

TD Gestion des commandes sous Powershell

Davia Moujabber – ESI3P

Partie 2 : Aide sur commande

1. Afficher l'aide sur la commande Get-Alias

```
PS C:\Users\davia> Get-Help Get-Alias

NOM
    Get-Alias

SYNTAX
    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.
```

2. Afficher l'aide avec les exemples sur la commande Get-Alias

```
PS C:\Users\davia> 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.
```

3. Afficher tous les alias dont le nom commence par la lettre g

```
PS C:\Users\davia> Get-Alias -name g**
```

CommandType	Name	Version	Source
Alias	gal -> Get-Alias		
Alias	gbp -> Get-PSBreakpoint		
Alias	gc -> Get-Content		
Alias	gcb -> Get-Clipboard	3.1.0.0	Microsoft.PowerShell.Management
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	3.1.0.0	Microsoft.PowerShell.Management
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	3.1.0.0	Microsoft.PowerShell.Management
Alias	gu -> Get-Unique		
Alias	gv -> Get-Variable		
Alias	gwmi -> Get-WmiObject		

4. Afficher la commande qui correspond à l'alias dont le nom est sl

```
PS C:\Users\davia> Get-Alias -name sl
```

CommandType	Name	Version	Source
Alias	sl -> Set-Location		

```
PS C:\Users\davia>
```

5. Afficher tous les alias dont la définition est Get-ChildItem

```
PS C:\Users\davia> Get-Help Get-ChildItem
```

NOM

Get-ChildItem

SYNTAXE

```
Get-ChildItem [[-Path] <string[]>] [[-Filter] <string>] [<CommonParameters>]
Get-ChildItem [[-Filter] <string>] [<CommonParameters>]
```

ALIAS

```
gci
ls
dir
```

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-ChildItem -Online » ou accédez à <https://go.microsoft.com/fwlink/?LinkID=113308>.

Ls : Linux

Dir : DOS

6. A l'aide la commande Get-PSDrive, afficher les informations du volume du nommé C

```
PS C:\Users\davia> get-PSDrive [c]
```

Name	Used (GB)	Free (GB)	Provider	Root	CurrentLocation
C	230,21	49,62	FileSystem	C:\	Users\davia

7. Et 8. Afficher les méthodes et les propriétés des objets retournés par la commande Get-Location et Get-PSDrive

```
PS C:\Users\davia> Get-Location | Get-member
```

TypeName : System.Management.Automation.PathInfo

Name	MemberType	Definition
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
Drive	Property	System.Management.Automation.PSDriveInfo Drive {get;}
Path	Property	string Path {get;}
Provider	Property	System.Management.Automation.ProviderInfo Provider {get;}
ProviderPath	Property	string ProviderPath {get;}

```
PS C:\Users\davia> Get-PSDrive | Get-member
```

TypeName : System.Management.Automation.PSDriveInfo

Name	MemberType	Definition
CompareTo	Method	int CompareTo(System.Management.Automation.PSDriveInfo drive), int CompareTo(System.Object obj),...
Equals	Method	bool Equals(System.Object obj), bool Equals(System.Management.Automation.PSDriveInfo drive)
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
Credential	Property	pscredential Credential {get;}
CurrentLocation	Property	string CurrentLocation {get;set;}
Description	Property	string Description {get;set;}
DisplayRoot	Property	string DisplayRoot {get;}
MaximumSize	Property	System.Nullable[long] MaximumSize {get;}
Name	Property	string Name {get;}
Provider	Property	System.Management.Automation.ProviderInfo Provider {get;}
Root	Property	string Root {get;}
Free	ScriptProperty	System.Object Free {get==# Ensure that this is a FileSystem drive...
Used	ScriptProperty	System.Object Used {get==# Ensure that this is a FileSystem drive...

