

Chapitre 3

Comment apprendre PowerShell ?

1. Par où commencer ?

Lorsque nous commençons à apprendre un langage, une des questions qui se pose le plus souvent est : par où commencer ? C'est d'ailleurs souvent cela qui constitue un « frein » à l'envie de s'y mettre :

- manque de motivation ;
- impression qu'il y a trop de choses à apprendre avant d'arriver à ses fins.

En ce qui nous concerne, nous en apprenons encore tous les jours sur PowerShell, ce qui rend cela passionnant. Il n'y a pas de méthode exacte, cela dépend aussi de votre historique :

- Utilisez-vous déjà un autre type de langage ?
- Êtes-vous à l'aise avec le scripting ?
- Aimez-vous apprendre tout seul ?

L'important est de ne pas brûler les étapes en voulant partir tout de suite sur des scripts complexes.

Il arrive régulièrement que des gens nous contactent car ils souhaitent apprendre PowerShell et nous demandent comment faire tel ou tel script tout en ne connaissant pas les bases.

Cela se retrouve également souvent sur des groupes. Il existe de nombreux forums ou groupes Facebook relatifs à PowerShell ou l'administration système. Sur ceux-ci, de nombreuses personnes demandent également des scripts tout faits. La réponse est généralement négative car, s'agissant de groupes faits pour apprendre, cela va à l'encontre de leur philosophie. Cela ne sert évidemment strictement à rien, ce n'est pas en partant tête baissée sur un script complexe que l'on va apprendre quoi que ce soit, surtout si celui-ci est réalisé par quelqu'un d'autre.

Demandez-vous ce que vous souhaitez accomplir, quels sont vos objectifs.

Retours sur nos débuts

Damien Van Robaey

En ce qui me concerne, avant de commencer PowerShell, j'avais déjà de toutes petites bases en *Visual Basic Script* (VBS) ou Batch, rien de vraiment sorcier ou compliqué.

N'ayant pas fait des études dans le développement, la partie scripting était un peu compliquée pour moi au début, je dirais même très compliquée.

En effet, j'ai commencé par un BTS IG (maintenant SIO) en option réseau.

En toute honnêteté, avant de choisir ce BTS, je n'avais jamais fait d'informatique. J'avais un PC dont je me servais uniquement pour aller sur Internet, MSN, et je n'avais jamais entendu parler de développement ou algorithme...

Je me souviens encore de mon premier contrôle en développement. La professeure de l'époque, pour je ne sais quelle raison, pensait que j'étais brillant dans ce domaine. Elle changea bien vite d'avis en me rendant ce devoir avec une belle note de 0, en soulignant qu'elle était déçue, ce qui ne me donna pas vraiment envie de m'y mettre.

En toute logique, je suis donc parti vers l'option réseau plutôt que développement.

Je me suis vraiment mis à la logique de scripting et algorithmique lors d'une de mes premières missions après être entré dans la vie active. Mon responsable et client de l'époque m'a fait confiance en me disant « Tu devras progresser sur cette partie-là » et je l'en remercie encore (merci Laurent).

C'est en partie pour cette raison que j'ai eu l'idée de ce livre, sur comment débiter PowerShell, en me replongeant quelques années auparavant lorsque je parcourais tout Internet pour apprendre.

Tout ceci pour dire que n'importe qui peut y arriver, il suffit de trouver le bon moment et d'en avoir l'envie.

Avant de démarrer PowerShell, j'avais réalisé quelques petits scripts pour gérer un poste de travail :

- lister l'état des services Windows dans une interface HTA ;
- une interface HTA pour afficher à l'utilisateur des informations rapides sur sa machine, afin de les donner au support ;
- des interfaces HTA pour des déploiements de systèmes d'exploitation avec *Microsoft Deployment Toolkit* (MDT).

Mon début d'apprentissage de PowerShell résidait donc dans l'idée de transformer ces différents scripts en quelque chose de plus récent avec PowerShell.

Jérôme Bezet-Torres

Pour ma part, mon parcours est encore plus atypique car j'ai commencé mes études dans l'agro-alimentaire (BTS) avec un titre de responsable de production en industrie agro-alimentaire (IUP). J'ai décidé de reprendre mes études pour devenir administrateur réseau. À la suite de cela, j'ai démissionné pour devenir administrateur réseau et formateur sur les technologies Microsoft. Après sept années, je suis rentré dans l'Éducation nationale.

