Chapitre 4
Comment s'améliorer ?

1. Testez, faites des erreurs, recommencez

Tout le monde connaît cette expression disant que nous apprenons de nos erreurs, ou que c'est en commettant des erreurs que nous progressons. Il en va de même avec PowerShell. Lors de vos débuts, vous ferez sûrement de nombreuses erreurs, et cela est normal de continuer à en faire par la suite. Il faut donc continuer à tester tout ce que vous pouvez avec PowerShell, et si vous faites des erreurs, essayez d'en comprendre la cause et comment les résoudre.

Vous pouvez, par exemple, saisir l'erreur retournée dans l'invite de commandes dans Google, ou autre, puis chercher la bonne manière de résoudre votre problème. Si vous bloquez sur un script et que vous n'y parvenez toujours pas, il peut parfois être utile de faire appel à un œil extérieur.

Un avis externe peut donc parfois être la solution.

Quelques exemples d'erreurs qui peuvent vous arriver :

– Oublier de fermer une parenthèse : "(" mais pas de ")".
– Oublier de fermer une condition : "{" mais pas de "}".
– Oublier un $ à une variable.
– Mettre deux $$ à une variable.
– Oublier de déclarer une fonction.
– Oublier de commenter une ligne.

2. Ressources sur Internet

Ci-dessous une liste de sites internet dont un des sujets phares est PowerShell dans différents contextes :

– https://www.it-connect.fr/
– https://www.tech2tech.fr/
– https://www.powershell-scripting.com/
– https://powershell.org/
– https://www.systanddeploy.com/
– https://jm2k69.github.io/
– https://jdhitsolutions.com/blog/
– https://www.planetpowershell.com/
– https://powershellexplained.com/
– https://adamtheautomator.com/
– https://sid-500.com/
– https://blog.robsewell.com/
– https://dfinke.github.io/
– https://powers-hell.com/
– https://mikefrobbins.com/
– https://guyrleech.wordpress.com/
– https://4bes.nl/

– https://tech.nicolonsky.ch/
– https://veronicageek.com/
– https://www.commandline.ninja/
– https://dev.to/irontux

3. Personnes à suivre pour apprendre

Ci-dessous, une liste de personnes à suivre sur Twitter, qui publient régulièrement des articles, astuces…

– Damien Van Robaeys (@syst_and_deploy)
– Jérôme Bezet-Torres (@JM2K69)
– Thomas Maurer (@ThomasMaurer)
– Jeffrey Snover (@jsnover)
– Lee Holmes (@Lee_Holmes)
– French PowerShell User Group (@FrPSUG)
– Mathias Cody (@CodyMathis123)
– Christopher Kibble (@Christopher83)
– Andrew Jimenez (@AndrewJimenez_)
– Nathan Ziehnert (@Theznerd)
– Adam Driscoll (@Adamdriscoll)
– Jeff Hicks (@Jeff Hicks)
– Bruce Payette (@BrucePayette)
– Kevin Marquette (@KevinMarquette)
– Jeff Wouters (@JeffWouters)
– Aleksandar Nikolić (@Alexandair)
– Tyler Leonhardt (@TylerLeonhardt)
– Adam Bertram (@Adbertram)
– Patrick Gruenauer (@pewa2303)
– Rob Sewell (@sqldbawithbeard)

- Doug Finke (@dfinke)
- Stefan Stranger (@sstranger)
- @PotentEngineer : Daniel Ratliff
- ShayLevy (@ShayLevy)
- @TobiasPSP
- Steve Lee (@Steve_MSFT)
- June Blender (@juneb_get_help)
- David Segura (@SeguraOSD)
- Ben Reader (@powers_hell)
- Arnaud Petitjean (@apetitjean)
- Mike f Robbins (@mikefrobbins)
- James Petty (@PSJamesP)
- Guyr Leech (@Guyrleech)
- Jordan Benzing (@JordanTheITguy)
- Mathias R. Jessen (@IISResetMe)
- Mick Pletcher (@mick_pletcher)
- Barbara Forbes (@Ba4bes)
- Klys Przemyslaw (@PrzemyslawKlys)
- Heiko Brenn (@HeikoBrenn)
- Gael Colas (@Gaelcolas)
- Nicola Suter (@Nicolonsky)
- Veronique Lengelle (@Veronicageek)
- Richard Siddaway (@RSiddaway)
- Mike Kanakos (@MikeKanakos)
- Dave Carroll (@thedavecarroll)
- Laurent Lienhard (@IronTUX)

Chapitre 5
Quelques bonnes pratiques

1. Toujours vérifier l'aide proposée par PowerShell

Comprendre le fonctionnement d'une cmdlet ou une fonction spécifique peut parfois être complexe, surtout si de nombreux paramètres existent et que certains sont obligatoires.

