PowerShell

Fundamentals and Command Discovery

This PowerShell lab explores the fundamentals of using PowerShell Core on macOS. It begins with version checks and environment discovery, then introduces tools to explore the PowerShell help system, command structure, and process management. You will learn how to inspect available commands, retrieve syntax, filter help content, and analyze object types returned by cmdlets such as Get-Process and Get-FileHash. The lab highlights object pipelines and shows how to refine output with Select-Object and filter with Where-Object.

1. Start PowerShell Core on macOS

Launch PowerShell using `pwsh` to enter the cross-platform shell environment.

```
PowerShell@MAC1 Powershell % pwsh
──────────────────────────────────────────────────────────────────
PowerShell 7.5.1
──────────────────────────────────────────────────────────────────
```

2. View Complete PowerShell Environment Metadata

Use `$PSVersionTable` to see version, OS, platform, and compatibility information.

```
PowerShell> $PSVersionTable
──────────────────────────────────────────────────────────────────

Name                           Value
----                           -----
PSVersion                      7.5.1
PSEdition                      Core
GitCommitId                    7.5.1
OS                             Darwin 24.5.0 Darwin Kernel Version 24.5.0: Tue Apr 22 19:48:46 PDT 2025;
root:xnu-11417.121.6~2/RELEASE_ARM64_T8103
Platform                       Unix
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0…}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0
──────────────────────────────────────────────────────────────────
```

3. Check PowerShell Version Number

Display only the `PSVersion` property to confirm the major, minor, and patch version.

```
PowerShell> $PSVersionTable.PSVersion
──────────────────────────────────────────────────────────────────

Major  Minor  Patch  PreReleaseLabel BuildLabel
-----  -----  -----  --------------- ----------
7      5      1

──────────────────────────────────────────────────────────────────
```

4. Retrieve Operating System Details

Inspect the specific operating system PowerShell is running on using `$PSVersionTable.OS`.

```
PowerShell> $PSVersionTable.OS
──────────────────────────────────────────────────────────────────────
Darwin 24.5.0 Darwin Kernel Version 24.5.0: Tue Apr 22 19:48:46 PDT 2025;
root:xnu-11417.121.6~2/RELEASE_ARM64_T8103
PowerShell>
──────────────────────────────────────────────────────────────────────
```

5. Display Git Commit ID of Build

Access the `GitCommitId` property to identify the exact build used for PowerShell Core.

```
$PSVersionTable.GitCommitId
──────────────────────────────────────────────────────────────────────
7.5.1
──────────────────────────────────────────────────────────────────────
```

6. Discover Standard PowerShell Verbs

Run `Get-Verb` to list the standardized verbs used in cmdlet naming conventions.

```
PowerShell> Get-Verb
──────────────────────────────────────────────────────────────────────

Verb       AliasPrefix Group        Description
----       ----------- -----        -----------
Add        a           Common       Adds a resource to a container, or attaches an item to another
item
Clear      cl          Common       Removes all the resources from a container but does not delete
the container
Close      cs          Common       Changes the state of a resource to make it inaccessible,
unavailable, or unusable
Copy       cp          Common       Copies a resource to another name or to another container
──────────────────────────────────────────────────────────────────────
```

7. List All Available Commands

Explore current session commands using `Get-Command`, including aliases and functions.

```
PowerShell> Get-Command
```

```
CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           Get-PSResource                                     1.1.1
Microsoft.PowerShell.PSResourceGet
Function        cd..
Function        cd\
Function        cd~
```

8. Access Help System Overview

Use `help` to see the basic structure and function of PowerShell's help system.

```
PowerShell> help
```

```
TOPIC
    PowerShell Help System

SHORT DESCRIPTION
    Displays help about PowerShell cmdlets and concepts.

LONG DESCRIPTION
    PowerShell Help describes PowerShell cmdlets, functions, scripts, and
    modules, and explains concepts, including the elements of the PowerShell
    language.

...
```