Partie 3 : Gestion des fichiers et des dossiers

1. Afficher le chemin du dossier courant

```
PS C:\Users\davia> Get-Location
```

Path

C:\Users\davia

2. Se déplacer à la racine de la partition C :

```
PS C:\Users\davia> Set-Location c:\
PS C:\>
```

3. Afficher la liste des dossiers et fichiers

```
PS C:\> ls

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----          11/11/2020         17:05      $WINDOWS.~BT
d-----          20/01/2021         11:00      composer
d-----          07/08/2020          03:24      eSupport
d-----          21/12/2020         10:44      jaspersoftstudio
d-----          07/12/2019         10:14      PerfLogs
d-r-----        20/01/2021         11:03      Program Files
d-r-----        28/12/2020         12:56      Program Files (x86)
d-r-----         02/10/2020         12:16      Users
d-----          24/01/2021          09:25      Windows
d-----          20/01/2021         10:54      xampp
```

4. A cet emplacement, créer un dossier nommé testPowershell

```
PS C:\> New-Item testPowershell -ItemType directory

Répertoire : C:\

Mode                LastWriteTime         Length Name
----                -
d-----          26/01/2021         12:11      testPowershell
```

5. Et 6. Se déplacer dans le dossier c:\testPowershell et créer un dossier nommé testdossier

```
PS C:\> Set-Location c:\testPowerShell
PS C:\testPowerShell> New-Item testdossier -ItemType directory

Répertoire : C:\testPowerShell

Mode                LastWriteTime         Length Name
----                -
d-----          26/01/2021         12:15      testdossier
```

7.et 8.Créer un fichier nommé test1.txt, contenant la phrase « TP Powershell 1 »et afficher la liste des dossiers et fichiers

```
PS C:\testPowerShell> New-Item test1.txt -ItemType file -Value "TP PowerShell 1"
```

Répertoire : C:\testPowerShell

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	26/01/2021 12:17	15	test1.txt

```
PS C:\testPowerShell> dir
```

Répertoire : C:\testPowerShell

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	26/01/2021 12:15		testdossier
-a----	26/01/2021 12:17	15	test1.txt

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

```
PS C:\testPowerShell> Copy-Item .\test1.txt -Destination test2.txt  
PS C:\testPowerShell>
```

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

```
PS C:\testPowerShell> Rename-Item test1.txt -NewName essai1.txt  
PS C:\testPowerShell> ls
```

Répertoire : C:\testPowerShell

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	26/01/2021 12:22		testDossier
-a----	26/01/2021 12:24	15	essai1.txt
-a----	26/01/2021 12:24	15	test2.txt

11. et 12 Copier le fichier essai1.txt dans le dossier testdossier\essai1.txt et afficher la liste des fichiers du dossier et des sous-dossiers de testPwershell

```
PS C:\testPowerShell> Copy-Item essai1.txt -Destination testdossier\essai1.txt
PS C:\testPowerShell> ls
```

Répertoire : C:\testPowerShell

Mode	LastWriteTime	Length	Name
d----	26/01/2021 12:15		testdossier
-a----	26/01/2021 12:17	15	essai1.txt
-a----	26/01/2021 12:27	15	test1.txt
-a----	26/01/2021 12:27	15	test2.txt

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

```
PS C:\testPowerShell> Copy-Item testdossier -Destination test2dossier -Recurse
```

14. Déplacer le fichier test2.txt dans le dossier testdossier

```
PS C:\testPowerShell> Move-Item test2.txt -Destination .\testdossier
PS C:\testPowerShell> ls
```

Répertoire : C:\testPowerShell

Mode	LastWriteTime	Length	Name
d----	26/01/2021 13:43		test2dossier
d----	26/01/2021 14:08		testdossier
-a----	26/01/2021 12:17	15	essai1.txt
-a----	26/01/2021 12:27	15	test1.txt

```
PS C:\testPowerShell> dir -Recurse
```

