TP2 : PowerShell Options avancés

Sommaire

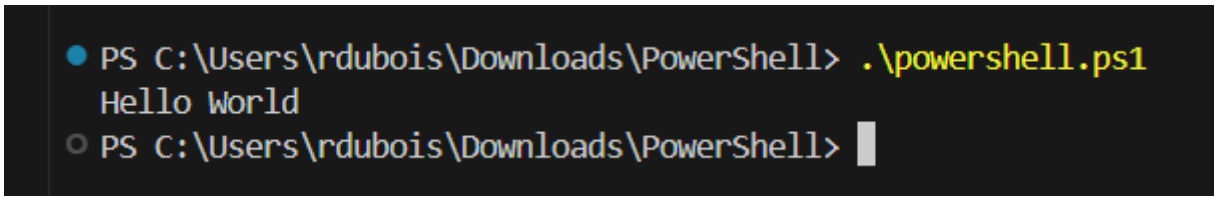
Table des matières

1. Commentaires 2
2. Variables 2
3. Get-Module 3
4. Clear-host 4
5. Write-Host vs Echo 5
6. Read-host 7
7. Pipes 8
8. Format-List 8
9. Opérateur de comparaison 10
10. Where-Object 11
11. Opérateurs de redirection 13
12. Enumération 15
13. Format-Table 16

1. Commentaires

Les commentaires correspondent à la partie du code qui ne seras pas exécuté mais qui sera visible uniquement dans le code, par exemple pour indiquer l'opération que va effectuer un code, une fonction…





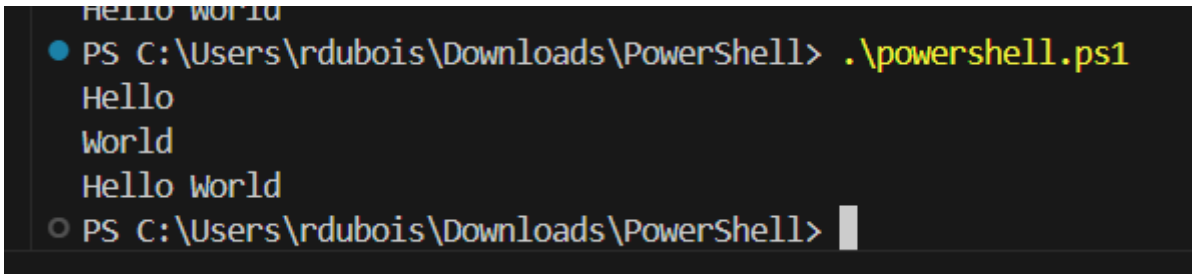
Comme nous pouvons l'observer après avoir exécuté le script, uniquement la partie après le commentaire s'est exécuté. Dans Visual Studio Code et PowerShell ISE, les commentaires sont représentés en vert.

2. Variables

Les variables permettent d'indiquer des informations qui pourront être réutilisées par la suite dans le code.

Voici un exemple de variables effectué sur Visual Studio Code et 3 variables (une variable contenante « Hello », une autre contenant « World » et une autre ayant pour but de concaténer les deux variables avec un espace au milieu) :

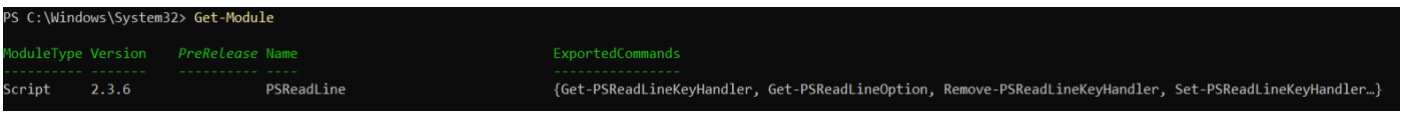
```powershell
$mavariable1 = "Hello"
$mavariable2 = "World"
# Cette variable permet de concaténer les deux premières variables
$mavariable3 = $mavariable1 + " " + $mavariable2

# Affichage des différentes variables dans le terminal
Write-Host $mavariable1
Write-Host $mavariable2
Write-Host $mavariable3
```



Les Write-Host tant qu'à eux ont pour but d'afficher le contenu des variables ayant été défini précédemment au début du code PowerShell.

3. Get-Module

La commande Get-Module permet d'afficher les modules disponibles dans PowerShell.

