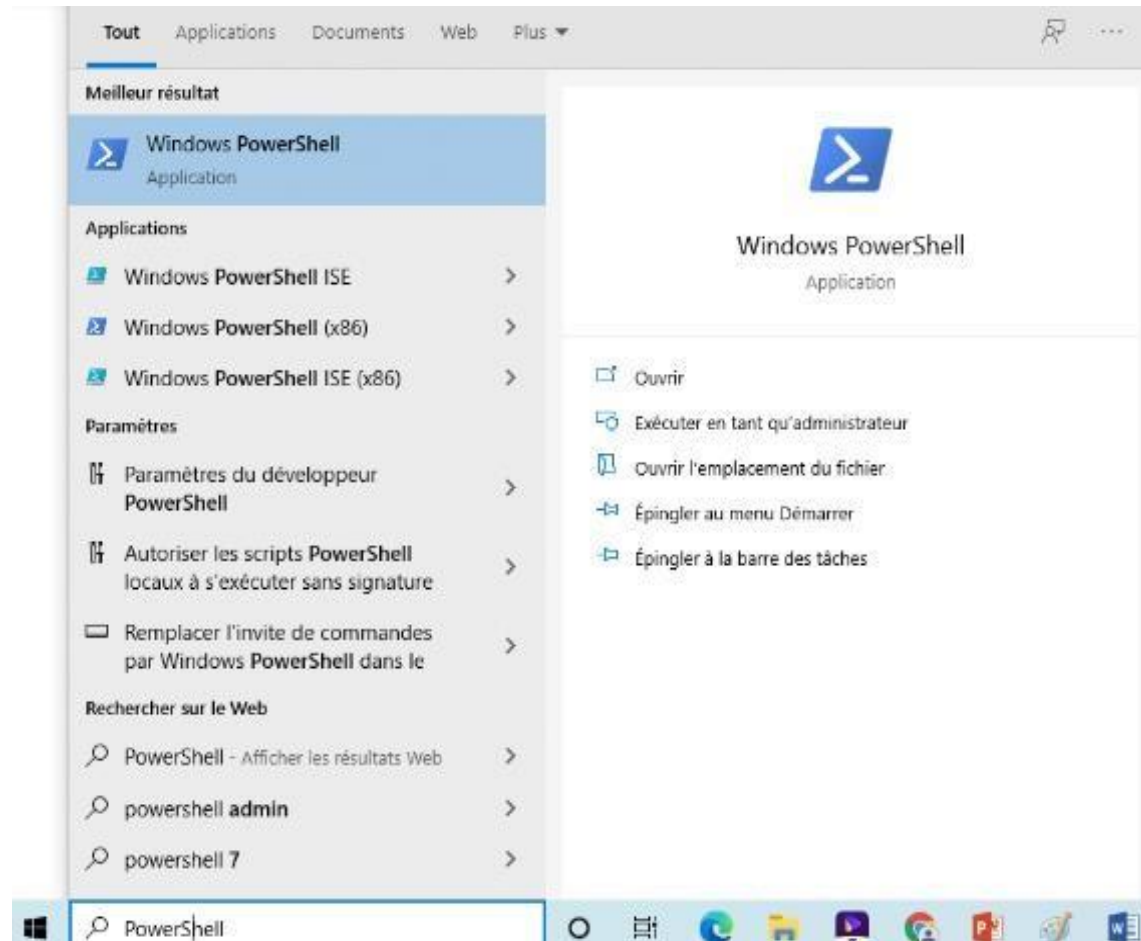


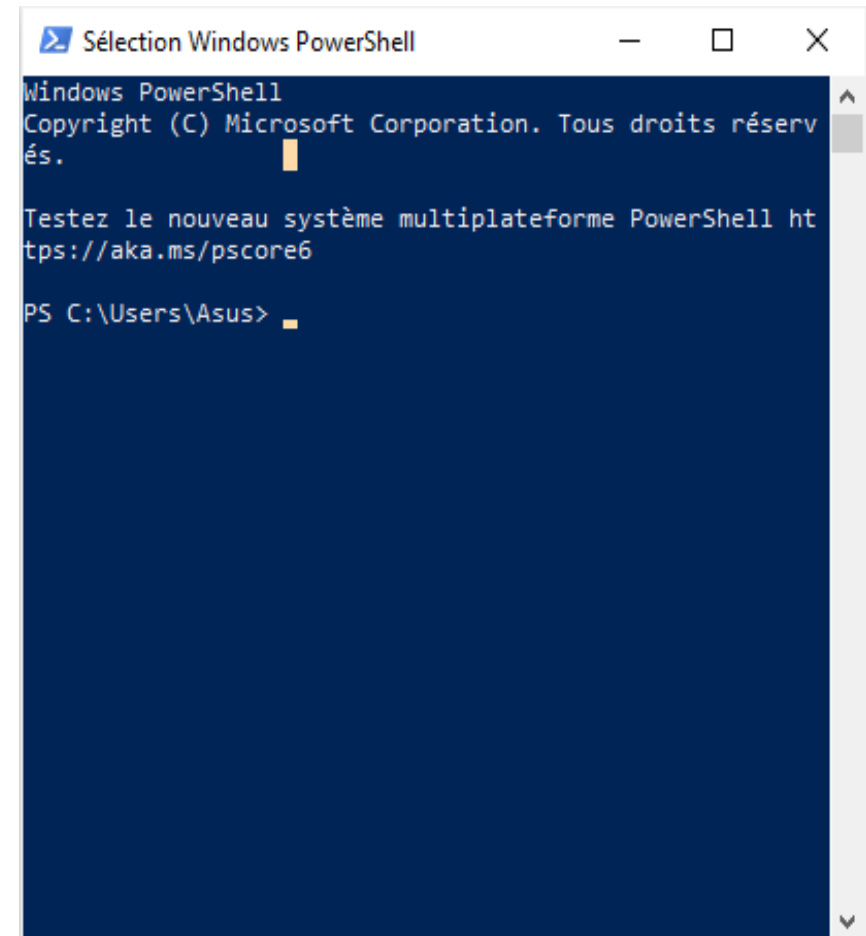
- **PowerShell** est un outil de ligne de commande (**CLI**) créé par Microsoft pour aider les administrateurs système et les utilisateurs avancés à automatiser des tâches et gérer des systèmes informatiques;
- Exemple d'utilisation : Get-Service;
- **PowerShell ISE** (**Integrated Scripting Environment**) est un environnement de développement graphique pour écrire, tester et déboguer des scripts PowerShell. Il propose une interface plus conviviale que la ligne de commande classique;
