

## CR SCRIPTING PWSHELL

Voici avant tout le définition de quelques commande powershell de base :

- **Get-ChildItem**

Liste les fichiers et dossiers du répertoire courant.

- **Write-Output (ou simplement Write)**

Affiche du texte à l'écran.

- **Read-Host**

Permet de demander une entrée à l'utilisateur.\*

- **New-Item**

Crée un fichier ou un dossier.

- **Get-Process**

Liste les processus en cours d'exécution.

exo 1

```
PS E:\script pw> E:\script pw\scripexo1.ps1
Entrez une chaîne : vhfjhv
la valeur saisie est vhfjhv et modifiée en : vhfjhv concaténée

PS E:\script pw> ./scripexo1.ps1
Entrez une chaîne : |
```

## exo 2

scripexo1.ps1

```
1 $n=Read-Host "donné votre nom">#permet de lire le clavier et de placer la valeur dans r
2 $p=Read-Host "donné votre prénom"
3 $ch=$n.Substring(0,1)+$p
4 Write-Host "votre login est : " $ch
5
```

<

```
fichier, fichier de script ou programme exécutable. Vérifiez l'orthographe du nom, ou si un
chemin d'accès existe, vérifiez que le chemin d'accès est correct et réessayez.
Au caractère E:\script pw\scripexo1.ps1:3 : 5
+ $ch=ch=$n.Substring(1)+$p " concatenee"
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (ch=$n.Substring:String) [], CommandNotFoundException
on
+ FullyQualifiedErrorId : CommandNotFoundException

la valeur saisie est vhfjhv et modifiee en :

PS E:\script pw> ./scripexo1.ps1
donné votre nom : mrabet
donné votre prénom : addel
la valeur saisie est vhfjhv et modifiee en : rabetaddel

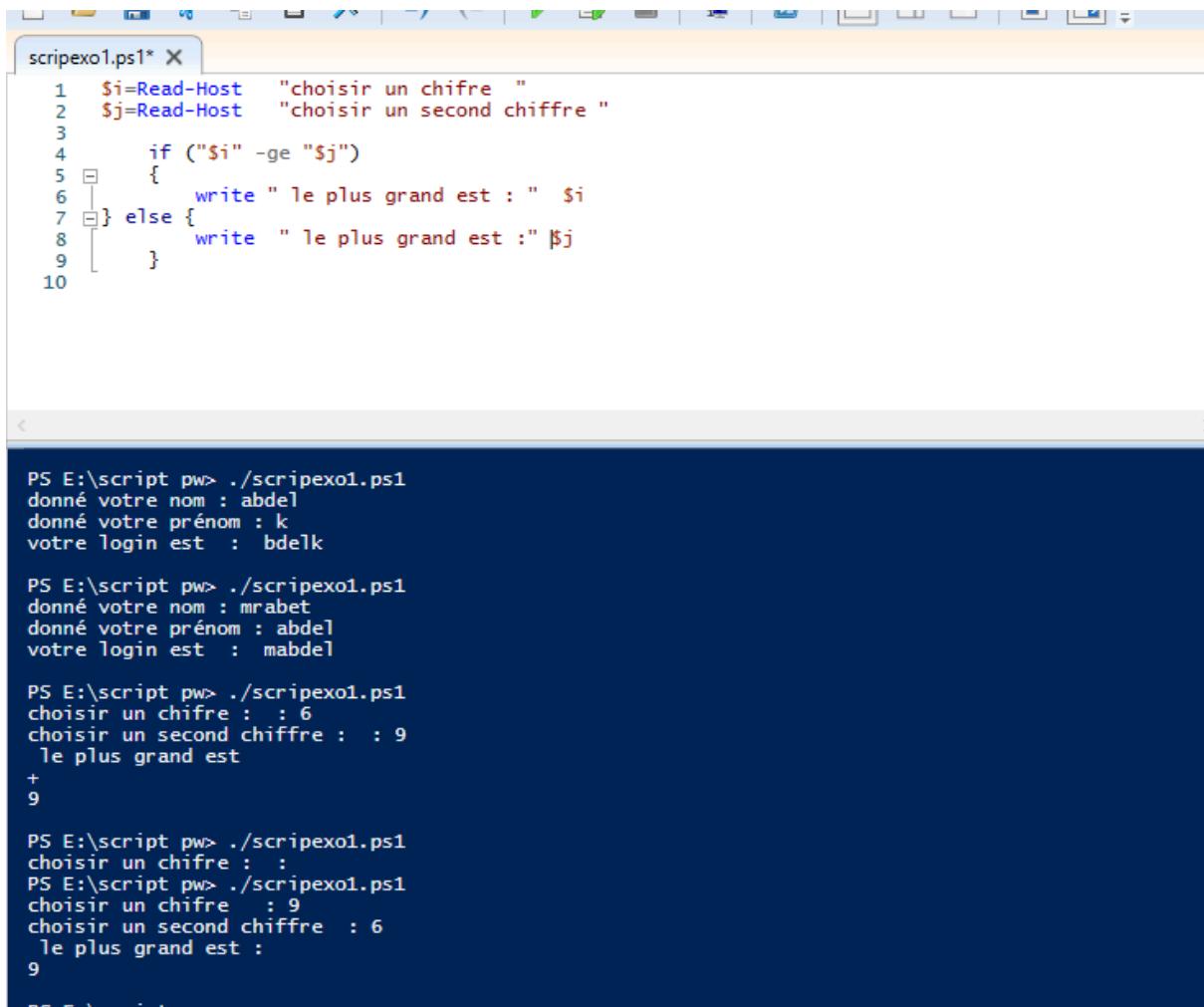
PS E:\script pw> ./scripexo1.ps1
donné votre nom : abdel
donné votre prénom : k
votre login est : bdelk

PS E:\script pw> ./scripexo1.ps1
donné votre nom : mrabet
donné votre prénom : abdel
votre login est : mabdel

PS E:\script pw>
```