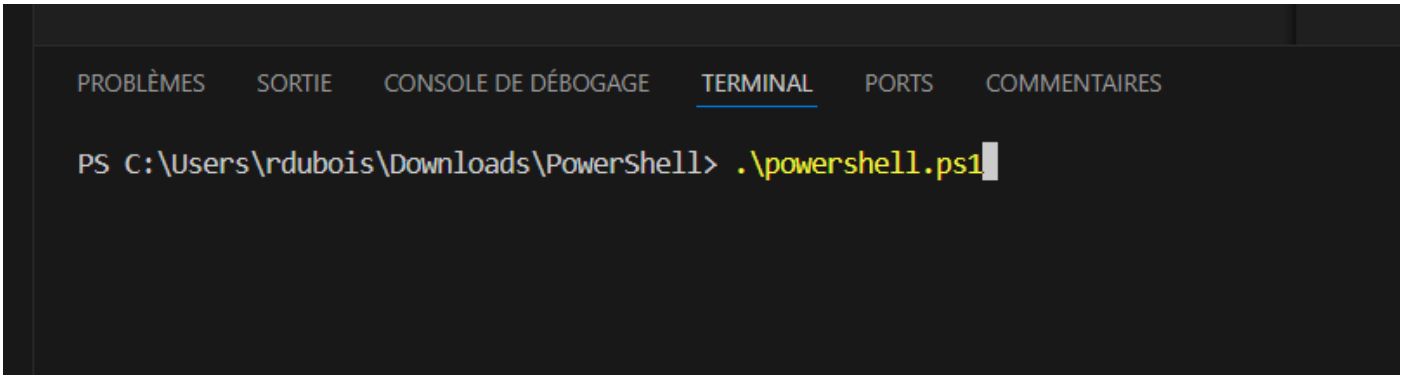


4. Clear-host

La commande clear-host a elle pour but de supprimer ce qui se trouve dans le terminal, il peut être placé avant ou après certaines commandes.

Exemple avec le script de l'exercice 2, avec le clear-host à la fin du script :

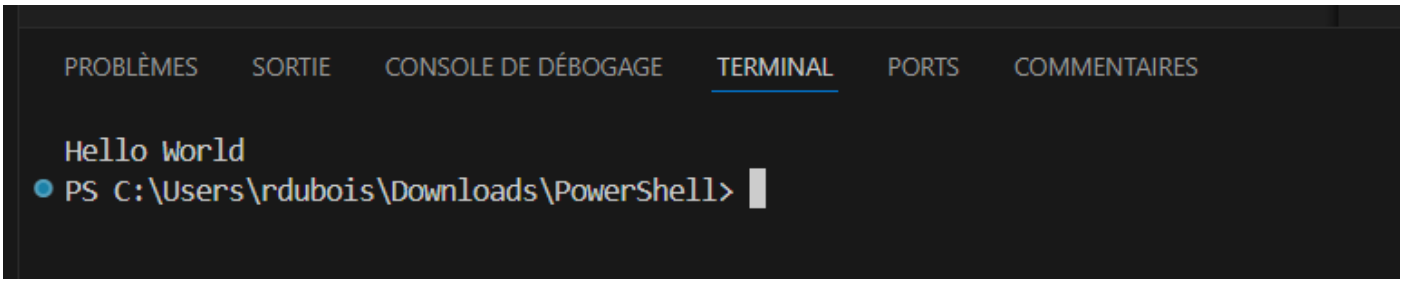
```powershell
$mavariable1 = "Hello"
$mavariable2 = "World"
# Cette variable permet de concaténer les deux premières variables
$mavariable3 = $mavariable1 + " " + $mavariable2

# Affichage des différentes variables dans le terminal
Write-Host $mavariable1
Write-Host $mavariable2
Write-Host $mavariable3

clear-host
```



Après l'exécution du script, rien ne s'affiche, c'est normal car le contenu du terminal a été effacé après avoir exécuté les commandes précédentes.

Si cependant, on décide de placer le clear-host avant le dernier write-host, on pourra observer ceci :

```powershell
$mavariable1 = "Hello"
$mavariable2 = "World"
# Cette variable permet de concaténer les deux premières variables
$mavariable3 = $mavariable1 + " " + $mavariable2

# Affichage des différentes variables dans le terminal
Write-Host $mavariable1
Write-Host $mavariable2
# Efface le terminal
clear-host
Write-Host $mavariable3
```



Ici, on peut observer que la dernière variable s'est affichée dans le terminal comme demandé.