- ISE permet d'écrire des scripts PowerShell dans une fenêtre dédiée avec des outils avancés comme la coloration syntaxique, la complétion automatique des commandes, et la possibilité de tester des scripts en partie ou en totalité.

Explorer Windows

Maîtriser l'utilisation de PowerShell



recherche application PowerShell sous Windows 10



Interface de l'application PowerShell sous Windows 10

Explorer Windows

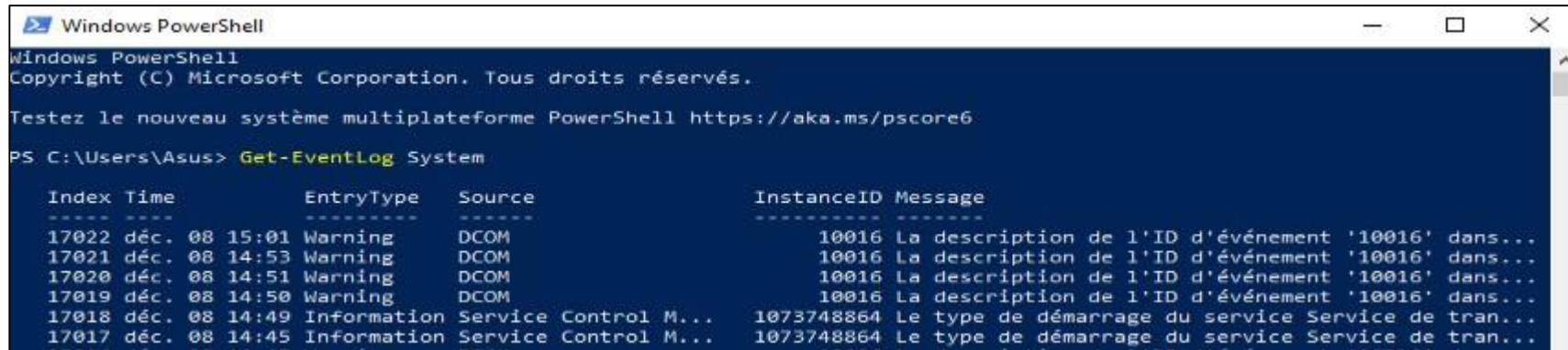
Maîtriser l'utilisation de PowerShell

Fonctionnement Windows PowerShell :

PowerShell est **composé** principalement de **deux parties**, le **moteur PowerShell** et le **langage de script PowerShell**.

