

PowerShell - Commandes fondamentales

Ce chapitre présente les bases indispensables pour utiliser PowerShell efficacement. Il ne s'agit pas d'apprendre des commandes par cœur, mais de comprendre comment elles sont construites, comment découvrir celles dont tu as besoin et comment interpréter ce qu'elles renvoient. Ces notions sont fondamentales pour tout technicien système.

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 fiable 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**, afin 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 :

- comment sont construites les cmdlets PowerShell
- que les commandes renvoient des **objets**
- comment découvrir et documenter une commande
- quelles sont les cmdlets essentielles pour débiter

Ces bases sont indispensables avant d'aborder le pipeline avancé, le filtrage, le tri et l'automatisation.

👉 Le filtrage et le tri des objets seront abordés dans un chapitre spécifique, une fois ces fondations maîtrisées.