5. Write-Host vs Echo

La commande Write-Host est une commande PowerShell, tandis que la commande echo est une commande DoS, qui elle aussi fonctionne sur PowerShell, voici un exemple :

```powershell
$mavariable1 = "Hello"
$mavariable2 = "World"
# Cette variable permet de concaténer les deux premières variables
$mavariable3 = $mavariable1 + " " + $mavariable2

clear-host

echo "Test avec echo"
echo "Ma variable 1 : $mavariable1"
echo "Ma variable 1 : $mavariable2"
echo "Ma variable 1 : $mavariable3"

Write-Host "Test avec echo"
Write-Host "Ma variable 1 : $mavariable1"
Write-Host "Ma variable 1 : $mavariable2"
Write-Host "Ma variable 1 : $mavariable3"
```

```
Test avec echo
Ma variable 1 : Hello
Ma variable 1 : World
Ma variable 1 : Hello World
Test avec echo
Ma variable 1 : Hello
Ma variable 1 : World
Ma variable 1 : Hello World
PS C:\Users\rdubois\Downloads\PowerShell>
```

Ici, on voit qu'il n'y a aucune différence entre les deux, tout simplement car PowerShell est capable d'exécuter des commandes DOS.

6. Read-host

La commande Read-host permet de demander à l'utilisateur de rentrer une information, par exemple avec « Entrez votre âge » l'utilisateur sera amené à répondre dans le terminal, exemple :




```
PS C:\Users\rdubois\Downloads\PowerShell> .\powershell.ps1
Entrez votre âge:
```

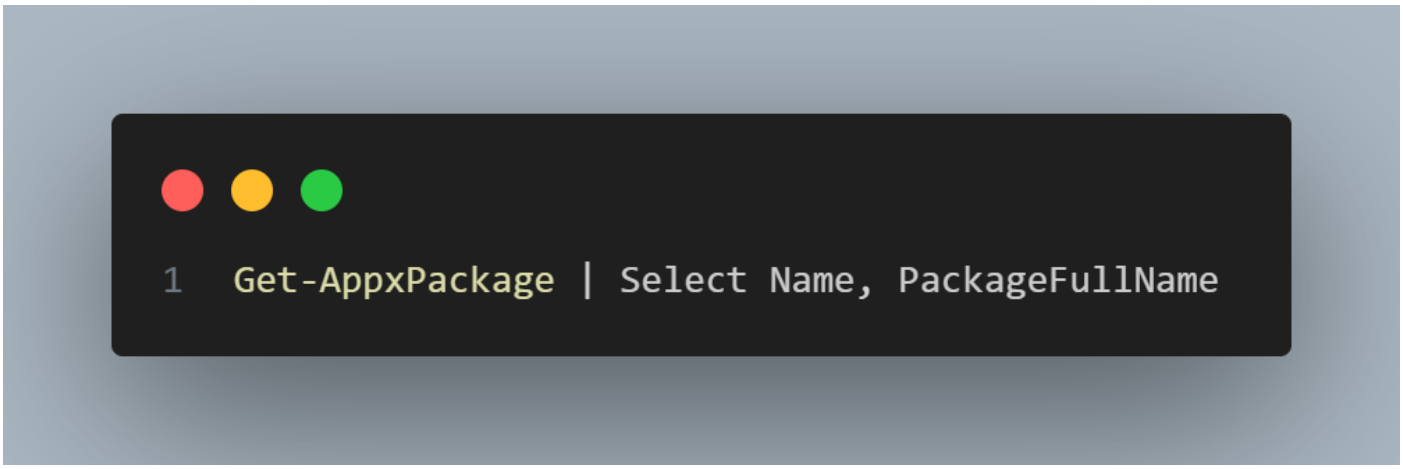
Une fois arrivé ici, il suffit d'insérer la valeur qu'on souhaite

```
PS C:\Users\rdubois\Downloads\PowerShell> .\powershell.ps1
Entrez votre âge: 19
19
PS C:\Users\rdubois\Downloads\PowerShell>
```

Le terminal répond la valeur qui a été inséré précédemment.