Le moteur PowerShell

- **Permet** à l'utilisateur **d'accéder** aux **différentes fonctions** internes **du système d'exploitation SE** **via le clavier**.
- **Les commandes** du programme sont **nommées cmdlets**. Ces commandes sont toujours **composées** d'un **verbe** et d'un **nom** au **singulier**, à titre d'exemple, **Stop-Process** ou **Sort-Object**.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\Asus> Get-EventLog System

Index Time          EntryType Source                                InstanceID Message
-----
17022 déc. 08 15:01 Warning DCOM                                10016 La description de l'ID d'événement '10016' dans...
17021 déc. 08 14:53 Warning DCOM                                10016 La description de l'ID d'événement '10016' dans...
17020 déc. 08 14:51 Warning DCOM                                10016 La description de l'ID d'événement '10016' dans...
17019 déc. 08 14:50 Warning DCOM                                10016 La description de l'ID d'événement '10016' dans...
17018 déc. 08 14:49 Information Service Control M... 1073748864 Le type de démarrage du service Service de tran...
17017 déc. 08 14:45 Information Service Control M... 1073748864 Le type de démarrage du service Service de tran...
```

Exécution de la commande Get-EventLog

Explorer Windows

Maîtriser l'utilisation de PowerShell

Fonctionnement Windows PowerShell :

PowerShell est **composé** principalement de **deux parties**, le **moteur PowerShell** et le **langage de script PowerShell**.

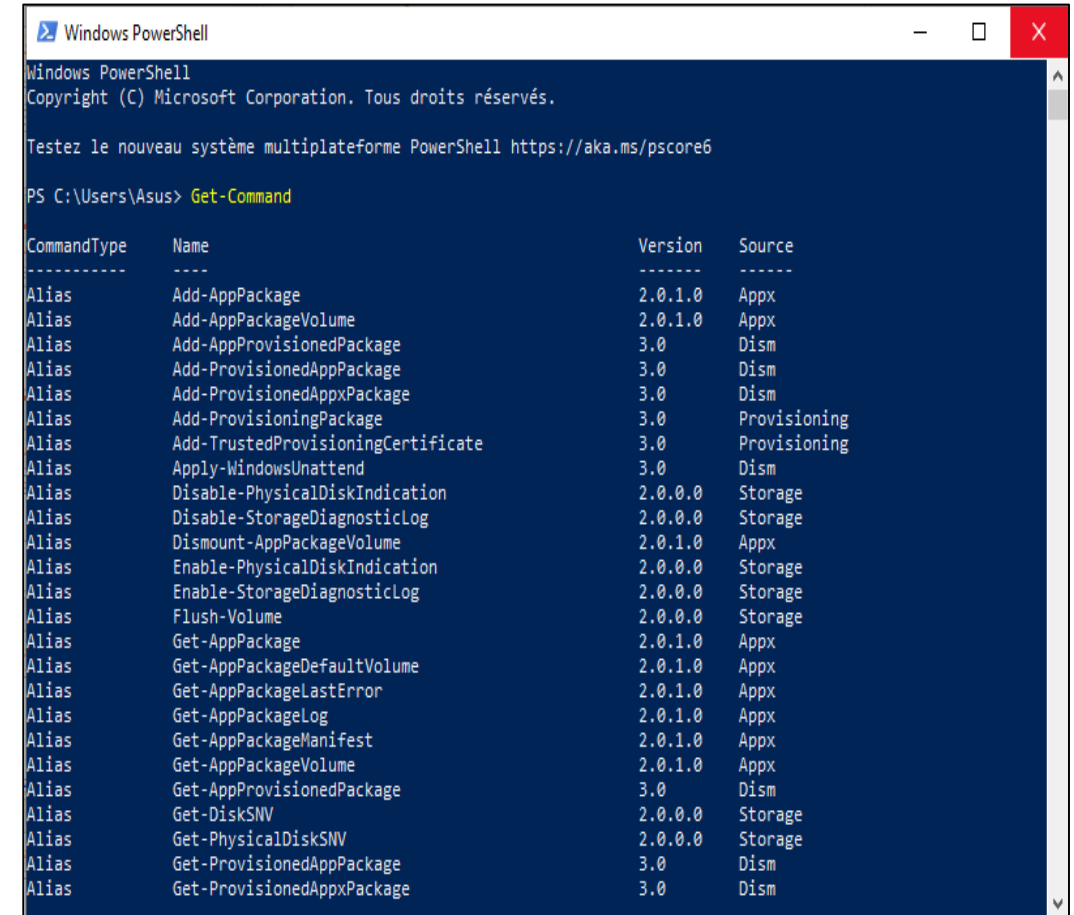
Le moteur PowerShell

- Le **moteur PowerShell** est la partie "**interne**" qui **exécute** les **commandes** et les **scripts** que vous tapez. C'est lui qui fait le travail de comprendre et d'exécuter les instructions, puis de donner des résultats.
- Le **langage de script** est le langage utilisé pour **écrire** les **commandes** et les **scripts**.
- **Les commandes** du programme sont **nommées cmdlets**. Ces commandes sont toujours **composées** d'un **verbe** et d'un **nom** au **singulier**, à titre d'exemple, **Stop-Process** ou **Sort-Object**.

Explorer Windows

Maîtriser l'utilisation de PowerShell

- ✓ Shell **offre plus de 100 commandlets** de **base** qui **contiennent** des **commandes standard** couramment **utilisées** à partir de **l'invite des commandes** et qui sont disponibles sous la forme d'un **alias**. Ces commandes sont destinées à faciliter l'utilisation du Shell par les utilisateurs.
- ✓ Pour **afficher** la **liste des commandes**, vous pouvez utiliser la commande suivante : **Get-Command**.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\Asus> Get-Command
```

