What is Powershell and How to Use it

A command-line shell, a scripting language, and a configuration management system combine to make PowerShell, a cross-platform job automation tool. Windows, Linux, and macOS all support PowerShell.

To assist IT workers in configuring systems and automating administrative activities, Microsoft created an object-oriented automation engine, scripting language, and interactive command-line shell.

Powershell commands and some Linux commands are comparable.

Why use PowerShell?

The most attractive reason to use any type of CLI is the ability for precise and repeatable control over a desired action or workflow that would be difficult or impossible to do with a traditional GUI.

Important PowerShell Features

Microsoft introduces updates and new features with each PowerShell release, but the following is a list of key features and capabilities.

Pipelining: With PowerShell, commands can be combined using the pipe operator, denoted by | er This method allows the output of a command to become the input of the next command in the pipeline. A PowerShell pipe allows objects to flow from one cmdlet to another cmdlet instead of text strings. This powerful feature is important for complex and detailed automation scripts.

Help Capabilities: Users can learn more about basic PowerShell and advanced features such as cmdlets via the Get-Help cmdlet. The -online parameter provides access to online help documentation, if any, for a topic.

Remote commands: Administrators can perform tasks remotely on one or more computers using technologies such as Windows Management Instrumentation and WS-Management. For example, the WS-Management protocol allows users to run PowerShell commands and scripts on remote computers.

Discoverability: Users can explore PowerShell capabilities using cmdlets, such as Get-Command, which lists all commands—including cmdlets and functions—available on a computer. Parameters can be used to limit the search scope.

PowerShell Functions:

1. Cmdlets

Cmdlets are command-line, single-function PowerShell commands. If PowerShell is a module, cmdlets are the letters of each word in the module. They can be used individually to perform a task and combined to perform larger tasks. Note that each cmdlet still works as a separate function to contribute the cmdlet output to the combined function.

However, cmdlets are not written in PowerShell. Written in another language, compiled and available in PowerShell. Cmdlets are important commands in PowerShell because the limits of their functionality depend on the creativity of the developer. Developers and DevOps engineers can use "pipes" to send the output of one cmdlet to the input of another cmdlet as an object.

2. PowerShell Functions

Functions are one of the many applications for running code in PowerShell. Unlike cmdlets, functions are written in PowerShell. A series of instructions will be created and received over the phone. Input is that parameter, but the output can be displayed on the user's screen or passed to another function or cmdlet entry.

There are two functions in Powershell: basic and advanced. The base function is the simplest type of function used in PowerShell. With original works, there is no inheritance. The body of the function is simply a set of successive parentheses. Advanced jobs are jobs that follow the characteristics of the original job as a core, but with other jobs. These features are built-in and do more.

3. PowerShell scripts

PowerShell scripts are written in cmdlets. These scripts are used to create automation for various tasks. There are three types of commands in a PowerShell script. The "get" command is the first command to retrieve data from the file system. The "set" command is used to edit Windows component information. It involves assigning items to different categories. The "delete" command is used to completely cancel the job. PowerShell scripts reduce code complexity when writing code and other automation applications.

4. Executable Applications

Executable applications are applications that manage executable files. Executive files with .exe extension are part of Microsoft Windows software. There are three commands used to run exe files. The first is with the "Invoke-expression" command. This is undoubtedly the most popular way to manage these files.

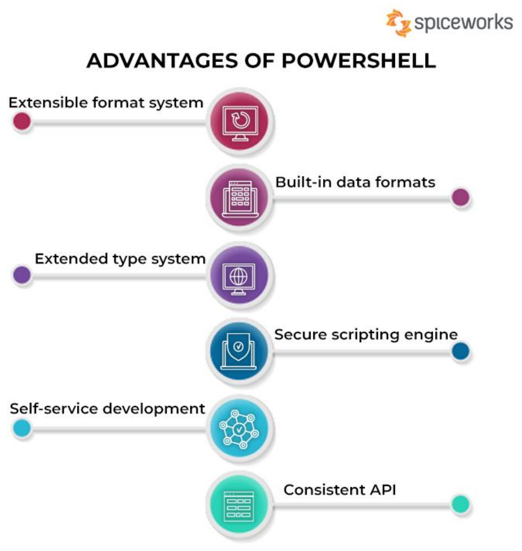
The second command is the "process-start" cmdlet. This application will start many tasks on your machine, but the results are the same as the first command. The third option is to type "." before the file name. This is the easiest way, but all the options are, run the exe file.