7. Pipes

Dans cet exemple, nous allons prendre la commande Get-AppxPackage permet d'afficher les applications « appx » installées. Cependant, pour afficher uniquement les informations nécessaires, on va y ajouter la une pipe avec la commande « Select » qui va nous permettre de sélectionner uniquement le nom de l'application avec le nom complet du paquet, exemple :

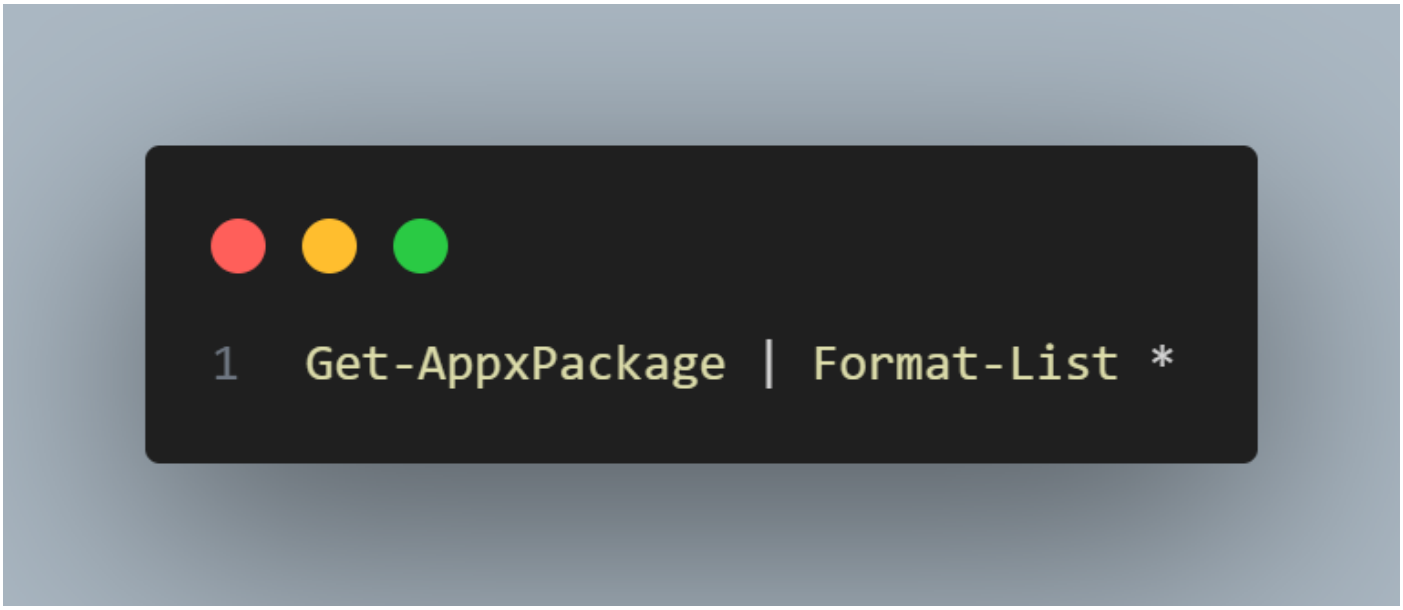



```
PS C:\Users\rdubois\Downloads\PowerShell> .\powershell.ps1

Name                                PackageFullName
----                                ---------------
Microsoft.UI.Xaml.CBS               Microsoft.UI.Xaml.CBS_9.2311.10002.0_x64__8wekyb3d8bbwe
1527c705-839a-4832-9118-54d4Bd6a0c89 1527c705-839a-4832-9118-54d4Bd6a0c89_10.0.19640.1000_neutral_neutral_cw5n1h2txyewy
c5e2524a-ea46-4f67-841f-6a9465d9d515 c5e2524a-ea46-4f67-841f-6a9465d9d515_10.0.26100.1_neutral_neutral_cw5n1h2txyewy
E2A4F912-2574-4A75-9BB0-0D023378592B E2A4F912-2574-4A75-9BB0-0D023378592B_10.0.19640.1000_neutral_neutral_cw5n1h2txyewy
F46D4000-FD22-4DB4-AC8E-4E1DDDE828FE F46D4000-FD22-4DB4-AC8E-4E1DDDE828FE_10.0.26100.1_neutral_neutral_cw5n1h2txyewy
Microsoft.AAD.BrokerPlugin          Microsoft.AAD.BrokerPlugin_1000.19580.1000.0_neutral_neutral_cw5n1h2txyewy
Microsoft.AccountsControl           Microsoft.AccountsControl_10.0.26100.1_neutral__cw5n1h2txyewy
Microsoft.AsyncTextService          Microsoft.AsyncTextService_10.0.26100.1_neutral__8wekyb3d8bbwe
Microsoft.BioEnrollment             Microsoft.BioEnrollment_10.0.19587.1000_neutral__cw5n1h2txyewy
Microsoft.CredDialogHost            Microsoft.CredDialogHost_10.0.19595.1001_neutral__cw5n1h2txyewy
Microsoft.LockApp                   Microsoft.LockApp_10.0.26100.1301_neutral__cw5n1h2txyewy
Microsoft.MicrosoftEdgeDevToolsClient Microsoft.MicrosoftEdgeDevToolsClient_1000.25128.1000.0_neutral_neutral_8wekyb3d8bbwe
Microsoft.Win32WebViewHost          Microsoft.Win32WebViewHost_10.0.26100.1_neutral_neutral_cw5n1h2txyewy
Microsoft.Windows.Apprep.ChxApp     Microsoft.Windows.Apprep.ChxApp_1000.25128.1000.0_neutral_neutral_cw5n1h2txyewy
Microsoft.Windows.AssignedAccessLockApp Microsoft.Windows.AssignedAccessLockApp_1000.25128.1000.0_neutral_neutral_cw5n1h2txyewy
```

