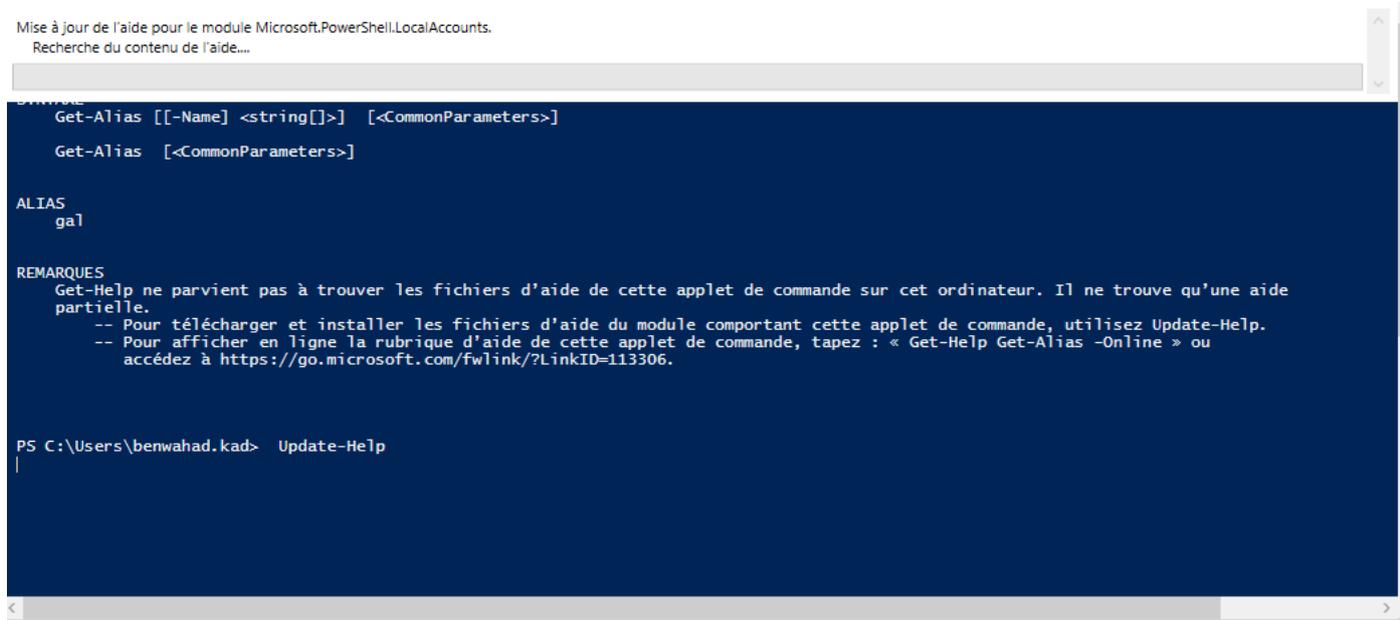


L'installation de windows corp na pas pue etre effectuer a cause de de la securite des ordinateur de windows

2



Get-Help Get-Alias

```
PS C:\Users\benwahad.kad> Get-Help Get-Alias
>>

NOM
    Get-Alias

SYNTAXE
    Get-Alias [[-Name] <string[]>] [<CommonParameters>]

    Get-Alias [<CommonParameters>]

ALIAS
    gal

REMARQUES
    Get-Help ne parvient pas à trouver les fichiers d'aide de cette applet de commande sur cet ordinateur. Il ne trouve qu'une aide partielle.
    -- Pour télécharger et installer les fichiers d'aide du module comportant cette applet de commande, utilisez Update-Help.
    -- Pour afficher en ligne la rubrique d'aide de cette applet de commande, tapez : «Get-Help Get-Alias -Online» ou accédez à https://go.microsoft.com/fwlink/?LinkID=113306.

PS C:\Users\benwahad.kad>
```

Get-Help Get-Alias –Exemples

Pour les exemple

```
PS C:\Users\benwahad.kad> Get-Help Get-Alias -Examples

NOM
    Get-Alias

ALIAS
    gal

REMARQUES
    Get-Help ne parvient pas à trouver les fichiers d'aide de cette applet de commande sur cet ordinateur. Il ne trouve qu'une aide partielle.
    -- Pour télécharger et installer les fichiers d'aide du module comportant cette applet de commande, utilisez Update-Help.
    -- Pour afficher en ligne la rubrique d'aide de cette applet de commande, tapez : «Get-Help Get-Alias -Online» ou accédez à https://go.microsoft.com/fwlink/?LinkID=113306.
```