Advantages:

1. Extended format system

Using PowerShell, it is easy for the user to format the input and output as he wants. PowerShell has three formatting methods. Each method can be manipulated for the settings required by the user.

The first formatting method is "long form". In this method, the user can display a function in an object. This can be used to fill columns in a table or create a list. Another method is "Format list". This method allows users to display the properties of the items to be listed with each item on a new line. On the third side of the die is the "pattern table". This method makes it easy for users to display the result in tabular form. There are parameters like "Autosize", "wrap" and "groupby" to customize the table columns.



2. Built-in data formats

Data formats support PowerShell, that one can use to store and transfer data and make data human-readable and machine-readable. These data formats are comma-separated value (CSV), JavaScript Object Notation (JSON), and Extensible Markup Language (XML). CSV format is an inbuilt data format that stores table data as plain text. In this format, each file is separated from another with a comma.

The JSON format is used for immediate communication between a browser and a client. It is an open standard format and also a human-readable format. If information is given from the browser, the information can be converted into PowerShell data using “Invoke-WebRequest” or “Invoke-RestMethod” commands. The XML format is also machine and human-readable. It is used to obtain data from a webpage, edit it and post it back. It is also used for configuring application performance in PowerShell.

3. Extended type system

There is an extended type system (ETS) in PowerShell that script and cmdlet developers can use to manipulate .NET objects. This system is done using the PSObject object. One can use the PSObject object to extend object types in two ways. In the first method, the PSObject object shows different views of specific object types. This is called an adapted view.

In the second method, the PSObject object provides means of adding members to an existing object. These new members extend the base object (the initial object that is worked on) by giving it additional information that can be useful in scripting.

4. Secure scripting engine

PowerShell places the security in the hands of the user with the configuration management feature. This feature allows users to decide which scripts to run through visually review. Given that malicious scripts are hardly noticed visually, there are automated security policies to help users.

PowerShell uses an execution policy as one of its security strategies. This execution policy determines the conditions under which PowerShell will execute specific configuration files and scripts. This policy is used on the Windows platform but can be set across various computers and devices with the group policy setting.

However, this policy is only active when the user decides. When made active, PowerShell runs only scripts authorized by an identifiable name. This is a fair choice against running malicious scripts on your computer that may allow cyber threats to enter.

5. Self-service development

PowerShell enables each team in an Exchange to build their cmdlets by themselves. This is a significant advantage over other options. This "self-service" model is crucial as it enables developers to build the management of their features even as they write their features.

This results in better management and, in turn, quality products. This is because products are most beneficial when component teams are "in charge" of their features. This ensures the users of a well-structured feature. It also makes the building faster as one can test the feature with real codes early. This eliminates bottlenecks in the building of features and scripts.