>

### exo 3



The screenshot shows a Windows PowerShell window with the following content:

```
scripexo1.ps1* X
1 $i=Read-Host "choisir un chiffre "
2 $j=Read-Host "choisir un second chiffre "
3
4     if ("$i" -ge "$j")
5     {
6         write " le plus grand est : " $i
7     } else {
8         write " le plus grand est :" $j
9     }
10

<

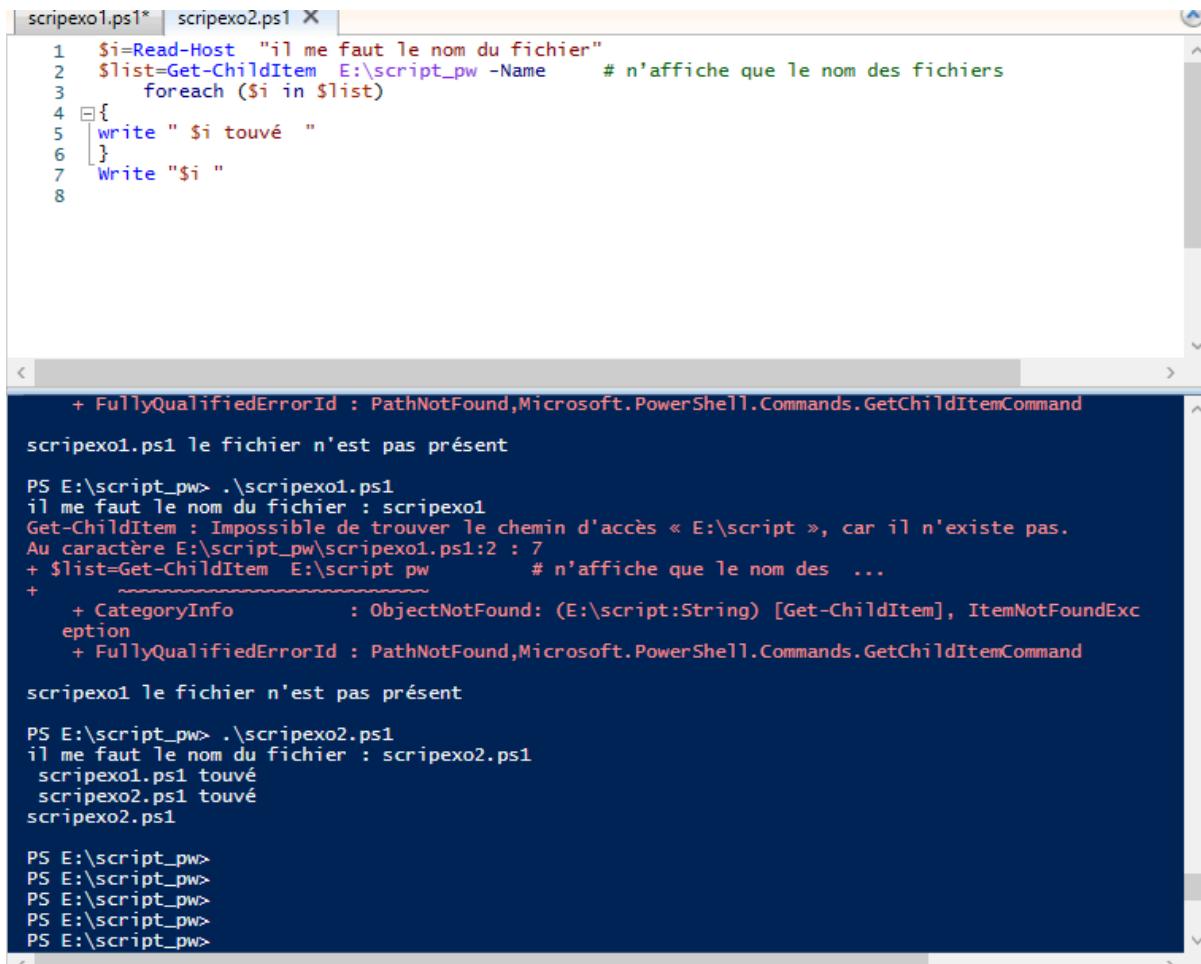
PS E:\script pw> ./scripexo1.ps1
donné votre nom : abdel
donné votre prénom : k
votre login est : bdelk

PS E:\script pw> ./scripexo1.ps1
donné votre nom : mrabet
donné votre prénom : abdel
votre login est : mabdel

PS E:\script pw> ./scripexo1.ps1
choisir un chiffre : 6
choisir un second chiffre : 9
le plus grand est
+
9

PS E:\script pw> ./scripexo1.ps1
choisir un chiffre : 9
PS E:\script pw> ./scripexo1.ps1
choisir un chiffre : 9
choisir un second chiffre : 6
le plus grand est :
9
```