CommandType	Name	Version	Source
Alias	Add-AppPackage	2.0.1.0	Appx
Alias	Add-AppPackageVolume	2.0.1.0	Appx
Alias	Add-AppProvisionedPackage	3.0	Dism
Alias	Add-ProvisionedAppPackage	3.0	Dism
Alias	Add-ProvisionedAppxPackage	3.0	Dism
Alias	Add-ProvisioningPackage	3.0	Provisioning
Alias	Add-TrustedProvisioningCertificate	3.0	Provisioning
Alias	Apply-WindowsUnattend	3.0	Dism
Alias	Disable-PhysicalDiskIndication	2.0.0.0	Storage
Alias	Disable-StorageDiagnosticLog	2.0.0.0	Storage
Alias	Dismount-AppPackageVolume	2.0.1.0	Appx
Alias	Enable-PhysicalDiskIndication	2.0.0.0	Storage
Alias	Enable-StorageDiagnosticLog	2.0.0.0	Storage
Alias	Flush-Volume	2.0.0.0	Storage
Alias	Get-AppPackage	2.0.1.0	Appx
Alias	Get-AppPackageDefaultVolume	2.0.1.0	Appx
Alias	Get-AppPackageLastError	2.0.1.0	Appx
Alias	Get-AppPackageLog	2.0.1.0	Appx
Alias	Get-AppPackageManifest	2.0.1.0	Appx
Alias	Get-AppPackageVolume	2.0.1.0	Appx
Alias	Get-AppProvisionedPackage	3.0	Dism
Alias	Get-DiskSNV	2.0.0.0	Storage
Alias	Get-PhysicalDiskSNV	2.0.0.0	Storage
Alias	Get-ProvisionedAppPackage	3.0	Dism
Alias	Get-ProvisionedAppxPackage	3.0	Dism

Avec la commande Get-Command, tous les commandes de PowerShell sont affichés.

Explorer Windows

Maîtriser l'utilisation de PowerShell

Maintenant, vous êtes invités à tester un ensemble de commande sur PowerShell.