Répertoire : C:\testPowerShell

Mode	LastWriteTime	Length	Name
d----	26/01/2021 13:43		test2dossier
d----	26/01/2021 14:08		testdossier
-a----	26/01/2021 12:17	15	essai1.txt
-a----	26/01/2021 12:27	15	test1.txt

Répertoire : C:\testPowerShell\test2dossier

Mode	LastWriteTime	Length	Name
d----	26/01/2021 13:43		testdossier
-a----	26/01/2021 12:17	15	essai1.txt

15. Supprimer le dossier test2dossier (avec ses fichiers)

```
PS C:\testPowerShell> Remove-Item test2dossier

Confirmer
L'élément situé à l'emplacement C:\testPowerShell\test2dossier a des enfants et le paramètre Recurse n'a pas été spécifié. Si vous
continuez, tous les enfants seront supprimés avec l'élément. Êtes-vous sûr de vouloir continuer ?
[0] Oui [T] Oui pour tout [N] Non [U] Non pour tout [S] Suspendre [?] Aide (la valeur par défaut est « 0 ») : o
```

16. Tester l'existence du dossier c:\windows

```
PS C:\testPowerShell> Test-Path c:\windows
True
```

17. Afficher le contenu du dossier c:\windows

```
PS C:\testPowerShell> Get-ChildItem C:\Windows

Répertoire : C:\Windows
```

Mode	LastWriteTime	Length	Name
d----	22/08/2013	17:39	ADFS
d----	28/02/2017	15:43	ADWS
d----	22/08/2013	17:39	AppCompat
d----	23/08/2013	00:27	apppatch
d----	22/08/2013	17:39	AppReadiness
d-r-s	01/03/2017	10:30	assembly
d----	22/08/2013	17:39	Boot
d----	22/08/2013	17:39	Branding
d----	03/03/2017	11:23	CbsTemp
d----	22/08/2013	17:39	Cursors
d----	08/03/2017	10:12	debug
d-r--	22/08/2013	17:39	DesktopTileResources
d----	22/08/2013	17:39	diagnostics
d----	22/08/2013	17:46	DigitalLocker
d---s	22/08/2013	17:39	Downloaded Program Files
d----	22/08/2013	17:39	drivers
d----	22/08/2013	16:53	en-US
d-r-s	01/03/2017	10:30	Fonts
d----	23/08/2013	00:27	fr-FR
d----	22/08/2013	17:39	Globalization
d----	23/08/2013	00:27	Help
d----	22/08/2013	17:46	IME
d-r--	22/08/2013	17:39	ImmersiveControlPanel
d----	08/03/2017	10:17	Inf
d----	22/08/2013	17:39	InputMethod
d----	22/08/2013	17:39	L2Schemas
d----	22/08/2013	17:39	LiveKernelReports
d----	28/02/2017	15:09	Logs

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

```
PS C:\testPowerShell> Get-Childitem C:\Windows\*.exe
```

Répertoire : C:\Windows

Mode		LastWriteTime	Length	Name
-a---	22/08/2013	13:21	56832	bfsvc.exe
-a---	22/08/2013	14:39	2328880	explorer.exe
-a---	22/08/2013	12:22	973312	HelpPane.exe
-a---	22/08/2013	13:32	17408	hh.exe
-a---	22/08/2013	12:57	151552	regedit.exe
-a---	22/08/2013	13:03	125952	splwow64.exe
-a---	22/08/2013	05:57	10752	winhlp32.exe
-a---	22/08/2013	13:22	10752	write.exe

Partie 4 : Accès aux propriétés et aux méthodes d'un objet

1 et 2 Affecter à la variable \$loc, le résultat de la commande Get-Location et afficher les propriétés et les méthodes de la variable \$loc

```
PS C:\windows> $loc = Get-Location
PS C:\windows> $loc | Get-Member

TypeName : System.Management.Automation.PathInfo

Name      MemberType Definition
-----
Equals     Method      bool Equals(System.Object obj)
GetHashCode Method      int GetHashCode()
GetType    Method      type GetType()
ToString   Method      string ToString()
Drive      Property    System.Management.Automation.PSDriveInfo Drive {get;}
Path       Property    string Path {get;}
Provider   Property    System.Management.Automation.ProviderInfo Provider {get;}
ProviderPath Property    string ProviderPath {get;}
```

3, 4 , 5, 6 et 7

Afficher le chemin du dossier courant contenu dans cette variable.