9. Invoke Full Help System

Run `Get-Help` to fetch detailed cmdlet documentation and conceptual guidance.

```
TOPIC
    PowerShell Help System

SHORT DESCRIPTION
    Displays help about PowerShell cmdlets and concepts.

LONG DESCRIPTION
    PowerShell Help describes PowerShell cmdlets, functions, scripts, and
    modules, and explains concepts, including the elements of the PowerShell
    language.

...
```

10. Analyze Help Output Object with Get-Member

Pipe `help` to `Get-Member` to identify its object type (`System.String`) and methods.

```
PowerShell> help | Get-Member
————————————————————————————————————————————————————————————
   TypeName: System.String

Name             MemberType          Definition
----             ----------          ----------
Clone            Method              System.Object Clone(), System.Object ICloneable.Clone()
CompareTo        Method              int CompareTo(System.Object value), int CompareTo(string
strB), int IComparable.CompareTo(System.Object o…
Contains         Method              bool Contains(string value), bool Contains(string value,
System.StringComparison comparisonType), bool Co…

…
————————————————————————————————————————————————————————————
```

11. Find Alias Management Cmdlets

Use `Get-Command -Noun alias*` to discover commands for creating and managing aliases.

```
PowerShell> Get-Command -Noun alias*
————————————————————————————————————————————————————————————
CommandType     Name                                Version    Source
-----------     ----                                -------    ------
Cmdlet          Export-Alias                        7.0.0.0    Microsoft.PowerShell.Utility
Cmdlet          Get-Alias                           7.0.0.0    Microsoft.PowerShell.Utility
Cmdlet          Import-Alias                        7.0.0.0    Microsoft.PowerShell.Utility
Cmdlet          New-Alias                           7.0.0.0    Microsoft.PowerShell.Utility
Cmdlet          Remove-Alias                        7.0.0.0    Microsoft.PowerShell.Utility
Cmdlet          Set-Alias                           7.0.0.0    Microsoft.PowerShell.Utility
————————————————————————————————————————————————————————————
```

12. Retrieve Specific Cmdlet Using Verb-Noun Syntax

Apply `Get-Command -Verb Get -Noun alias` to directly locate the `Get-Alias` cmdlet.

```
PowerShell> Get-Command -Verb Get -Noun alias
————————————————————————————————————————————————————————————
CommandType     Name                                Version    Source
-----------     ----                                -------    ------
Cmdlet          Get-Alias                           7.0.0.0    Microsoft.PowerShell.Utility
```

```
--------------------------------------------------------------------------------
```

13. View Help Syntax for Get-Help Cmdlet

Use `Get-Help -Name Get-Help` to view syntax, parameters, and usage of the help system itse

```
PowerShell> Get-Help -Name Get-Help
--------------------------------------------------------------------------------
NAME
    Get-Help

SYNTAX
    Get-Help [[-Name] <string>] [-Path <string>] [-Category {Alias | Cmdlet | Provider | General |
FAQ | Glossary | HelpFile | ScriptCommand |
    Function | Filter | ExternalScript | All | DefaultHelp | DscResource | Class | Configuration}] [-Full]
[-Component <string[]>] [-Functionality
    <string[]>] [-Role <string[]>] [<CommonParameters>]
--------------------------------------------------------------------------------
```

14. Update Help Files (Intro)

Understand that newer PowerShell versions don’t include help files by default. Use `Update-Help` to download them.

```
PowerShell> Update-Help

New versions of PowerShell don't include the help system by default. The first time you run Get-
Help, you're asked to install the help files. You can also run the Update-Help cmdlet to install the
help files. Because a call to Update-Help downloads many help files, the command can fetch only
once per day by default. You can override this fetching behavior by using the -Force flag.
```