- Pour **afficher** la **liste des commandes**, vous tapez la commande **Get-Command**.
- Pour **connaître** les **correspondances** entre les **anciennes commandes cmd** de l'invité de commande et les nouveaux **cmdlets de PowerShell**, vous pouvez utiliser la commande **Get-Alias**.

```
PS C:\Users\Asus> Get-Alias sc, dir, ls, man_
```

CommandType	Name	Version	Source
-----	----	-----	-----
Alias	sc -> Set-Content		
Alias	dir -> Get-ChildItem		
Alias	ls -> Get-ChildItem		
Alias	man -> help		

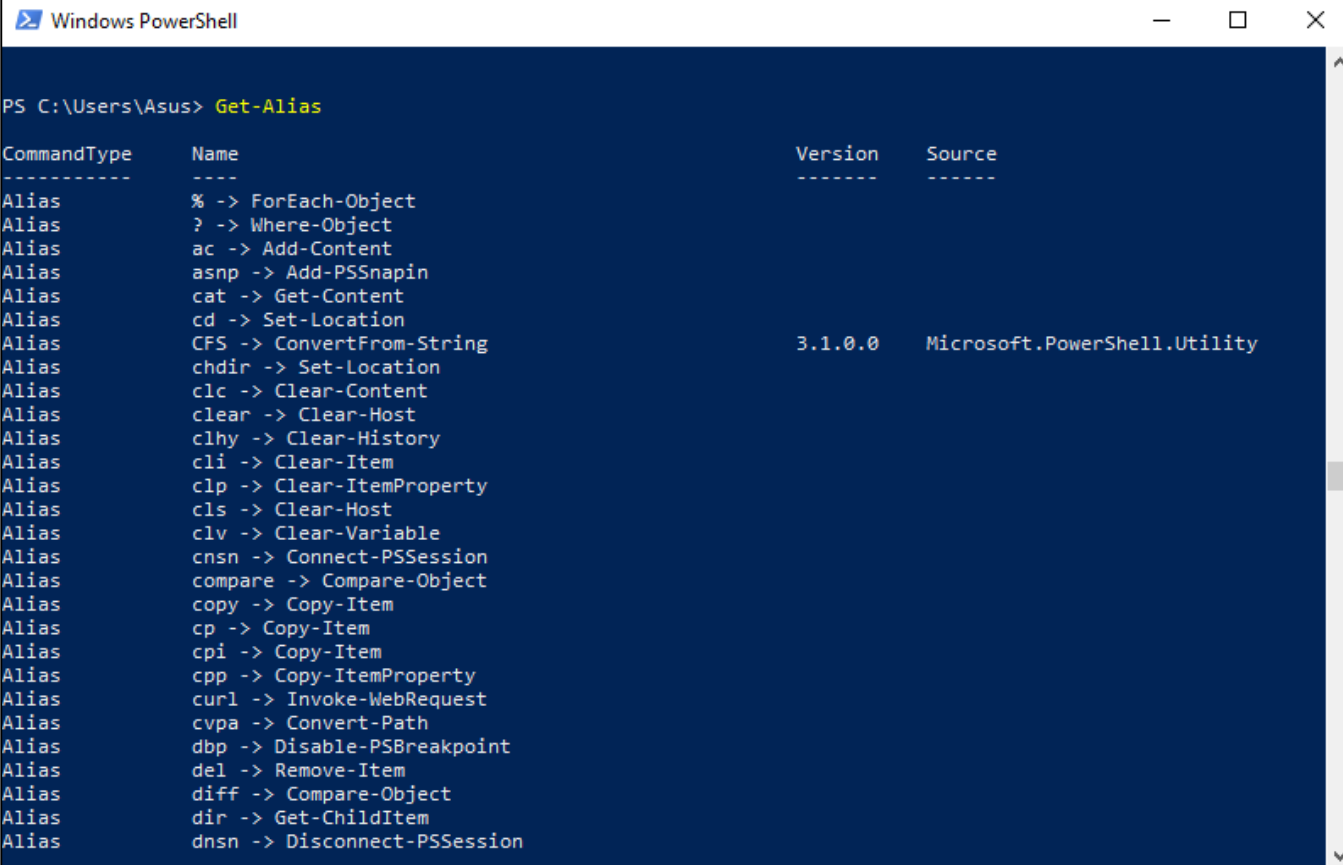
Exemple d'utilisation de la commande Get-Alias.

- Pour **afficher toutes les correspondances**, vous pouvez taper tout simplement la commande **Get-Alias**.