Comme nous pouvons l'observer, nous obtenons un tableau avec uniquement les éléments que nous avons sélectionner grâce à la commande « Select »

8. Format-List

Dans cet exemple, nous allons utiliser la commande Get-AppxPackage avec une pipe pour ainsi y exécuter la commande Format-List * qui permettra d'afficher chaque propriété sur une ligne distincte, exemple :



Exemple pour l'appx de Microsoft Teams :

```
Name                   : MSTeams
Publisher              : CN=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=Washington, C=US
PublisherId            : 8wekyb3d8bbwe
Architecture           : X64
ResourceId             :
Version                : 25044.2208.3471.2155
PackageFamilyName      : MSTeams_8wekyb3d8bbwe
PackageFullName        : MSTeams_25044.2208.3471.2155_x64__8wekyb3d8bbwe
InstallLocation        : C:\Program Files\WindowsApps\MSTeams_25044.2208.3471.2155_x64__8wekyb3d8bbwe
IsFramework            : False
PackageUserInformation : {}
IsResourcePackage      : False
IsBundle               : False
IsDevelopmentMode      : False
NonRemovable           : False
Dependencies           : {}
IsPartiallyStaged      : False
SignatureKind          : Developer
Status                 : Ok

PS C:\Users\rdubois\Desktop\PowerShell>
```

Chaque propriété sont affichés sur une ligne distincte.

9. Opérateur de comparaison

Dans notre exemple, nous allons créer un code PowerShell qui va comparer si deux nombres sont égaux ou non grâce à l'opération de comparaison « -eq » (pour equal). Si les nombre sont égaux, le terminal devra en sorti indiquer qu'ils sont égaux, sinon, il devra indiquer qu'ils sont différents :

```powershell
# Définition de deux variables numériques
$nombre1 = 5
$nombre2 = 5

# Comparaison des variables avec l'opérateur -eq
if ($nombre1 -eq $nombre2) {
    Write-Host "Les deux nombres sont égaux."
} else {
    Write-Host "Les deux nombres sont différents."
}
```

Exemple avec 5 en nombre1 et 5 en nombre2 :

```
Les deux nombres sont égaux.
PS C:\Users\rdubois\Desktop\PowerShell>
```

Exemple avec 5 en nombre1 et 6 en nombre2 :

```
Les deux nombres sont différents.
PS C:\Users\rdubois\Desktop\PowerShell>
```

10. Where-Object

La commande Where-Object permet de cibler une valeur, par exemple pour la commande Get-Service, il est possible de cibler les services en cours d'exécution, exemple :



```
# Tous les services seront affichés
Get-Service

# Seulement les services en cours d'exécution seront affichés
Get-Service | Where-Object -Property status -EQ 'Running'
```

Résultat sans la commande « Where-Object » :