Afficher tous les alias dont le nom commence par la lettre "g" : **Get-Alias g***

```

PS C:\Users\benwahad.kad> Get-Alias g*
>>
CommandType      Name
-----
Alias             gal -> Get-Alias
Alias             gbp -> Get-PSBreakpoint
Alias             gc -> Get-Content
Alias             gcb -> Get-Clipboard
Alias             gci -> Get-ChildItem
Alias             gcm -> Get-Command
Alias             gcs -> Get-PSCallStack
Alias             gdr -> Get-PSDrive
Alias             ghy -> Get-History
Alias             gi -> Get-Item
Alias             gin -> Get-ComputerInfo
Alias             gjb -> Get-Job
Alias             gl -> Get-Location
Alias             gm -> Get-Member
Alias             gmo -> Get-Module
Alias             gp -> Get-ItemProperty
Alias             gps -> Get-Process
Alias             gpv -> Get-ItemPropertyValue
Alias             group -> Group-Object
Alias             gsn -> Get-PSSession
Alias             gsnp -> Get-PSSnapin
Alias             gsv -> Get-Service
Alias             gtz -> Get-TimeZone
Alias             gu -> Get-Unique
Alias             gv -> Get-Variable
Alias             gwmi -> Get-WmiObject

PS C:\Users\benwahad.kad>

```

Afficher la commande qui correspond à l'alias dont le nom est "sl" : `Get-Alias sl`

```

PS C:\Users\benwahad.kad> Get-Alias sl
>>
CommandType      Name
-----
Alias             sl -> Set-Location

```

Afficher tous les alias dont la définition est "Get-ChildItem" : `Get-Alias | Where-Object {$_.Definition -eq "Get-ChildItem"}`

```

PS C:\Users\benwahad.kad> Get-Alias | Where-Object {$_.Definition -eq "Get-ChildItem"}
>>
CommandType      Name
-----
Alias             dir -> Get-ChildItem
Alias             gci -> Get-ChildItem
Alias             ls -> Get-ChildItem

PS C:\Users\benwahad.kad>

```

pour afficher les informations du volume nommé "C" en utilisant l'exemple 2 de la commande `Get-PSDrive`, : `Get-PSDrive -Name C`

```
PS C:\Users\benwahad.kad> Get-PSDrive -Name C
>>
Name           Used (GB)  Free (GB) Provider      Root           CurrentLocation
-----
C              238,82    207,75  FileSystem    C:\           Users\benwahad.kad

PS C:\Users\benwahad.kad>
```

Pour afficher les méthodes et les propriétés des objets retournés par la commande **Get-Location** :

`Get-Location | Get-Member`

De même, pour afficher les méthodes et les propriétés des objets retournés par la commande `Get-PSDrive` :

`Get-PSDrive | Get-Member`

-----3-----

Afficher le chemin du dossier courant

`Get-Location`

Se déplacer à la racine de la partition C:

`Set-Location C:\`

Afficher la liste des dossiers et fichiers

`Get-ChildItem`

Créer un dossier nommé testPowerShell

`New-Item -ItemType Directory -Name testPowerShell`

Se déplacer dans le dossier c:\testPowerShell

`Set-Location .\testPowerShell`

Créer un dossier nommé testdossier

`New-Item -ItemType Directory -Name testdossier`

Créer un fichier nommé test1.txt, contenant la phrase "Tp PowerShell 1"

Set-Content -Path .\test1.txt -Value "Tp PowerShell 1"

Afficher la liste des dossiers et fichiers

Get-ChildItem

Copier le fichier test1.txt sous le nom test2.txt

Copy-Item -Path .\test1.txt -Destination .\test2.txt

Renommer le fichier test1.txt avec le nom essai1.txt

Rename-Item -Path .\test1.txt -NewName essai1.txt

Copier le fichier essai1.txt dans le dossier testdossier\essai1.txt

Copy-Item -Path .\essai1.txt -Destination .\testdossier\essai1.txt

Afficher la liste des fichiers du dossier et des sous-dossiers de testPowerShell

Get-ChildItem -Recurse

Copier le dossier testdossier (avec ses fichiers) dans un nouveau dossier test2dossier

Copy-Item -Path .\testdossier -Destination .\test2dossier -Recurse

Déplacer le fichier test2.txt dans le dossier testdossier

Move-Item -Path .\test2.txt -Destination .\testdossier

Supprimer le dossier test2dossier (avec ses fichiers)

Remove-Item -Path .\test2dossier -Recurse

Tester l'existence du dossier c:\windows

Test-Path -Path C:\Windows

Afficher le contenu du dossier c:\windows

Get-ChildItem -Path C:\Windows

Afficher la liste des fichiers .exe du dossier c:\windows

```
Get-ChildItem -Path C:\Windows -Filter *.exe
```