Il existe pour cela une méthode très efficace permettant d'obtenir des informations sur le fonctionnement et l'utilisation des paramètres. Celle-ci est `Get-Help`, qui, comme son nom l'indique, permet d'obtenir de l'aide sur une cmdlet, une fonction… Le prérequis est bien entendu que l'aide soit documentée, sinon aucune information n'apparaît.

La cmdlet `Get-Help` permet par exemple de récupérer :

– la liste des paramètres disponibles,

– le fonctionnement de ces paramètres,

– des exemples d'utilisation de la commande.

Remarque

Nous ne détaillerons pas dans ce chapitre cette cmdlet, car le chapitre L'aide PowerShell, un précieux allié traite déjà le sujet.

2. Get-Member : comprendre ce que contient un objet

2.1 Fonctionnement

Nous avons vu précédemment que pour obtenir des informations sur une cmdlet il fallait utiliser la cmdlet `Get-Help`.

Utilisons donc cette cmdlet pour comprendre le fonctionnement de la cmdlet `Get-Member` :

```
get-help get-member
```

Ci-dessous le retour :

```
PS C:\Users\damien.vanrobaeys> get-help get-member
NAME
    Get-Member

SYNOPSIS
    Gets the properties and methods of objects.

SYNTAX
    Get-Member [[-Name] <System.String[]>] [-Force] [-InputObject
<System.Management.Automation.PSObject>] [-MemberType
{AliasProperty | CodeProperty | Property | NoteProperty |
    ScriptProperty | Properties | PropertySet | Method |
CodeMethod | ScriptMethod | Methods | ParameterizedProperty |
MemberSet | Event | Dynamic | All}] [-Static] [-View {Extended |
    Adapted | Base | All}] [<CommonParameters>]

DESCRIPTION
    The `Get-Member` cmdlet gets the members, the properties and
methods, of objects.
    To specify the object, use the InputObject parameter or pipe
an object to `Get-Member`. To get information about static members,
the members of the class, not of the instance, use the
    Static parameter. To get only certain types of members, such as
NoteProperties, use the MemberType parameter.
```

Cette cmdlet permettra donc de lister tous les éléments (méthodes, propriétés, évènements) inclus dans un objet comme une cmdlet.

L'ajout du paramètre `-Examples` permettra d'obtenir des exemples concrets sur son utilisation :

```
get-help get-member -Examples
```

2.2 Lister uniquement les propriétés

Les propriétés permettront d'obtenir des informations fournies par un objet tel que le nom d'un utilisateur Active Directory ou autre. Pour lister les différentes propriétés disponibles, la commande à utiliser sera la suivante :

```
Get-Member -MemberType Property
```

2.3 Lister uniquement les méthodes

Les méthodes permettront d'effectuer des actions sur un objet, par exemple ajouter un élément dans un fichier XML. Pour lister les différentes méthodes disponibles, la commande à utiliser sera la suivante :

```
Get-Member -MemberType Method
```

3. Get-Command pour trouver la bonne commande

PowerShell contient de nombreuses cmdlets par défaut.

Lorsque vous souhaitez effectuer une action, il est important de trouver la cmdlet qui peut être utile en saisissant, par exemple, un mot-clé. La cmdlet à utiliser sera `Get-Command`.

Dans l'exemple suivant, nous souhaitons savoir comment gérer la partie WMI avec PowerShell. Notre mot est `WMI`, tel que ci-dessous :

```
Get-Command "*wmi*"
```

Ci-dessous le résultat :

```
PS C:\Users\damien.vanrobaeys> Get-Command "*wmi*"

CommandType     Name
-----------     ----
Alias           gwmi -> Get-WmiObject
Alias           iwmi -> Invoke-WmiMethod
Alias           rwmi -> Remove-WmiObject
Alias           swmi -> Set-WmiInstance
Cmdlet          Get-WmiObject
Cmdlet          Invoke-WmiMethod
Cmdlet          Register-WmiEvent
Cmdlet          Remove-WmiObject
Cmdlet          Set-WmiInstance
Application     WMIADAP.exe
Application     WmiApSrv.exe
Application     WMIC.exe
Application     WmiMgmt.msc
Application     WmiPrvSE.exe
```

Nous obtenons différents éléments (alias, cmdlets, applications, fonctions).

Pour n'afficher que les cmdlets, nous utiliserons :

```
Get-Command "*wmi*" -CommandType cmdlet
```

Nous souhaitons maintenant lister les commandes nous permettant de gérer les CSV :

```
Get-Command "*csv*" -CommandType cmdlet
```

```
PS C:\Users\damien.vanrobaeys> Get-Command "*csv*" -CommandType cmdlet

CommandType     Name                                               Version
-----------     ----                                               -------
Cmdlet          ConvertFrom-Csv                                    3.1.0.0
Cmdlet          ConvertTo-Csv                                      3.1.0.0
Cmdlet          Export-Csv                                         3.1.0.0
Cmdlet          Import-Csv                                         3.1.0.0
```

Lors de l'installation d'un module tiers, d'autres cmdlets sont également ajoutées.