15. Update Help with Language and Verbose Output

Execute `Update-Help -UICulture en-US -Verbose` to update help for English content and view module-specific messages.

```
PowerShell> Update-Help -UICulture en-US -Verbose
--------------------------------------------------------------------------------
VERBOSE: Help was not updated for the module Microsoft.PowerShell.Management, because the
Update-Help command was run on this computer within the last 24 hours.
To update help again, add the Force parameter to your command.
VERBOSE: Help was not updated for the module Microsoft.PowerShell.Utility, because the
Update-Help command was run on this computer within the last 24 hours.
--------------------------------------------------------------------------------
```

16. Display Command Examples Only

Use `Get-Help Get-FileHash -Examples` to retrieve usage examples without loading the full help page.

```
PowerShell> Get-Help Get-FileHash -Examples
──────────────────────────────────────────────────────────────────────────────
NAME
    Get-FileHash

SYNOPSIS
    Computes the hash value for a file by using a specified hash algorithm.


    --------- Example 1: Compute the hash value for a file ---------

    Get-FileHash /etc/apt/sources.list | Format-List

    Algorithm : SHA256
    Hash      : 3CBCFDDEC145E3382D592266BE193E5BE53443138EE6AB6CA09FF20DF609E268
    Path      : /etc/apt/sources.list

...
──────────────────────────────────────────────────────────────────────────────
```

If you don't want to display the full help page, narrow the response by adding flags to your Get-Help command. Here are some flags you can use:

- **Full**: Returns a detailed help page. It specifies information like parameters, inputs, and outputs that you don't get in the standard response.
- **Detailed**: Returns a response that looks like the standard response, but it includes a section for parameters.
- **Examples**: Returns only examples, if any exist.
- **Online**: Opens a web page for your command.
- **Parameter**: Requires a parameter name as an argument. It lists a specific parameter's properties.

```
PowerShell> Get-Help Get-FileHash -Detailed
──────────────────────────────────────────────────────────────────────────────
NAME
    Get-FileHash

SYNOPSIS
    Computes the hash value for a file by using a specified hash algorithm.


SYNTAX
    Get-FileHash [-InputStream] <System.IO.Stream> [[-Algorithm] {SHA1 | SHA256 | SHA384 |
SHA512 | MD5}] [<CommonParameters>]

    Get-FileHash [-LiteralPath] <System.String[]> [[-Algorithm] {SHA1 | SHA256 | SHA384 | SHA512
| MD5}] [<CommonParameters>]

    Get-FileHash [-Path] <System.String[]> [[-Algorithm] {SHA1 | SHA256 | SHA384 | SHA512 |
MD5}] [<CommonParameters>]


DESCRIPTION
```

The `Get-FileHash` cmdlet computes the hash value for a file by using a specified hash algorithm. A hash value is a unique value that
corresponds to the content of the file. Rather than identifying the contents of a file by its file name, extension, or other designation, a hash
assigns a unique value to the contents of a file. File names and extensions can be changed without altering the content of the file, and without
changing the hash value. Similarly, the file's content can be changed without changing the name or extension. However, changing even a single
character in the contents of a file changes the hash value of the file.

...
——

17. Show Help in Friendly Default Format

Run `help Get-FileHash` to see a readable default view of the command's name, syntax, and description.

PowerShell> help Get-FileHash
——
NAME
Get-FileHash

SYNOPSIS
Computes the hash value for a file by using a specified hash algorithm.

SYNTAX
Get-FileHash [-InputStream] <System.IO.Stream> [[-Algorithm] {SHA1 | SHA256 | SHA384 | SHA512 | MD5}] [<CommonParameters>]

Get-FileHash [-LiteralPath] <System.String[]> [[-Algorithm] {SHA1 | SHA256 | SHA384 | SHA512 | MD5}] [<CommonParameters>]

Get-FileHash [-Path] <System.String[]> [[-Algorithm] {SHA1 | SHA256 | SHA384 | SHA512 | MD5}] [<CommonParameters>]

DESCRIPTION
...
——

18. Re-Check Only Examples for Hash Command

Use `help Get-FileHash -Examples` again for a focused, clean output of real use cases.

PowerShell> help Get-FileHash -Examples
——
NAME
Get-FileHash

SYNOPSIS
Computes the hash value for a file by using a specified hash algorithm.