4

Affecter à la variable \$loc, le résultat de la commande Get-Location

```
$loc = Get-Location
```

Afficher les propriétés et les méthodes de la variable \$loc

```
$loc | Get-Member
```

Afficher le chemin du dossier courant contenu dans cette variable

```
$loc.Path
```

Afficher les informations sur le disque contenu par cette variable

```
$loc.Drive
```

Afficher les informations sur le 'Provider' contenu par cette variable

```
$loc.ProviderName
```

Affecter à la variable \$lect, le résultat de la commande Get-PSDrive -Name C

```
$lect = Get-PSDrive -Name C
```

Afficher les propriétés et les méthodes de la variable \$lect

```
$lect | Get-Member
```

A partir de la variable \$lect, afficher la description du lecteur C

```
$lect.Description
```

Afficher la taille en octet du volume utilisé

`$lect.Used / 1GB`

Afficher la taille en octet du volume libre

`$lect.Free / 1GB`

Affecter à la variable \$fichier, le résultat de la commande Get-ChildItem
c:\testPowerShell\essai1.txt

`$fichier = Get-ChildItem C:\testPowerShell\essai1.txt`

Afficher les propriétés et les méthodes de la variable \$fichier

`$fichier | Get-Member`

A partir de la variable \$fichier, afficher le nom du fichier

`$fichier.Name`

Afficher la taille en octet du fichier

`$fichier.Length`

Afficher le nom complet du fichier (avec le chemin)

`$fichier.FullName`

Afficher l'extension seule du fichier

`$fichier.Extension`

Afficher la date du dernier accès

`$fichier.LastAccessTime`

A l'aide d'une méthode de la variable \$fichier, copier ce fichier dans un nouveau fichier nommé
C:\TestPowerShell\essai2.txt

`$fichier.CopyTo("C:\TestPowerShell\essai2.txt")`

A partir de la variable \$fichier, supprimer le fichier essai1.txt

```
Remove-Item $fichier
```

Vérifier avec la commande Get-ChildItem

```
Get-ChildItem C:\testPowerShell
```

Lancer notepad.exe et réduire la fenêtre du Bloc-notes

```
Start-Process notepad.exe
```

```
Start-Sleep -Seconds 2
```

```
$notepad = Get-Process notepad
```

```
$notepad.Minimize()
```

Lancer la commande Get-Process et vérifier que le Bloc-notes soit bien dans les processus actifs

```
Get-Process | Where-Object {$_.Name -eq "notepad"}
```

Affecter à la variable \$proc, le résultat de la commande Get-Process notepad

```
$proc = Get-Process notepad
```

Afficher les propriétés et les méthodes de la variable \$proc

```
$proc | Get-Member
```

A partir de la variable \$proc, afficher la description du processus

```
$proc.Description
```

Afficher le chemin d'accès de l'exécutable

```
$proc.Path
```

A partir de la variable \$proc, supprimer (tuer) le processus du Bloc-notes

```
$proc.Kill()
```

Afficher toutes les informations concernant le contrôleur vidéo de votre système

```
Get-WmiObject -Class Win32_VideoController
```

```
# Affecter à la variable $video, le résultat de la commande précédente
```

```
$video = Get-WmiObject -Class Win32_VideoController
```

```
# Afficher les propriétés et les méthodes de la variable $video
```

```
$video | Get-Member
```

```
# A partir de la variable $video, afficher le nom du contrôleur, la version du driver, le mode video (résolution) et le nom du processeur video
```

```
$video | Select-Object Name, DriverVersion, VideoModeDescription, VideoProcessor
```

```
# Afficher les informations concernant le système d'exploitation
```

```
Get-WmiObject -Class Win32_OperatingSystem
```

```
# Affecter à la variable $os, le résultat de la commande précédente
```

```
$os = Get-WmiObject -Class Win32_OperatingSystem
```

```
# A partir de la variable $os, afficher le nom du système, le type d'architecture (32-64 bits), la date d'installation
```

```
$os | Select-Object Caption, OSArchitecture, InstallDate
```

```
# Afficher les informations concernant les disques logiques de votre système
```

```
Get-WmiObject -Class Win32_LogicalDisk
```

```
# Affecter à la variable $vol, le résultat de la commande précédente
```

```
$vol = Get-WmiObject -Class Win32_LogicalDisk
```

```
# A partir de la variable $vol, et pour le premier disque logique seulement, afficher le nom du volume, la taille, l'espace libre, et le système de fichiers
```

```
$vol[0] | Select-Object VolumeName, @{Name="Size(GB)"; Expression={$_.Size / 1GB}},  
@{Name="FreeSpace(GB)"; Expression={$_.FreeSpace / 1GB}}, FileSystem
```