Afficher les informations sur le disque contenu par cette variable.

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

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

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


```

PS C:\testPowershell> $loc.path
C:\testPowershell
PS C:\testPowershell> $loc.drive

Name            Used (GB)    Free (GB) Provider      Root
-----
C                230,43      49,40  FileSystem    C:\
CurrentLocation
-----
testPowershell

PS C:\testPowershell> $loc.Provider

Name            Capabilities          Drives
-----
FileSystem      Filter, ShouldProcess, Credentials {C, D}

PS C:\testPowershell> $lect = Get-PSDrive -Name C
PS C:\testPowershell> $lect | Get-Member

TypeName : System.Management.Automation.PSDriveInfo

Name            MemberType      Definition
-----
CompareTo       Method          int CompareTo(System.Management.Automation.PSDriveInfo drive), int CompareTo(System.Object obj),...
Equals          Method          bool Equals(System.Object obj), bool Equals(System.Management.Automation.PSDriveInfo drive)
GetHashCode     Method          int GetHashCode()
GetType        Method          type GetType()
ToString        Method          string ToString()
Credential       Property        pscredential Credential {get;}
CurrentLocation Property        string CurrentLocation {get;set;}
Description     Property        string Description {get;set;}
DisplayRoot     Property        string DisplayRoot {get;}
MaximumSize     Property        System.Nullable[long] MaximumSize {get;}
Name            Property        string Name {get;}
Provider        Property        System.Management.Automation.ProviderInfo Provider {get;}
Root            Property        string Root {get;}
Free            ScriptProperty System.Object Free {get==## Ensure that this is a FileSystem drive...
Used            ScriptProperty System.Object Used {get==## Ensure that this is a FileSystem drive...

```

8. A partir de la variable \$lect, afficher la description du lecteur C, afficher la taille en octet du volume libre et afficher la taille en octet du volume utilisé

```

PS C:\testPowershell> $lect.Description
OS
PS C:\testPowershell> $lect.Free
53048578048
PS C:\testPowershell> $lect.used/1GB
230,431945800781
PS C:\testPowershell>

```

9. Et 10. Affecter à la variable \$fichier, le résultat de la commande Get-ChildItem c:\testPowerShell\essai1.txt et Afficher les propriétés et les méthodes de la variable \$fichier

```

PS C:\testPowershell> $fichier = Get-ChildItem c:\testPowershell\essai1.txt
PS C:\testPowershell> $fichier | Get-member

TypeName : System.IO.FileInfo

Name            MemberType      Definition
-----
LinkType        CodeProperty    System.String LinkType{get=GetLinkType;}
Mode            CodeProperty    System.String Mode{get=Mode;}
Target          CodeProperty    System.Collections.Generic.IEnumerable`1[[System.String, mscorlib, Version=4.0.0.0, Cu...
AppendText      Method          System.IO.StreamWriter AppendText()
CopyTo          Method          System.IO.FileInfo CopyTo(string destFileName), System.IO.FileInfo CopyTo(string destF...
Create          Method          System.IO.FileStream Create()
CreateObjRef    Method          System.Runtime.Remoting.ObjRef CreateObjRef(type requestedType)
CreateText      Method          System.IO.StreamWriter CreateText()
Decrypt         Method          void Decrypt()
Delete          Method          void Delete()
Encrypt         Method          void Encrypt()