6. Consistent API

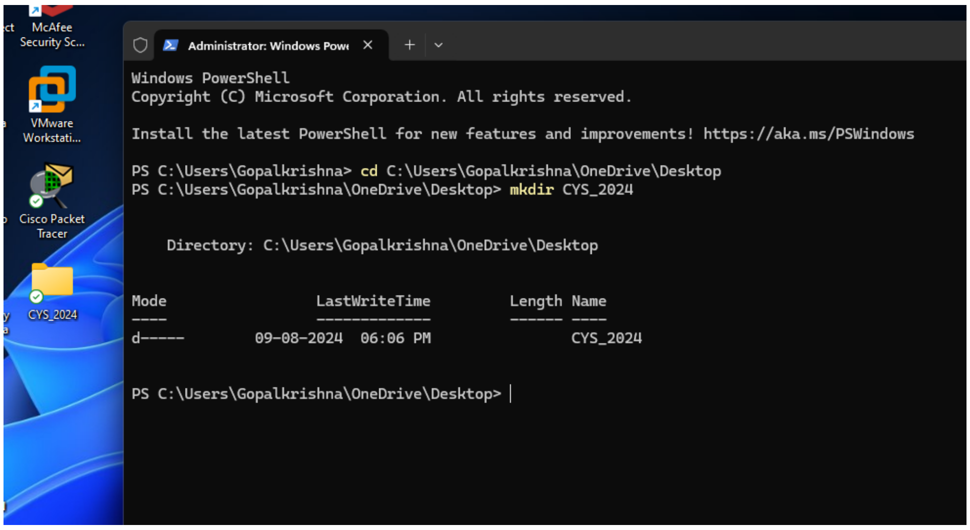
APIs like Windows Management Instrumentation (WMI) and Component Object Model (COM) used on Windows is known to be inconsistent or incomplete. This makes them inefficient in carrying requests to and from the browser.

However, PowerShell is a 100% comprehensive and consistent API that 3rd parties use. This is the REST API from the power shell. It is activated using the "Invoked-RestMethod" cmdlet. This activation is simply a request through HTTPS or HTTP. The API would be sent to get data through a URL.

How to Write a Powershell Script:

Step 1: Set the Path where your Folder needs to be stored.

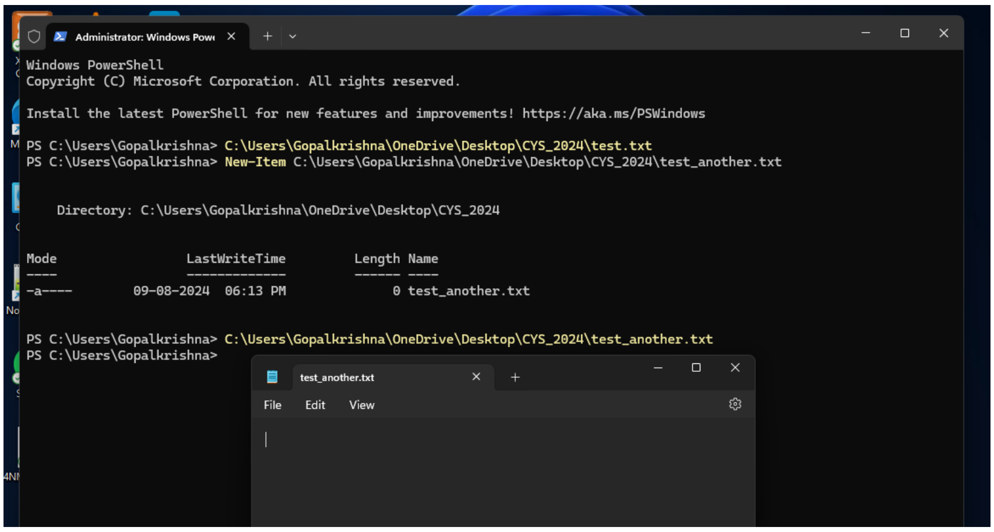
To go to the Directory use **`cd path_name`** and to create the directory use **`mkdir`** command;



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Gopalkrishna> cd C:\Users\Gopalkrishna\OneDrive\Desktop
PS C:\Users\Gopalkrishna\OneDrive\Desktop> mkdir CYS_2024


    Directory: C:\Users\Gopalkrishna\OneDrive\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        09-08-2024   06:06 PM                CYS_2024


