

# PowerShell

## PowerShell : Introduction et Guide Avancé

[https://lecoursgratuit.com/wp-content/uploads/2024/05/PowerShell\\_Guide\\_pdf.pdf](https://lecoursgratuit.com/wp-content/uploads/2024/05/PowerShell_Guide_pdf.pdf)

### 1. Qu'est-ce que PowerShell ?

PowerShell est un langage de script et une interface en ligne de commande développé par Microsoft pour l'automatisation des tâches et la gestion des configurations système sur les plates-formes Windows. Il offre un ensemble de commandes appelées 'cmdlets' (command lets) qui permettent d'effectuer diverses opérations sur le système.

### 2. Installation de PowerShell :

- Vérification de la version de PowerShell : `$PSVersionTable.PSVersion` - Installation sur différentes versions de Windows : # Installation de PowerShell 7 sur Windows 10

```
iex '& { $(irm https://aka.ms/install-powershell.ps1) } -UseMSI'
```

### 3. Lancement de PowerShell :

- Interface en ligne de commande : # Lancer PowerShell depuis l'invite de commande

```
powershell
```

- Interface graphique (PowerShell ISE) :

# Lancer PowerShell ISE

```
powershell_ise
```

### 4. Les bases de PowerShell :

- Cmdlets : # Afficher la liste des processus en cours d'exécution

```
Get-Process
```

- Variables :

```
$nom = 'John'
```

```
Write-Host 'Bonjour, $nom !'
```

## 5. Exploration du système de fichiers :

- Navigation entre les répertoires : # Accéder au répertoire C:\Users

```
cd C:\Users
```

- Création de fichiers : # Créer un nouveau fichier texte

```
New-Item -Path .\example.txt -ItemType File
```

## 6. Gestion des processus et des services :

- Recherche des processus en cours d'exécution : # Rechercher les processus notepad en cours d'exécution

```
Get-Process -Name notepad
```

- Interrogation des services : # Afficher la liste des services en cours

```
Get-Service
```

## 7. Gestion des utilisateurs et des groupes :

- Création d'un nouvel utilisateur : # Créer un nouvel utilisateur

```
New-LocalUser -Name 'utilisateur1' -Password (ConvertTo-SecureString 'P@ssw0rd' -AsPlainText -Force)
```

- Ajout d'un utilisateur à un groupe :

# Ajouter un utilisateur à un groupe

```
Add-LocalGroupMember -Group 'Administrateurs' -Member 'utilisateur1'
```

## 8. Automatisation des tâches :

- Scripting : # Exemple de script : sauvegarde des fichiers

```
$source = 'C:\Source'  
$destination = 'D:\Backup'  
Copy-Item $source -Destination $destination -Recurse
```

## 9. Gestion des fichiers et dossiers avancée :

- Renommage de fichiers en masse : # Renommer tous les fichiers avec l'extension .txt en ajoutant un préfixe 'backup\_'

```
Get-ChildItem -Path C:\Path\To\Folder -Filter *.txt | Rename-Item -NewName { 'backup_' + $_.Name }
```

- Suppression de fichiers/dossiers : # Supprimer un fichier

```
Remove-Item -Path C:\Path\To\File.txt
```

# Supprimer un dossier et son contenu récursivement

```
Remove-Item -Path C:\Path\To\Folder -Recurse
```

## 10. Gestion des registres Windows :

- Lecture de clés de registre : # Lire la valeur d'une clé de registre

```
Get-ItemProperty -Path 'HKCU:\Software\Microsoft\Windows\CurrentVersion' -Name 'ProgramFilesDir'
```

- Modification de clés de registre :

# Modifier la valeur d'une clé de registre

```
Set-ItemProperty -Path 'HKCU:\Software\Microsoft\Windows\CurrentVersion' -Name 'ProgramFilesDir' -Value 'C:\Program Files (x86)'
```

## 11. Interactions réseau :

- Test de connectivité :

# Vérifier la connectivité avec un hôte distant

```
Test-Connection -ComputerName 'example.com' -Count 4
```

- Récupération d'informations sur les interfaces réseau :

# Obtenir la liste des interfaces réseau

```
Get-NetAdapter
```

## 12. Gestion des données :

- Manipulation de fichiers CSV :

# Importer un fichier CSV

```
$data = Import-Csv -Path 'C:\Path\To\Data.csv'
```

# Afficher les données

```
$data
```

- Manipulation de fichiers JSON :

# Importer un fichier JSON

```
$data = Get-Content -Path 'C:\Path\To\Data.json' | ConvertFrom-Json
```

# Afficher les données

```
$data
```

## Annexe A : Glossaire PowerShell

- **Variable** : Un conteneur pour stocker des valeurs ou des objets dans PowerShell.
- **Script** : Un fichier texte contenant une séquence de commandes PowerShell qui peuvent être exécutées ensemble.
- **Interface en ligne de commande (CLI)** : Une interface utilisateur qui permet à l'utilisateur de communiquer avec le système en saisissant des commandes textuelles.
- **Interface graphique utilisateur (GUI)** : Une interface utilisateur qui permet à l'utilisateur d'interagir avec le système à l'aide d'éléments visuels tels que des boutons, des menus et des fenêtres.
- **Automatisation** : Le processus d'exécution de tâches répétitives ou complexes de manière programmée plutôt que manuelle.
- **Fichier batch** : Un fichier texte contenant une séquence de commandes qui peuvent être exécutées par le shell de commandes Windows (cmd.exe).
- **Registre Windows** : Une base de données hiérarchique utilisée par le système d'exploitation Windows pour stocker des informations de configuration, telles que les paramètres système et les informations sur les logiciels installés.
- **Cmdlet** : Abréviation de 'command let', il s'agit des commandes de base de PowerShell qui effectuent des actions sur le système