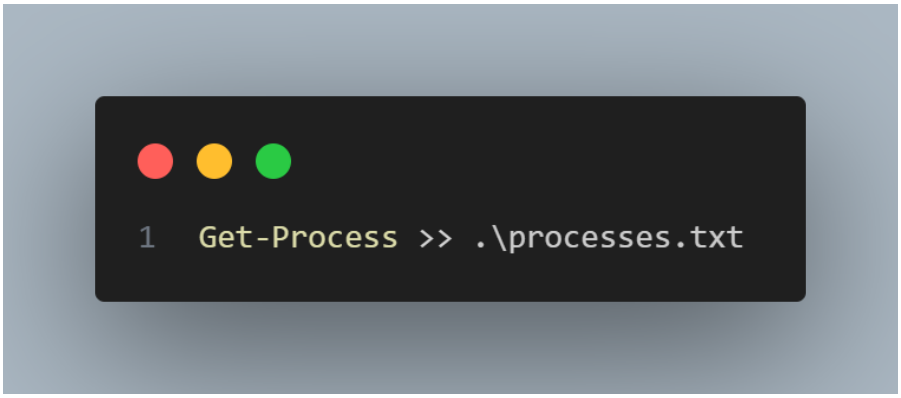
```
Stopped  ApxSvc             Service proxy de périphérique audio v…
Stopped  AssignedAccessMan… Service AssignedAccessManager
Running  AudioEndpointBuil… Générateur de points de terminaison d…
Running  Audiosrv           Audio Windows
Stopped  autotimesvc        Heure cellulaire
Stopped  AxInstSV           Programme d'installation ActiveX (AxI…
Stopped  BcastDVRUserServi… Service utilisateur de diffusion et G…
Running  BDESVC             Service de chiffrement de lecteur Bit…
Running  BFE                Moteur de filtrage de base
Stopped  BITS               Service de transfert intelligent en a…
Stopped  BluetoothUserServ… Service de support des utilisateurs d…
Running  BrokerInfrastruct… Service d'infrastructure des tâches e…
Stopped  BTAGService        Service de passerelle audio Bluetooth
Stopped  BthAvctpSvc        Service AVCTP
Running  bthserv            Service de prise en charge Bluetooth
Running  camsvc             Service Gestionnaire d'accès aux fonc…
Stopped  CaptureService_36… CaptureService_36a8567
Running  cbdhsvc_36a8567    Service utilisateur du Presse-papiers…
Running  CDPSvc             Service de plateforme des appareils c…
Running  CDPUserSvc_36a8567 Service pour utilisateur de plateform…
Stopped  CertPropSvc        Propagation du certificat
Running  ClickToRunSvc      Microsoft Office Click-to-Run Service
Stopped  ClipSVC            Service de licences de client (ClipSV…
Stopped  CloudBackupRestor… Service de restauration et de sauvega…
Stopped  cloudidsvc         Service d'identité de Microsoft Cloud
Running  CmService          Service du gestionnaire de conteneurs
Stopped  COMSysApp          Application système COM+
Stopped  ConsentUxUserSvc_… Service utilisateur ConsentUX_36a8567
Running  CoreMessagingRegi… CoreMessaging
Running  cphs               Intel(R) Content Protection HECI Serv…
Running  cplspcon           Intel(R) Content Protection HDCP Serv…
Stopped  CredentialEnrollm… CredentialEnrollmentManagerUserSvc_36…
Running  CryptSvc           Services de chiffrement
```

Résultat avec la commande « Where-Object » :

```
Running  PlugPlay           Plug-and-Play
Running  Power              Alimentation
Running  ProfSvc            Service de profil utilisateur
Running  ProtonVPN Service  ProtonVPN Service
Running  RasMan             Gestionnaire des connexions d'accès à…
Running  RmSvc              Service de gestion radio
Running  RpcEptMapper       Mappeur de point de terminaison RPC
Running  RpcSs              Appel de procédure distante (RPC)
Running  RstMwService       Intel(R) Storage Middleware Service
Running  RtkAudioUniversal… Realtek Audio Universal Service
Running  SamSs              Gestionnaire de comptes de sécurité
Running  Schedule           Planificateur de tâches
Running  SecurityHealthSer… Service Sécurité Windows
Running  SENS               Service de notification d'événements …
Running  SensorService      Service de capteur
Running  SharedAccess       Partage de connexion Internet (ICS)
Running  ShellHWDetection   Détection matériel noyau
Running  Spooler            Spouleur d'impression
Running  SSDPSRV            Découverte SSDP
Running  SstpSvc            Service SSTP (Secure Socket Tunneling…
Running  StateRepository    Service State Repository (StateReposi…
Running  StorSvc            Service de stockage
Running  SysMain            SysMain
Running  SystemEventsBroker Service Broker des événements système
Running  TbtHostController… Thunderbolt(TM) Application Launcher
Running  TbtP2pShortcutSer… Thunderbolt(TM) Peer to Peer Shortcut
Running  TextInputManageme… Service de gestion des entrées de tex…
Running  Themes             Thèmes
Running  TimeBrokerSvc      Service Broker pour les événements ho…
Running  TokenBroker        Gestionnaire de comptes web
Running  TrkWks             Client de suivi de lien distribué
Running  UdkUserSvc_36a8567 Service utilisateur du kit de dévelop…
Running  UserManager        Gestionnaire des utilisateurs
```