C'est à ce moment-là que j'ai décidé d'essayer d'automatiser le maximum de mes travaux pratiques, et c'est là que PowerShell s'est imposé comme le langage de référence. Alors oui, mes débuts n'ont pas été si faciles, mais une chose qui est importante, c'est de ne jamais abandonner. Mon formateur et ami maintenant (Laurent) me disait « Apprends le langage PowerShell, il te permettra de te faciliter les choses plus tard ». Et il avait raison.

2. Transformez vos vieux scripts

Si vous avez déjà réalisé des scripts avec Batch, *Visual Basic Script* (VBS), Python ou autre, une façon simple d'apprendre et comprendre PowerShell est de « convertir » ces scripts en PowerShell, comme évoqué plus haut. D'une part, vous avez créé ces scripts et en connaissez donc par cœur le contenu ; d'autre part, cela vous permettra d'apprendre la correspondance entre les commandes et la syntaxe de vos anciens scripts avec celles de PowerShell.

■ Remarque

Vous trouverez dans les chapitres De CMD à PowerShell et De VBS à PowerShell quelques exemples sur la manière de passer de VBS à PowerShell ou encore de Batch à PowerShell.

3. La clé de la réussite : l'autoformation

Lorsque nous avons commencé à apprendre PowerShell, il n'existait pas réellement de formation sur PowerShell.

Il en existe maintenant quelques-unes, dont celles dispensées chez ENI par Arnaud Petitjean, un des pionniers en matière d'apprentissage de ce langage et premier MVP français en PowerShell.

De manière générale, les formations vous permettent d'acquérir les bases et de vous sentir à l'aise avec ce pour quoi vous êtes venu. L'important derrière est de mettre en pratique ce que vous avez appris ou commencé à apprendre. Sans cela, vous oublierez vite le contenu de votre formation. Pour mettre en pratique et continuer à apprendre PowerShell, Internet est votre meilleur ami et atout. Il existe des centaines (voire milliers) de sites relatifs à PowerShell, astuces, outils, scripts...

La plupart des sites étant en anglais, il sera souvent plus judicieux de préférer les recherches en anglais plutôt qu'en français, même si de nombreuses ressources dans cette langue existent.

■ Remarque

Nous verrons dans le chapitre Comment s'améliorer une liste de ressources utiles pour progresser.

Tout est donc à votre disposition pour apprendre, il n'y a plus qu'à vous servir.

4. Maîtrisez l'anglais technique

Il existe de nombreux sites et forums en français pour obtenir de l'aide. Cependant, cela peut ne pas suffire. Il faudra alors faire une recherche du problème en anglais. En ce qui nous concerne, nous avons pour habitude de toujours rechercher en anglais. Il y aura beaucoup plus de réponses et, parmi elles, des réponses pertinentes.

5. Automatisez votre quotidien

Nous avons vu dans le chapitre Scripting et automatisation, ce qu'était l'automatisation et quels en étaient les avantages.

Pour apprendre PowerShell, habituez-vous, forcez-vous à automatiser les tâches manuelles que vous effectuez au quotidien.

La moindre petite action peut vous apprendre des choses ou permettre d'acquérir des automatismes :

- test d'existence de dossier ;
- création ou suppression de dossiers ;
- test d'existence de clé de registre ;
- création ou suppression de clés de registre ;
- vérification d'évènement dans le gestionnaire d'évènements Windows ;
- gestion des tâches planifiées ;

56 _____ Débuter avec PowerShell

- gestion du système Windows ;
- création de machines virtuelles ;
- installation de logiciels.

Tout (ou presque) est automatisable dans l'environnement Windows, et plus particulièrement avec PowerShell.

Essayez de « scripter » un peu tous les jours afin de :

- ne pas oublier ce que vous avez appris ;
- rester sur votre lancée et votre soif d'apprentissage.

6. Ne renoncez jamais

Si vous débutez avec PowerShell et que vous n'avez aucune base, la tâche peut vous sembler ardue. Il faut commencer par le commencement sans sauter les étapes, apprenez déjà les bases. L'apprentissage se faisant au quotidien, il faut prendre l'habitude d'automatiser tout ce que vous faites au quotidien.

7. Aidez-vous de l'intelligence artificielle

PowerShell est un outil puissant pour l'automatisation des tâches et la gestion des systèmes. Cependant, sa maîtrise peut représenter un défi, surtout pour les débutants.