--------- Example 1: Compute the hash value for a file ---------

Get-FileHash /etc/apt/sources.list | Format-List

Algorithm : SHA256
Hash : 3CBCFDDEC145E3382D592266BE193E5BE53443138EE6AB6CA09FF20DF609E268
Path : /etc/apt/sources.list

…
——

19. List All Running Processes

Run `Get-Process` to list all active processes on the system along with memory and CPU metrics.

```
PowerShell> Get-Process
————————————————————————————————————————————————
 NPM(K)    PM(M)      WS(M)     CPU(s)      Id  SI ProcessName
 ------    -----      -----     ------      --  -- -----------
…
     0     0.00      18.61       9.34    4631 …31 Ad Block One Dashboard
…
————————————————————————————————————————————————
```

20. Filter for a Specific Process by Name

Use `Get-Process -Name "Ad Block One Dashboard"` to isolate a specific running application.

```
PowerShell> Get-Process -Name "Ad Block One Dashboard"
————————————————————————————————————————————————
 NPM(K)    PM(M)      WS(M)     CPU(s)      Id  SI ProcessName
 ------    -----      -----     ------      --  -- -----------
     0     0.00      20.11       9.43    4631 …31 Ad Block One Dashboard
————————————————————————————————————————————————
```

21. Inspect the Returned Object with Get-Member

Pipe the process object into `Get-Member` to reveal its type (`System.Diagnostics.Process`) and its members.

```
PowerShell> Get-Process -Name "Ad Block One Dashboard" | Get-Member
————————————————————————————————————————————————————————————
   TypeName: System.Diagnostics.Process

Name                 MemberType     Definition
----                 ----------     ----------
Handles              AliasProperty  Handles = Handlecount
Name                 AliasProperty  Name = ProcessName
————————————————————————————————————————————————————————————
```

The first line of the response, running the Get-Member command, is the type of the returned object. When you know the type, you can search for other cmdlets that operate on the same type. Explore these related commands to quickly build your knowledge in the domain you're working in.

The first row indicates that the type is System.Diagnostics.Process. Use this type as a search argument to look for other cmdlets that use this type.

22. Discover Cmdlets by Parameter Type

Use `Get-Command -ParameterType Process` to find cmdlets that accept `Process` objects.

```
PowerShell> Get-Command -ParameterType Process
————————————————————————————————————————————————————————————
CommandType     Name                                Version    Source
-----------     ----                                -------    ------
Cmdlet          Debug-Process                       7.0.0.0    Microsoft.PowerShell.Management
————————————————————————————————————————————————————————————
```

23. Refine Object Output with Select and Where

Use `Select-Object` to show specific object properties, and `Where-Object` to filter for methods only

```
PowerShell> Get-Process -Name "Ad Block One Dashboard" | Get-Member
————————————————————————————————————————————————————————————
   TypeName: System.Diagnostics.Process

Name                 MemberType     Definition
----                 ----------     ----------
Handles              AliasProperty  Handles = Handlecount
Name                 AliasProperty  Name = ProcessName
NPM                  AliasProperty  NPM = NonpagedSystemMemorySize64
————————————————————————————————————————————————————————————
```

```
PowerShell> Get-Process -Name "Ad Block One Dashboard" | Get-Member | Select-Object
Name, MemberType
```

Name	MemberType
Handles	AliasProperty
Name	AliasProperty

By introducing the Select-Object cmdlet, you can choose which columns appear in the response. The command expects either a comma-separated list of column names or a wildcard character, such as an asterisk (*), which indicates all columns.

```
PowerShell> Get-Process -Name "Ad Block One Dashboard" | Get-Member | Where-Object
MemberType -eq 'Method'
```

TypeName: System.Diagnostics.Process

Name	MemberType	Definition
BeginErrorReadLine	Method	void BeginErrorReadLine()
BeginOutputReadLine	Method	void BeginOutputReadLine()