11. Opérateurs de redirection

Les opérateurs de redirection permettent de rediriger le résultat d'une commande vers un fichier, dans un dossier… Exemple avec la commande Get-Process :



```
PS C:\Users\rdubois\Desktop\PowerShell> . 'C:\Users\rdubois\Desktop\PowerShell\powershell.ps1'
PS C:\Users\rdubois\Desktop\PowerShell>
```

A première vu rien ne s'est passé, mais si on vérifie le contenu du dossier dans lequel on est, on peut observer qu'on a bien un fichier « processes.txt » qui c'est créer content le résultat de la commande :

```
PS C:\Users\rdubois\Desktop\PowerShell> dir

    Directory: C:\Users\rdubois\Desktop\PowerShell

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          16/03/2025    14:49             30 powershell.ps1
-a---          16/03/2025    14:49          13831 processes.txt

PS C:\Users\rdubois\Desktop\PowerShell>
```

```
PS C:\Users\rdubois\Desktop\PowerShell> Get-Content .\processes.txt

 NPM(K)    PM(M)      WS(M)     CPU(s)      Id  SI ProcessName
 ------    -----      -----     ------      --  -- -----------
     13     3,83       4,09       0,00    7536   0 AggregatorHost
     15    21,73       4,11       0,64    9236   5 ai
     22    14,21      19,55       1,34    8700   5 ApplicationFrameHost
     27    81,34      33,71       1,92    7776   5 Code
     23    31,96     106,42       0,69    7824   5 Code
     46   448,79      95,06      29,95    8280   5 Code
     49   242,33     224,82     126,02    9040   5 Code
     28    84,77      46,61       5,70   10456   5 Code
     59   107,40      89,61      45,53   11588   5 Code
     38   114,99      54,92       7,62   13964   5 Code
     30   162,01     153,75      54,42   14984   5 Code
     20    14,14      23,54       0,72   16464   5 Code
     13    11,04      12,16       0,08   16936   5 Code
      9     1,29       4,46       0,56    4116   5 conhost
      9     1,27       4,12       0,09   16208   5 conhost
     27     2,21       1,21       0,00     960   0 csrss
     25     2,84       2,88       0,00   14152   5 csrss
     22    10,65      15,11      39,52   11584   5 ctfmon
      9     1,56       0,03       0,00   12056   0 dasHost
     10     1,89       2,68       0,39   13696   5 dllhost
     24     5,69       5,33       0,73   16488   5 dllhost
      8     1,53       1,29       9,20   14276   5 dptf_helper
     57   163,25      90,32       0,00   16500   5 dwm
      9     2,07       1,89       0,00    5400   0 esif_uf
    190   247,15     182,76     427,05    1956   5 explorer
     21     8,78      12,84       6,36   13232   5 FileCoAuth
     18     5,89       9,40       0,00    1192   0 FileSyncHelper
     66   758,56     372,61     123,67    1604   5 firefox
     44   120,19     117,94      45,52    1660   5 firefox
```

12. Enumération

Exemple d'une énumération de priorité allant de faible (0) à élever (2) :

```powershell
# Définition d'une énumération pour représenter la priorité d'une tâche
enum Priorite {
    Faible    # La valeur par défaut est 0
    Normal    # La valeur sera 1
    Elevee    # La valeur sera 2
}

# Affectation d'une valeur de l'énumération à une variable
$niveauTache = [Priorite]::Normal

# Affichage du résultat
Write-Host "La priorité de la tâche est:" $niveauTache
Write-Host "La valeur numérique associée est:" ([int]$niveauTache)

```

Dans ce script, une table d'énumération est définie allant de faible à élever, les valeurs 0,1,2 sont défini automatiquement par défaut en suivant l'ordre de chaque élément.

Ensuite, une variable est créée « niveauTache » contenant le nom de notre table d'énumération suivi de la priorité que l'on souhaite définir. Dans l'exemple ci-dessus, la priorité est définie en normal.

Puis enfin, il suffit d'afficher nos valeurs avec un Write-Host. Le premier Write-Host va seulement afficher la priorité avec la valeur avec son texte défini, tandis que le second Write-Host va afficher la valeur en nombre entier (ou numérique) de la priorité « normal ».