## Annexe B : Ressources supplémentaires

### - Sites Web :

- Site officiel de PowerShell :

[<https://docs.microsoft.com/en-us/powershell/>](<https://docs.microsoft.com/en-us/powershell/>)

- PowerShell Gallery : [<https://www.powershellgallery.com/>](<https://www.powershellgallery.com/>)

### - Livres :

- 'Learn PowerShell in a Month of Lunches' par Don Jones et Jeffrey Hicks

- 'Windows PowerShell Cookbook' par Lee Holmes

### - Communauté :

- PowerShell Subreddit :

[<https://www.reddit.com/r/PowerShell/>](<https://www.reddit.com/r/PowerShell/>)

- PowerShell.org : [<https://powershell.org/>](<https://powershell.org/>)

## Listes des commandes powershell

powershell-commandes.pdf

### Philosophie générale des commandes PowerShell

PowerShell repose sur des cmdlets, des commandes natives conçues pour être cohérentes, lisibles et prévisibles.

Une cmdlet :

- effectue une action précise
- renvoie des objets
- s'intègre naturellement dans un pipeline

Tu ne travailles pas avec du texte brut, mais avec des entités structurées.

### Syntaxe générale d'une cmdlet

Toutes les cmdlets suivent la convention :

Verbe-Nom

Exemples :

- Get-Process : récupérer des processus
- Get-Service : récupérer des services
- Set-Date : modifier un paramètre système
- Remove-Item : supprimer un élément

Le verbe indique l'action, le nom indique la cible. Cette régularité te permet de deviner des commandes même sans les connaître.

## Les commandes renvoient des objets

Contrairement à CMD, une commande PowerShell ne renvoie pas du texte, mais des objets.

**Get-Member** permet d'inspecter ce qu'une commande renvoie :

```
1 Get-Process | Get-Member
```

Tu découvres :

- les propriétés (Name, Id, CPU, etc.)
- les méthodes (actions possibles)

C'est cette structure qui rend PowerShell puissant et utile pour l'administration système.

## Trouver de l'aide : Get-Help

PowerShell intègre sa propre documentation.

Afficher l'aide d'une commande :

```
1 Get-Help Get-Process
```

Afficher uniquement des exemples :

```
1 Get-Help Get-Process -Examples
```

Consulter la documentation complète :

```
1 Get-Help Get-Process -Full
```

☐ En pratique, Get-Help est l'outil le plus important à maîtriser pour devenir autonome.

## Rechercher des commandes : Get-Command

Tu n'as pas besoin de connaître le nom exact d'une cmdlet.

Exemple :

## 1 Get-Command \*service\*

PowerShell te propose toutes les commandes liées aux services. C'est la méthode recommandée pour explorer un nouveau domaine (réseau, système, fichiers...).

### Alias : raccourcis utiles (mais à utiliser avec prudence)

PowerShell propose des alias pour aller plus vite :

Alias Cmdlet réelle

```
dir Get-ChildItem
```

Alias Cmdlet réelle

- ls Get-ChildItem
- cat Get-Content
- ps Get-Process

Les alias sont pratiques en console, mais à éviter dans les scripts, au de garantir lisibilité et portabilité.

### Cmdlets fondamentales à connaître

Voici une sélection de cmdlets essentielles, classées par usage.

#### Fichiers et dossiers

Cmdlet	Rôle
Get-ChildItem	Lister fichiers et dossiers
Set-Location	Changer de dossier
New-Item	Créer fichier ou dossier
Copy-Item	Copier
Move-Item	Déplacer
Remove-Item	Supprimer

Système et administration Cmdlet Rôle Get-Process Lister les processus Stop-Process Arrêter un processus Get-Service Lister les services Start-Service Démarrer un service Stop-Service Arrêter un service Restart-Computer Redémarrer la machine Réseau Cmdlet Rôle Test-Connection Tester la connectivité (ping) Resolve-DnsName Résoudre un nom DNS Get-NetAdapter Afficher les interfaces réseau Lecture et export Cmdlet Rôle Get-Content Lire un fichier Set-Content Écrire dans un fichier Export-Csv Exporter vers CSV ConvertTo-Json Convertir en JSON Conclusion Dans ce chapitre, tu as compris : 4 | <https://v2.ttrinfo.be> © 2025 Mr. Lambrechts comment sont construites les cmdlets PowerShell que les commandes renvoient des objets comment découvrir et documenter une commande quelles sont les cmdlets essentielles pour débuter Ces bases sont indispensables avant d'aborder le pipeline avancé, le ltrage, le tri et l'automatisation. □ Le ltrage et le tri des objets seront abordés dans un chapitre spécifique, une fois ces fondations maîtrisées.

Last  
update: start:windows11:powershell <https://www.fablab37110.chanterie37.fr/doku.php?id=start:windows11:powershell&rev=1772464229>  
2026/03/02 16:10

---

From:  
<https://www.fablab37110.chanterie37.fr/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:  
<https://www.fablab37110.chanterie37.fr/doku.php?id=start:windows11:powershell&rev=1772464229>

Last update: **2026/03/02 16:10**