```

11. A partir de la variable \$fichier, afficher le nom du fichier, Afficher le nom complet du fichier (avec le chemin), Afficher l'extension seule du fichier, Afficher la date du dernier accès

```
PS C:\testPowershell> $fichier.name
essai1.txt
PS C:\testPowershell> $fichier.fullname
C:\testPowershell\essai1.txt
PS C:\testPowershell> $fichier.Extension
.txt
PS C:\testPowershell> $fichier.LastAccessTime

mardi 26 janvier 2021 13:31:31
```

Afficher la taille en octet du fichier

```
PS C:\testPowerShell> $fichier.Length
15
PS C:\testPowerShell>
```

12, 13 et 14 A l'aide d'une méthode de la variable \$fichier, copier ce fichier dans un nouveau fichier nommé C:\TestPowerShell\essai2.txt, A partir de la variable \$fichier, supprimer le fichier essai1.txt et Vérifier avec la commande Get-ChildItem

```
PS C:\testPowershell> Copy-Item $fichier -Destination c:\testpowershell\essai2.txt
PS C:\testPowershell> Remove-Item $fichier
PS C:\testPowershell> Get-ChildItem
```

```
Répertoire : C:\testPowershell

Mode                LastWriteTime         Length Name
----                -
d-----         26/01/2021    14:08             testdossier
-a-----         26/01/2021    12:17          15 essai2.txt
-a-----         26/01/2021    12:27          15 test1.txt
```

15.et 16 Lancer notepad.exe et Get-Process

```
PS C:\testPowershell> notepad.exe
PS C:\testPowershell> Get-Process
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
496	28	23116	28776	2,25	1016	1	ApplicationFrameHost
161	8	1696	524		17872	0	AppVShNotify
161	9	1912	552	0,02	17884	1	AppVShNotify
280	13	3588	2304		5044	0	AsusLinkNear
59	5	768	1176		5092	0	AsusLinkNearExt
295	14	7668	8600		5020	0	AsusLinkRemote
186	10	2292	4512		3816	0	AsusOptimization
298	14	2820	3412	0,81	7996	1	AsusOptimizationStartupTask
188	15	2380	4152	0,30	17684	1	AsusOSD
429	18	8832	14408		5100	0	AsusSoftwareManager
487	22	33508	9196	0,41	16376	1	AsusSoftwareManagerAgent
2341	14	4384	9548		5028	0	AsusSystemAnalysis
144	8	1336	1060		3996	0	AsusSystemDiagnosis
959	22	96712	34508	3 902,53	24348	0	audiodg
139	7	2224	4764	0,44	17708	1	CompPkgSrv
97	7	6212	520		3972	0	conhost
128	9	6508	860		7004	0	conhost
97	7	6312	556	0,05	14904	1	conhost
273	14	6572	15560	25,50	16648	1	conhost
711	50	29052	1240	1,00	13652	1	Cortana
911	28	2028	2308		892	0	csrss
992	29	2532	3848		992	1	csrss
703	20	53376	24876	66,31	4640	1	ctfmon
333	15	3704	5900		2536	0	dasHost
84	6	912	672		4824	0	dasHost
617	47	17476	26304	268,70	7108	1	devcpp
143	9	1980	6536	0,58	8296	1	dllhost
203	17	3676	5500		11188	0	dllhost
249	17	4332	7968	0,69	16020	1	dllhost

17. et 18 Affecter à la variable \$proc, le résultat de la commande Get-Process notepad et afficher les propriétés et les méthodes de la variable \$proc

```
PS C:\testPowershell> $proc = Get-Process notepad
PS C:\testPowershell> $proc | Get-member
```