L'intelligence artificielle, en particulier ChatGPT, offre une assistance précieuse pour apprendre, créer et optimiser des scripts PowerShell.

7.1 Générer un script complet

Prenons par exemple le cas d'usage suivant :

Nous souhaitons :

- créer un fichier toto.txt dans l'emplacement suivant
« c:\Windows\Temp » > ;

Comment apprendre PowerShell ? _____ 57

Chapitre 3

- vérifier l'existence de ce fichier ;
- dans le cas où celui-ci n'est pas présent, le créer automatiquement.

Notre prompt dans ChatGPT sera le suivant :

« Je souhaite créer un script PowerShell permettant de créer un fichier toto.txt dans C:\Windows\Temp en vérifiant d'abord si celui-ci existe. »

Ci-dessous le retour de ChatGPT :

Je souhaite créer un script PowerShell permettant de créer un fichier toto.txt dans C:\Windows\Temp en vérifiant d'abord si celui-ci existe.

D'accord, voici un exemple de script PowerShell simple qui fait exactement ça :

```
powershell

# Définir le chemin complet du fichier
$fichier = "C:\Windows\Temp\toto.txt"

# Vérifier si le fichier existe déjà
if (Test-Path -Path $fichier) {
    Write-Host "Le fichier existe déjà : $fichier"
} else {
    # Créer le fichier (vide)
    New-Item -Path $fichier -ItemType File -Force | Out-Null
    Write-Host "Le fichier a été créé : $fichier"
}
```

Explications :

- `$fichier` : variable contenant le chemin complet du fichier cible.
- `Test-Path` : vérifie l'existence du fichier.
- `New-Item` : crée le fichier s'il n'existe pas.
- `-Force` : évite les blocages si le dossier est protégé mais que vous avez les droits.
- `| Out-Null` : supprime la sortie inutile de la commande.

💡 Si vous exécutez ce script dans **C:\Windows\Temp**, vous aurez probablement besoin de lancer PowerShell en tant qu'administrateur pour éviter les erreurs d'accès.

Nous avons donc un bout de code complet permettant de faire ce que l'on souhaite.

7.2 Apprendre PowerShell

L'IA vous permet bien entendu de créer un script en partant de zéro. Cependant, il est toujours utile d'apprendre et non de simplement demander à une IA de faire quelque chose à sa place.

L'IA nous permet d'apprendre PowerShell en détails.

Prenons un autre cas d'usage :

Nous récupérons un script dans lequel il y a une cmdlet que nous ne comprenons pas. Dans notre exemple il s'agit de la cmdlet `New-LocalUser`. Nous pouvons donc demander à ChatGPT ce que fait cette cmdlet.

Ci-dessous le résultat retourné :

Que fait cette cmdlet New-LocalUser ?

La cmdlet `New-LocalUser` en PowerShell sert à **créer un nouveau compte utilisateur local** sur la machine où elle est exécutée.

Elle fait partie du module `Microsoft.PowerShell.LocalAccounts`, disponible à partir de **Windows 10 / Windows Server 2016** et ultérieur.

Usage de base

powershell

Copy

Edit

```
New-LocalUser -Name "MonUtilisateur" -Password (Read-Host -AsSecureString "Entrer le mot de passe")
```

Exemple complet

powershell

Copy Edit

```
# Définir un mot de passe sécurisé
$Password = Read-Host "Entrez le mot de passe" -AsSecureString

# Créer l'utilisateur local
New-LocalUser -Name "techuser" `
    -Password $Password `
    -FullName "Technicien Support" `
    -Description "Compte technique local" `
    -PasswordNeverExpires `
    -AccountNeverExpires
```

Remarques importantes

- **Droits requis** : vous devez exécuter PowerShell **en tant qu'administrateur** pour créer un compte local.
- **Différent d'AD** : cette cmdlet ne gère que les comptes locaux, pas ceux d'Active Directory (pour AD, on utilise `New-ADUser`).
- **Pas de gestion de groupes** : pour ajouter le nouvel utilisateur à un groupe local, on utilise ensuite `Add-LocalGroupMember`.

7.3 Déboguer ou améliorer un script existant

Prenons le cas d'usage suivant :

Nous avons récupéré ou créé un script, mais ce dernier ne fonctionne pas. Il est alors possible de demander de l'aide à ChatGPT pour corriger ce script.

Ci-dessous l'extrait de code posant problème :

```
■ New-Item Path "C:\Windows\Temp\toto.txt" -Force
```