PS C:\Users\Gopalkrishna\OneDrive\Desktop>
```

Step2: To create a text file in the location use **`New-Item`.**

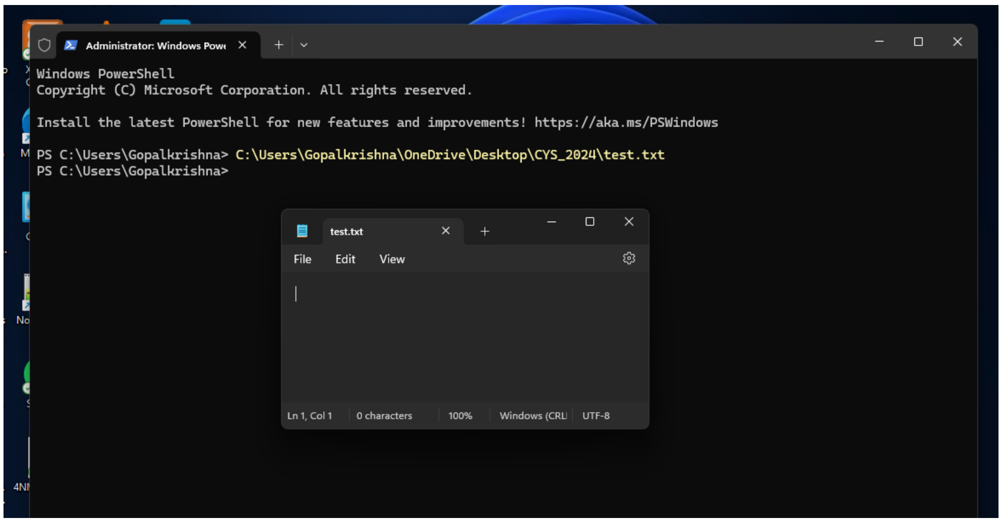
To create a file use **New-Item recent_path_name\filename.txt**



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Gopalkrishna> C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test.txt
PS C:\Users\Gopalkrishna> New-Item C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt


    Directory: C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        09-08-2024   06:13 PM              0 test_another.txt


PS C:\Users\Gopalkrishna> C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt
PS C:\Users\Gopalkrishna>
```

Step 3: To Open a text File

Use “ **recent_path_name\filename.txt** “ to open the text file to view.