01 - Explorer Windows

Maîtriser l'utilisation de PowerShell

- Pour afficher toutes les correspondances, vous pouvez taper tout simplement la commande **Get-Alias**.



```
PS C:\Users\Asus> Get-Alias
```

CommandType	Name	Version	Source
Alias	% -> ForEach-Object		
Alias	? -> Where-Object		
Alias	ac -> Add-Content		
Alias	asnp -> Add-PSSnapin		
Alias	cat -> Get-Content		
Alias	cd -> Set-Location		
Alias	CFS -> ConvertFrom-String	3.1.0.0	Microsoft.PowerShell.Utility
Alias	chdir -> Set-Location		
Alias	clc -> Clear-Content		
Alias	clear -> Clear-Host		
Alias	clhy -> Clear-History		
Alias	cli -> Clear-Item		
Alias	clp -> Clear-ItemProperty		
Alias	cls -> Clear-Host		
Alias	clv -> Clear-Variable		
Alias	cnsn -> Connect-PSSession		
Alias	compare -> Compare-Object		
Alias	copy -> Copy-Item		
Alias	cp -> Copy-Item		
Alias	cpi -> Copy-Item		
Alias	cpl -> Copy-ItemProperty		
Alias	curl -> Invoke-WebRequest		
Alias	cvpa -> Convert-Path		
Alias	dbp -> Disable-PSBreakpoint		
Alias	del -> Remove-Item		
Alias	diff -> Compare-Object		
Alias	dir -> Get-ChildItem		
Alias	dnsn -> Disconnect-PSSession		

la commande Get-Alias.

Explorer Windows

Maîtriser l'utilisation de PowerShell

Cmdlets de commandes dans PowerShell de date et heure

- Si vous voudrez **changer la date** et l'heure de l'appareil, il suffit d'utiliser la commande **Set-Date** suivi du paramètre **-date** suivi de la nouvelle date et de l'heure. Par exemple, pour définir la date et l'heure à 10h30 le 6 novembre 2022, vous allez exécuter la ligne suivante :

```
Set-Date -date "6/11/2022 10:30 AM"
```

- pour faire **avancer l'horloge** de cinq jours, vous tapez une commande qui utilise l'applet de commande **Get-Date** et la **méthode AddDays** :

```
Set-Date (Get-Date).AddDays(5)
```

- Il existe également d'autres méthodes que vous pouvez utiliser, telles que **AddHours**, **AddMinutes** et **AddSeconds**, ces options nous permettent d'ajuster les heures, les minutes ou les secondes en fonction des contraintes soit météorologiques soit administratives. Par exemple, dans les pays où l'été a une incidence sur l'heure, vous pouvez ajuster l'horloge moins de deux heures en exécutant la ligne suivante :

```
Set-Date (Get-Date).AddHours(-1)
```

- Maintenant, vous pouvez **visualiser la date** avec la commande **Get-Date**.

```
PS C:\Users\Asus> Get-Date  
vendredi 4 février 2022 19:09:02
```

01 - Explorer Windows

Maîtriser l'utilisation de PowerShell

Cmdlets de commandes dans PowerShell de gestion des fichiers : Créer un nouveau dossier ou fichier

- Pour **créer un nouveau fichier ou dossier**, vous pouvez utiliser la cmdlet **New-Item**. Premièrement, vous allez créer un **nouveau répertoire** appelé **CommandePowerShell** dans le dossier **C:\CoursSystème**. Pour cela, vous allez exécuter **New-Item** en regard du chemin d'accès complet au nouveau dossier et au nouveau type d'élément (que vous pouvez spécifier à l'aide du paramètre **-type** et du répertoire de valeurs). Maintenant, vous pouvez tester la commande suivante et observer son résultat.

```
PS C:\Users\Asus> New-Item c:\CoursSystème\CommandePowerShell -type directory

Répertoire : C:\CoursSystème

Mode                LastWriteTime         Length Name
----                -
d-----          04/02/2022    20:10             CommandePowerShell
```

- Deuxièmement, vous allez créer un **nouveau fichier**, vous devez suivre le même processus, en spécifiant le nom complet du chemin, mais en configurant le type de fichier. Maintenant, vous pouvez tester la commande suivante et observer son résultat.

```
PS C:\Users\Asus> New-Item c:\CoursSystème\CommandePowerShell\introduction.txt -type file

Répertoire : C:\CoursSystème\CommandePowerShell

Mode                LastWriteTime         Length Name
----                -
-a-----          04/02/2022    20:33             0 introduction.txt
```