```

    TypeName : System.Diagnostics.Process

Name      MemberType Definition
-----
Handles   AliasProperty Handles = Handlecount
Name       AliasProperty Name = ProcessName
NPM        AliasProperty NPM = NonpagedSystemMemorySize64
PM         AliasProperty PM = PagedMemorySize64
SI         AliasProperty SI = SessionId
VM         AliasProperty VM = VirtualMemorySize64
WS         AliasProperty WS = WorkingSet64
Disposed   Event      System.EventHandler Disposed(System.Object, System.EventArgs)
ErrorDataReceived Event      System.Diagnostics.DataReceivedEventHandler ErrorDataReceived(System.Object, System.D...
Exited     Event      System.EventHandler Exited(System.Object, System.EventArgs)
OutputDataReceived Event      System.Diagnostics.DataReceivedEventHandler OutputDataReceived(System.Object, System....
BeginErrorReadLine Method     void BeginErrorReadLine()
BeginOutputReadLine Method     void BeginOutputReadLine()
CancelErrorRead Method     void CancelErrorRead()
CancelOutputRead Method     void CancelOutputRead()
Close      Method     void Close()
CloseMainWindow Method     bool CloseMainWindow()
CreateObjRef Method     System.Runtime.Remoting.ObjRef CreateObjRef(type requestedType)
Dispose    Method     void Dispose(), void IDisposable.Dispose()
Equals     Method     bool Equals(System.Object obj)
GetHashCode Method     int GetHashCode()
GetLifetimeService Method     System.Object GetLifetimeService()

```

19. A partir de la variable \$proc, afficher la description du processus, Afficher le chemin d'accès de l'exécutable

```
PS C:\testPowershell> $proc.Description
Bloc-notes
PS C:\testPowershell> $proc.Path
C:\WINDOWS\system32\notepad.exe
```

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

```
PS C:\testPowershell> $proc.kill()
PS C:\testPowershell>
```

Partie 5 : Accès aux informations du système

1. Afficher toutes les informations concernant le contrôleur vidéo de votre système

```
PS C:\testPowershell> Get-WmiObject win32_videocontroller

GENUS           : 2
CLASS           : Win32_VideoController
SUPERCLASS      : CIM_PCVideoController
DYNASTY         : CIM_ManagedSystemElement
RELPATH        : Win32_VideoController.DeviceID="VideoController1"
PROPERTY_COUNT  : 59
DERIVATION      : {CIM_PCVideoController, CIM_VideoController, CIM_Controller, CIM_LogicalDevice...}
SERVER         : LAPTOP-FLAM4A86
NAMESPACE      : root\cimv2
PATH           : \\LAPTOP-FLAM4A86\root\cimv2:Win32_VideoController.DeviceID="VideoController1"
AcceleratorCapabilities :
AdapterCompatibility   : Intel Corporation
AdapterDACType        : Internal
AdapterRAM            : 1073741824
Availability           : 3
CapabilityDescriptions :
Caption               : Intel(R) UHD Graphics
ColorTableEntries     :
ConfigManagerErrorCode : 0
ConfigManagerUserConfig : False
CreationClassName     : Win32_VideoController
CurrentBitsPerPixel    : 32
CurrentHorizontalResolution : 1920
CurrentNumberOfColors   : 4294967296
CurrentNumberOfColumns  : 0
CurrentNumberOfRows     : 0
CurrentRefreshRate      : 60
CurrentScanMode         : 4
CurrentVerticalResolution : 1080
Description           : Intel(R) UHD Graphics
DeviceID              : VideoController1
DeviceSpecificPens      :
DitherType             : 0
DriverDate             : 20200205000000.000000-000
DriverVersion          : 26.20.100.7870
```

2. et 3. Affecter à la variable \$video, le résultat de la commande précédente et Afficher les propriétés et les méthodes de la variable \$video

```

PS C:\testPowershell> $video = Get-WmiObject win32_videocontroller
PS C:\testPowershell> $video | Get-member