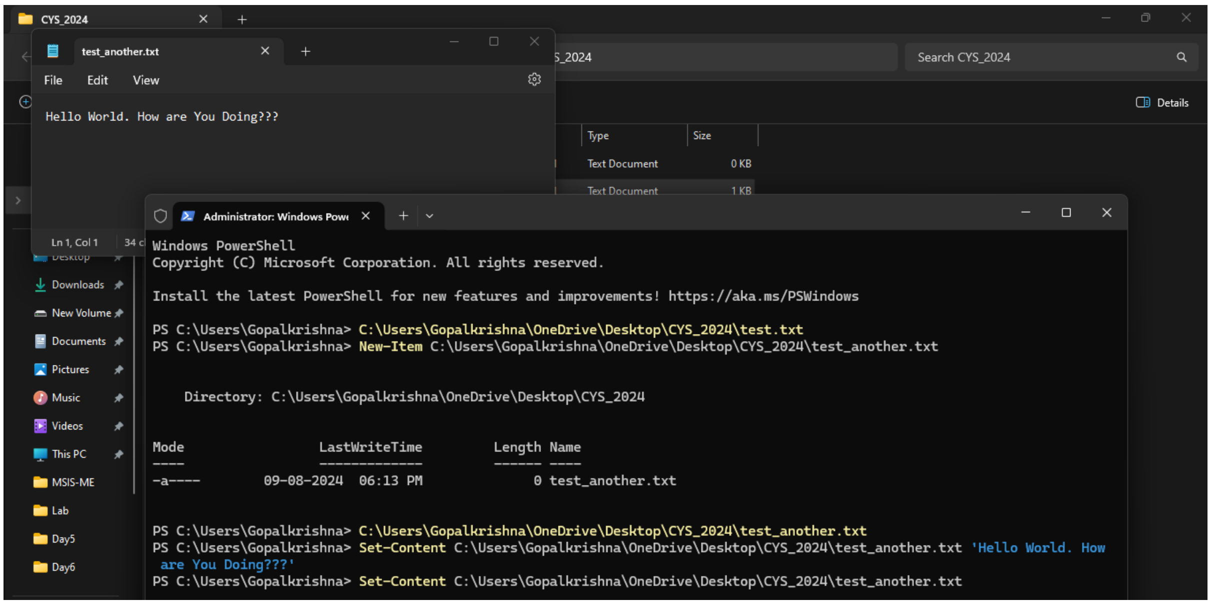


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Gopalkrishna> C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test.txt
PS C:\Users\Gopalkrishna>
```

Step 4: To write in the file

To write on the File use **Set-Content.**

Set-Content recent_path_name\filename.txt



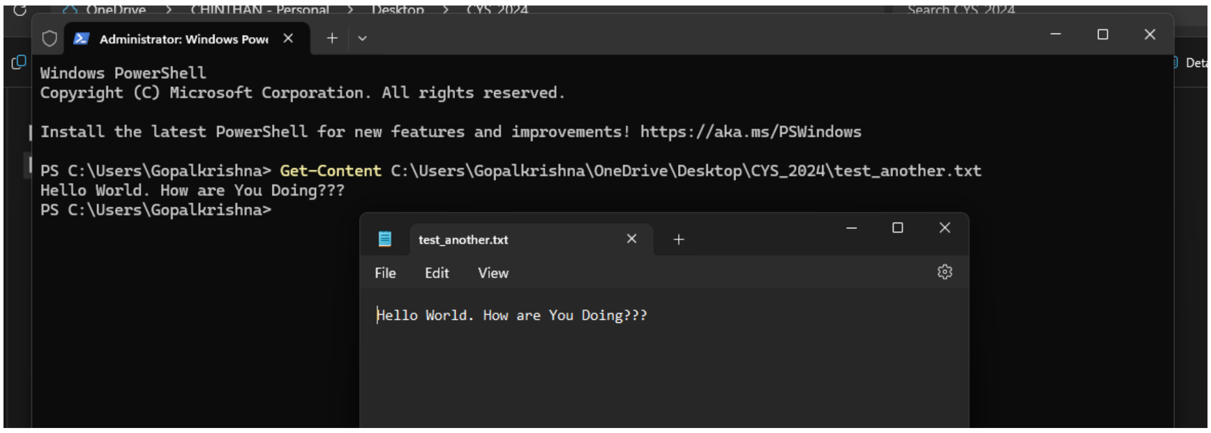
```
Hello World. How are You Doing???
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Gopalkrishna> C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test.txt
PS C:\Users\Gopalkrishna> New-Item C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt


    Directory: C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        09-08-2024   06:13 PM              0 test_another.txt


PS C:\Users\Gopalkrishna> C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt
PS C:\Users\Gopalkrishna> Set-Content C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt 'Hello World. How
 are You Doing???'
PS C:\Users\Gopalkrishna> Set-Content C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt
```

Step 5: To read the file

To read the File use **Get-Content.**

Get-Content recent_path_name\filename.txt



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Gopalkrishna> Get-Content C:\Users\Gopalkrishna\OneDrive\Desktop\CYS_2024\test_another.txt
Hello World. How are You Doing???
PS C:\Users\Gopalkrishna>
```

```
Hello World. How are You Doing???
```

Steps for excecuting Python Program in Powershell:

One way of creating python program using powershell.

1. cd path_name   (To go to specified path)

2. echo 'print("Hello, World!")' > hello.py (print “Hello World” in hello.py)

3. ls   (To verify the file is present or not)

4. python hello.py   (Run the Script)

There are numerous number of commands and scripting techniques to explore and here just the basic is being documented.

For more, refer --   https://learn.microsoft.com/en-us/powershell/