Cmdlets de commandes dans PowerShell de gestion des fichiers : Créer un nouveau dossier ou fichier

- Si vous essayez de créer un nouveau fichier dont le nom existe déjà, une exception **IOException** est levée et un message s'affiche indiquant que **le fichier existe**.

```
PS C:\Users\Asus> New-Item c:\CoursSystème\CommandePowerShell\introduction.txt -type file
New-Item : Le fichier 'C:\CoursSystème\CommandePowerShell\introduction.txt' existe déjà.
Au caractère Ligne:1 : 1
+ New-Item c:\CoursSystème\CommandePowerShell\introduction.txt -type fi ...
+ ~~~~~
+ CategoryInfo          : WriteError: (C:\CoursSystème...ntroduction.txt:String) [New-Item], IOException
+ FullyQualifiedErrorId : NewItemIOError,Microsoft.PowerShell.Commands.NewItemCommand
```

- Si vous voulez quand même **écraser le fichier existant**, vous pouvez tout simplement ajouter le paramètre **-force** comme ceci:

```
PS C:\Users\Asus> New-Item c:\CoursSystème\CommandePowerShell\introduction.txt -type file -force

Répertoire : C:\CoursSystème\CommandePowerShell

Mode                LastWriteTime         Length Name
----                -
-a----           04/02/2022   20:39             0 introduction.txt
```

Explorer Windows

Maîtriser l'utilisation de PowerShell

Cmdlets de commandes dans PowerShell de gestion des fichiers : copier des fichier ou des dossier

- Pour **copier des dossiers et des fichiers**, vous utilisez la cmdlet Copy-Item qui vous permet de copier un élément sous forme de fichier ou de dossier dans un autre emplacement.

Par exemple, vous pouvez exécuter la commande suivante qui copie le fichier Introduction.docx du dossier D: \CoursSystème dans le dossier C:\ CoursSystème :

```
Copy-Item D:\CoursSystème\Introduction.docx C:\CoursSystème
```

- Si vous souhaitez **copier** tout **le contenu d'un dossier**, y compris les sous-dossiers, vous devriez tout simplement utiliser un caractère générique comme le suivant :

```
Copy-Item D:\CoursSystème\* C:\Cours Système
```

- Si vous voulez juste **copier** les **fichiers** avec une **certaine extension**, par exemple des images, vous exécutez la ligne suivante :

```
Copy-Item D:\CoursSystème\*.png C:\Cours Système
```

Explorer Windows

Maîtriser l'utilisation de PowerShell

Cmdlets de commandes dans PowerShell de gestion des fichiers : Supprimer un fichier ou un dossier

- Si vous voulez **supprimer des fichiers** ou des dossiers du système, vous allez tout simplement utiliser la cmdlet **Remove-Item** en utilisant la ligne suivante:

```
PS C:\Users\Asus> Remove-Item c:\CoursSystème\CommandePowerShell\introduction.txt
PS C:\Users\Asus>
```

- Egalement, vous pouvez utiliser des **caractères génériques** afin de supprimer plusieurs éléments.

Par exemple, la commande suivante supprime tous les fichiers du chemin C: \CoursSystème

```
PS C:\Users\Asus> Remove-Item c:\CoursSystème\*
PS C:\Users\Asus>
```

- Autres options sont disponibles qui permettent de spécifier les éléments à supprimer et lesquels non, pour cela, vous pouvez tester la commande suivante:

```
Remove-Item c:\scripts\* -include *.txt -exclude *Tests*
```

01 - Explorer Windows

Maîtriser l'utilisation de PowerShell

Cmdlets de commandes dans PowerShell de gestion des fichiers : Déplacer un fichier ou un dossier

- Si vous voulez déplacer un fichier ou un dossier d'un emplacement à un autre, vous utilisez la commande **Move-Item**.

Par exemple, vous allez déplacer un dossier du lecteur C vers le lecteur D en exécutant la ligne suivante :

```
PS C:\Users\Asus> Move-Item c:\CoursSystème\ D:\CoursSystème
PS C:\Users\Asus>
```

- Si vous essayez de déplacer un fichier ou un dossier qui n'existe pas, une exception **ItemNotFoundException** est levée comme suit :