Voici le résultat de sorti pour la valeur « Normal » :

```
PS C:\Users\rdubois\Desktop\PowerShell> . 'C:\Users\rdubois\Desktop\PowerShell\powershell.ps1'
La priorité de la tâche est: Normal
La valeur numérique associée est: 1
PS C:\Users\rdubois\Desktop\PowerShell>
```

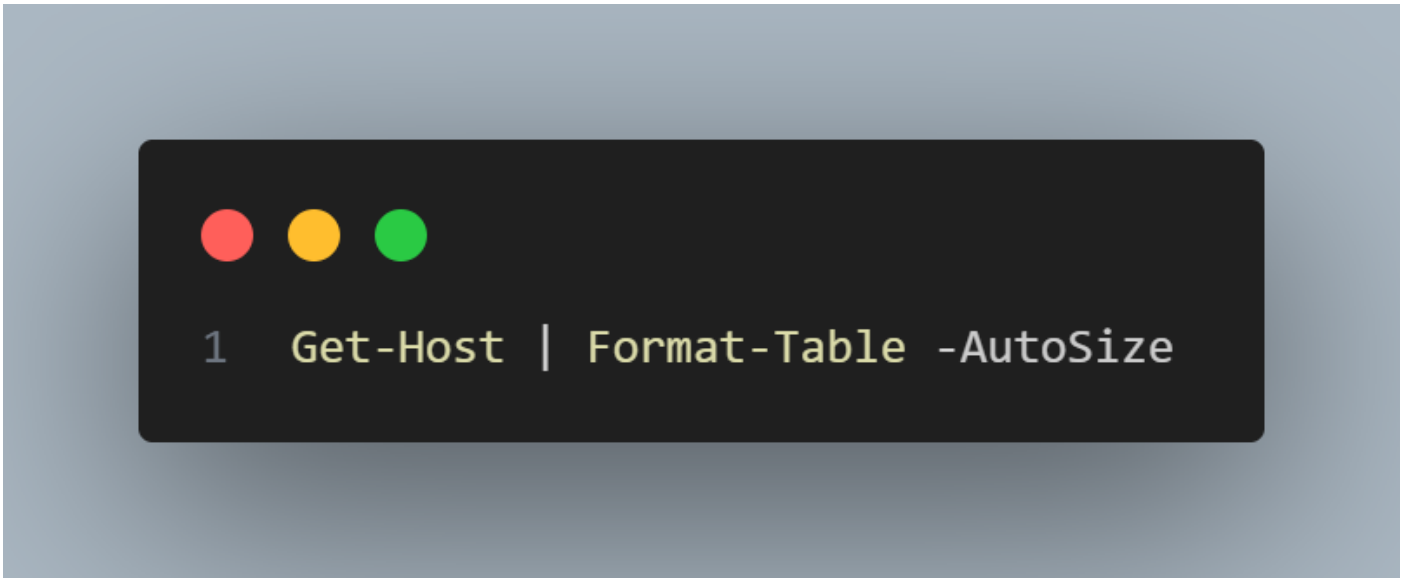
On remarque que la valeur numérique est égale à 1 ce qui est normal étant donné que la première valeur est 0 (faible), puis 1 (normal) et enfin 2 (élevé). Exemple avec Faible et Elevée :

```
PS C:\Users\rdubois\Desktop\PowerShell> . 'C:\Users\rdubois\Desktop\PowerShell\powershell.ps1'
La priorité de la tâche est: Faible
La valeur numérique associée est: 0
PS C:\Users\rdubois\Desktop\PowerShell>
```

```
PS C:\Users\rdubois\Desktop\PowerShell> . 'C:\Users\rdubois\Desktop\PowerShell\powershell.ps1'
La priorité de la tâche est: Elevee
La valeur numérique associée est: 2
PS C:\Users\rdubois\Desktop\PowerShell>
```

13. Format-Table

La commande « Format-Table », souvent utilisé en pipe avec une première commande, permet d'afficher le résultat de la commande sous forme de tableau, exemple avec Get-Host :



```
PS C:\Users\rdubois\Desktop\PowerShell> . 'C:\Users\rdubois\Desktop\PowerShell\powershell.ps1'

Name                  Version  InstanceId                           UI
----                  -------  ----------                           --
Visual Studio Code Host 2025.0.0 0e58d5c5-2945-4c9e-990c-a905a28bc037 System.Management.Automation.Internal.Host.I…

PS C:\Users\rdubois\Desktop\PowerShell>
```