    TypeName : System.Management.ManagementObject#root\cimv2\Win32_VideoController

Name           MemberType Definition
-----
PSComputerName AliasProperty PSComputerName = __SERVER
Reset          Method      System.Management.ManagementBaseObject Reset()
SetPowerState  Method      System.Management.ManagementBaseObject SetPowerState(System.UInt16 PowerState, Syste...
AcceleratorCapabilities Property    uint16[] AcceleratorCapabilities {get;set;}
AdapterCompatibility Property    string AdapterCompatibility {get;set;}
AdapterDACType Property    string AdapterDACType {get;set;}
AdapterRAM     Property    uint32 AdapterRAM {get;set;}
Availability    Property    uint16 Availability {get;set;}
CapabilityDescriptions Property    string[] CapabilityDescriptions {get;set;}
Caption         Property    string Caption {get;set;}
ColorTableEntries Property    uint32 ColorTableEntries {get;set;}
ConfigManagerErrorCode Property    uint32 ConfigManagerErrorCode {get;set;}
ConfigManagerUserConfig Property    bool ConfigManagerUserConfig {get;set;}
CreationClassName Property    string CreationClassName {get;set;}
CurrentBitsPerPixel Property    uint32 CurrentBitsPerPixel {get;set;}
CurrentHorizontalResolution Property    uint32 CurrentHorizontalResolution {get;set;}
CurrentNumberOfColors Property    uint64 CurrentNumberOfColors {get;set;}
CurrentNumberOfColumns Property    uint32 CurrentNumberOfColumns {get;set;}
CurrentNumberOfRows Property    uint32 CurrentNumberOfRows {get;set;}
CurrentRefreshRate Property    uint32 CurrentRefreshRate {get;set;}
CurrentScanMode Property    uint16 CurrentScanMode {get;set;}
CurrentVerticalResolution Property    uint32 CurrentVerticalResolution {get;set;}
Description     Property    string Description {get;set;}
DeviceID        Property    string DeviceID {get;set;}
DeviceSpecificPens Property    uint32 DeviceSpecificPens {get;set;}
DitherType      Property    uint32 DitherType {get;set;}

```

4. A partir de la variable \$video, afficher le nom du contrôleur, la version du driver, Le mode video (résolution)

```

PS C:\testPowershell> $video.name
Intel(R) UHD Graphics
PS C:\testPowershell> $video.DriverVersion
26.20.100.7870
PS C:\testPowershell> $video.VideoModeDescription
1920 x 1080 x 4294967296 couleurs

```

Le nom du processeur video

Afficher les informations concernant le système

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

Afficher le nom du système

Le type d'architecture (32-64 bits).

La date d'installation

Afficher les informations concernant les disques logiques de votre système

```

PS C:\testPowershell> $video.VideoProcessor
Intel(R) UHD Graphics Family
PS C:\testPowershell> Get-WmiObject win32_operatingsystem

SystemDirectory : C:\WINDOWS\system32
Organization    :
BuildNumber     : 19041
RegisteredUser  : davia.dav@hotmail.fr
SerialNumber    : 00325-81834-94442-AAOEM
Version        : 10.0.19041

PS C:\testPowershell> $os = Get-WmiObject win32_operatingsystem
PS C:\testPowershell> $os.name
Microsoft Windows 10 Famille|C:\WINDOWS|\Device\Harddisk0\Partition3
PS C:\testPowershell> $os.OSArchitecture
64 bits
PS C:\testPowershell> $os.InstallDate
20201002112030.000000+060
PS C:\testPowershell> Get-WmiObject win32_logicaldisk

DeviceID       : C:
DriveType      : 3
ProviderName   :
FreeSpace      : 53030318080
Size           : 300472930304
VolumeName     : OS

DeviceID       : D:
DriveType      : 3
ProviderName   :
FreeSpace      : 209610129408
Size           : 209714147328
VolumeName     : Data

```

9. et 10 Affecter à la variable \$vol, le résultat de la commande précédente, Afficher le nom du volume, Afficher l'espace libre, Afficher le système de fichiers

```

PS C:\testPowershell> $vol = Get-WmiObject win32_logicaldisk
PS C:\testPowershell> $vol[0].name
C:
PS C:\testPowershell> $vol[0].size/1GB
279,837223052979
PS C:\testPowershell> $vol[0].filesystem
NTFS
PS C:\testPowershell>

```