```
PS C:\Users\Asus> Move-Item c:\CoursSystème\ D:\CoursSystème
Move-Item : Impossible de trouver le chemin d'accès « C:\CoursSystème\ », car il n'existe pas.
Au caractère Ligne:1 : 1
+ Move-Item c:\CoursSystème\ D:\CoursSystème
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\CoursSystème:String) [Move-Item], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.MoveItemCommand
```

- Finalement, si l'**élément** à déplacer **existe déjà** dans le chemin de destination et vous voulez quand même le déplacer vous pouvez utiliser le paramètre **-force** comme suit :

```
PS C:\Users\Asus> Move-Item c:\CoursSystème\ D:\CoursSystème -force
PS C:\Users\Asus>
```

01 - Explorer Windows

Maîtriser l'utilisation de PowerShell

Cmdlets de commandes dans PowerShell de gestion des fichiers : Renommer un fichier ou un dossier

- Si vous voulez **renommer un fichier ou un dossier**, vous pouvez exécuter la ligne de commande suivante :

```
PS C:\Users\Asus> Rename-Item D:\CoursSystème CoursSystèmeExploitation_
PS C:\Users\Asus>
```

- Si vous essayez de donner le même nom à un dossier ou à un fichier, une exception **PSInvalidOperationException** est levée comme suit :

```
PS C:\Users\Asus> Rename-Item D:\CoursSystème CoursSystèmeExploitation
Rename-Item : Impossible de renommer l'élément situé à l'emplacement « D:\CoursSystème », car il n'existe pas.
Au caractère Ligne:1 : 1
+ Rename-Item D:\CoursSystème CoursSystèmeExploitation
+ ~~~~~
+ CategoryInfo          : InvalidOperation : (:) [Rename-Item], PSInvalidOperationException
+ FullyQualifiedErrorId : InvalidOperation,Microsoft.PowerShell.Commands.RenameItemCommand
```

Vérifier l'existence d'un fichier ou d'un dossier

- Si vous voulez **vérifier l'existence d'un fichier ou un dossier**, vous pouvez utiliser la cmdlet **Test-Path**. Par exemple, cette commande vérifie si le fichier

C: \ CoursSystèmeExploitation\ introduction.txt existe ou non :

```
PS C:\Users\Asus> Test-Path D:\CoursSystèmeExploitation\introduction.txt
False
PS C:\Users\Asus>
```

Explorer Windows

Maîtriser l'utilisation de PowerShell

- Le table contient d'autres exemples de cmdlets qui sont fréquemment utilisées ainsi que leurs alias associés, et il en existe d'autres :

Alias	Cmdlet	Fonction
Cd	Set-Location	Modifier le répertoire courant
dir	Get-ChildItem	Lister tous les éléments d'un dossier
gi	Get-Item	Appeler un élément spécifique
Ps	Get-Process	Lister tous les processus
gsv	Get-Service	Lister tous les services installés
gm	Get-Member	Afficher toutes les propriétés et méthodes d'un objet
clear	Clear-Host	Vider l'hôte PowerShell

Table : Exemple de commande Shell.

- Maintenant, vous pouvez tester chaque commande dans l'application PowerShell de votre PC. Vous pouvez également découvrir toute la liste des commandes en visitant ce lien [Table des commandes de base PowerShell](#)
- Une grande partie du [langage de script](#) de PowerShell peut vous sembler familier à partir de C# et d'autres langages de script. Non seulement vous pouvez l'utiliser pour écrire vos propres cmdlets (et les partager avec d'autres utilisateurs si nécessaire), mais vous pouvez **également regrouper plusieurs commandes consécutives dans un fichier script.ps1** pour **étendre les fonctionnalités du shell**. De nombreux exemples d'applications sont envisageables, de l'exécution de tâches de routine simples à l'automatisation presque complète des processus de surveillance et de contrôle. Les scripts contiennent toujours une description de l'application et sont exécutés avec le préfixe « . » suivi du chemin complet du fichier.

Exercice 6:

1. Définir PowerShell.
2. Comparer PowerShell & PowerShell ISE.
3. Quels sont les principaux composants de PowerShell ?
4. Donner la syntaxe d'une cmdlet (commande PowerShell).
5. Citer des exemples de cmdlets PowerShell et expliquer leur fonctionnement.

TP N° 9