect objects	Get-Member	powershellgenie.com/powershell-commands/get-member/
List files	Get-ChildItem	powershellgenie.com/powershell-commands/get-childitem/
Copy files	Copy-Item	powershellgenie.com/powershell-commands/copy-item/
Verify file integrity	Get-FileHash	powershellgenie.com/powershell-commands/get-filehash/
Filter objects	Where-Object	powershellgenie.com/powershell-commands/where-object/
Sort results	Sort-Object	powershellgenie.com/powershell-commands/sort-object/
Choose properties	Select-Object	powershellgenie.com/powershell-commands/select-object/
Export a report	Export-Csv	powershellgenie.com/powershell-commands/export-csv/
Inspect services	Get-Service	powershellgenie.com/powershell-commands/get-service/
Inspect processes	Get-Process	powershellgenie.com/powershell-commands/get-process/
Query Windows inventory	Get-CimInstance	powershellgenie.com/powershell-commands/get-ciminstance/
Query AD users	Get-ADUser	powershellgenie.com/powershell-commands/get-aduser/
Read event logs	Get-WinEvent	powershellgenie.com/powershell-commands/get-winevent/
Inspect job output	Receive-Job	powershellgenie.com/powershell-commands/receive-job/
Preview a reboot	Restart-Computer	powershellgenie.com/powershell-commands/restart-computer/

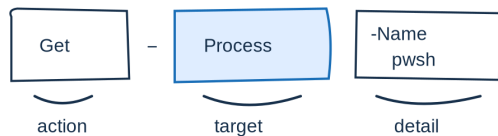
Six visual pipeline concepts

These sketches compress the most important object and pipeline ideas into one printable page.

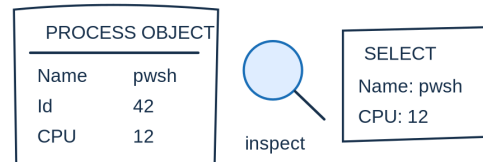
PowerShell: see what the pipeline does

PowerShell Genie · Six concepts to keep beside your commands

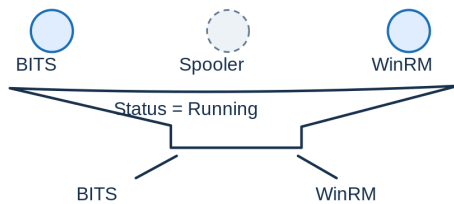
1. Read a command like a sentence



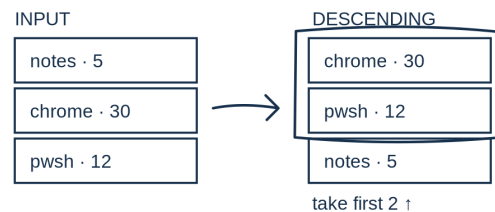
2. A row is a view, not the object



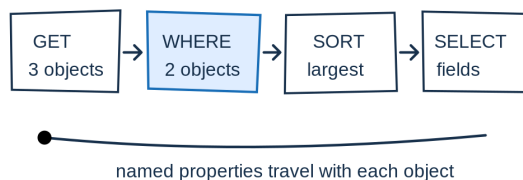
3. Where-Object is a sieve



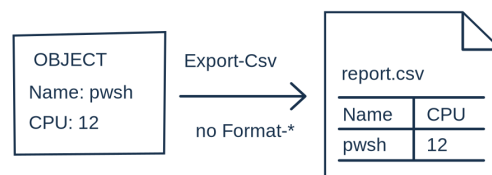
4. Sort before taking the top rows



5. Pass objects along the pipeline



6. Export objects, not a drawn table



Step through each sketch: powershellgenie.com/powershell-cheat-sheet/ · Sample data only

PowerShell memory rules

- Inspect first. Use `Get-Member` when you do not know which properties an object exposes.
- Filter with properties. `Where-Object` should normally compare real properties such as `Status`, `Name`, `CPU`, or `WorkingSet64`.
- Select deliberately. `Select-Object` can choose properties and can limit the number of objects. These are different jobs.
- Sort before selecting top rows. Top-N questions require an ordering before `-First` is meaningful.
- Export objects. `Export-Csv` should receive structured objects, not `Format-Table` output.
- Know units. `WorkingSet64` is bytes; `CPU` is accumulated seconds. Convert units explicitly when presenting results.
- Know scope. Commands can depend on Windows, modules, tenant context, permissions, and PowerShell edition.
- Use safe discovery. `Get-Help -Examples` and `Get-Command -Syntax` are cheap ways to confirm intent before executing a changing command.

Practice next

Use the browser exercises at powershellgenie.com/powershell-practice/ and the guided mission lab at powershellgenie.com/mission-lab/. The exercises use fictional fixture data and do not make changes to your machine.