#### exo4



```
scripexo1.ps1* | scripexo2.ps1 X
1  $i=Read-Host "il me faut le nom du fichier"
2  $list=Get-ChildItem E:\script_pw -Name      # n'affiche que le nom des fichiers
3      foreach ($i in $list)
4  {
5      write " $i trouvé "
6  }
7  Write "$i "
8

+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand
scripexo1.ps1 le fichier n'est pas présent

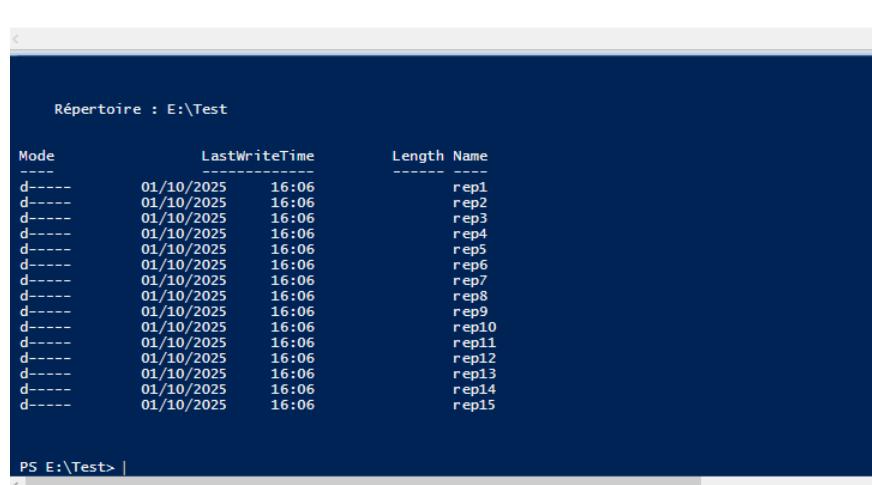
PS E:\script_pw> .\scripexo1.ps1
il me faut le nom du fichier : scripexo1
Get-ChildItem : Impossible de trouver le chemin d'accès « E:\script », car il n'existe pas.
Au caractère E:\script_pw\scripexo1.ps1:2 : 7
+ $list=Get-ChildItem E:\script pw      # n'affiche que le nom des ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (E:\script:String) [Get-ChildItem], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand
scripexo1 le fichier n'est pas présent

PS E:\script_pw> .\scripexo2.ps1
il me faut le nom du fichier : scripexo2.ps1
scripexo1.ps1 trouvé
scripexo2.ps1 trouvé
scripexo2.ps1

PS E:\script_pw>
PS E:\script_pw>
PS E:\script_pw>
PS E:\script_pw>
PS E:\script_pw>
```

## exo 5

```
scripexo1.ps1* scripexo2.ps1* scripexo5.ps1 X
1 New-Item E:\Test -ItemType directory
2 $ch="rep"
3 set-location "E:\Test"
4 for ($i=1;$i -le 15;$i++) {
5     New-Item $ch$i -ItemType directory
6 }
7
```

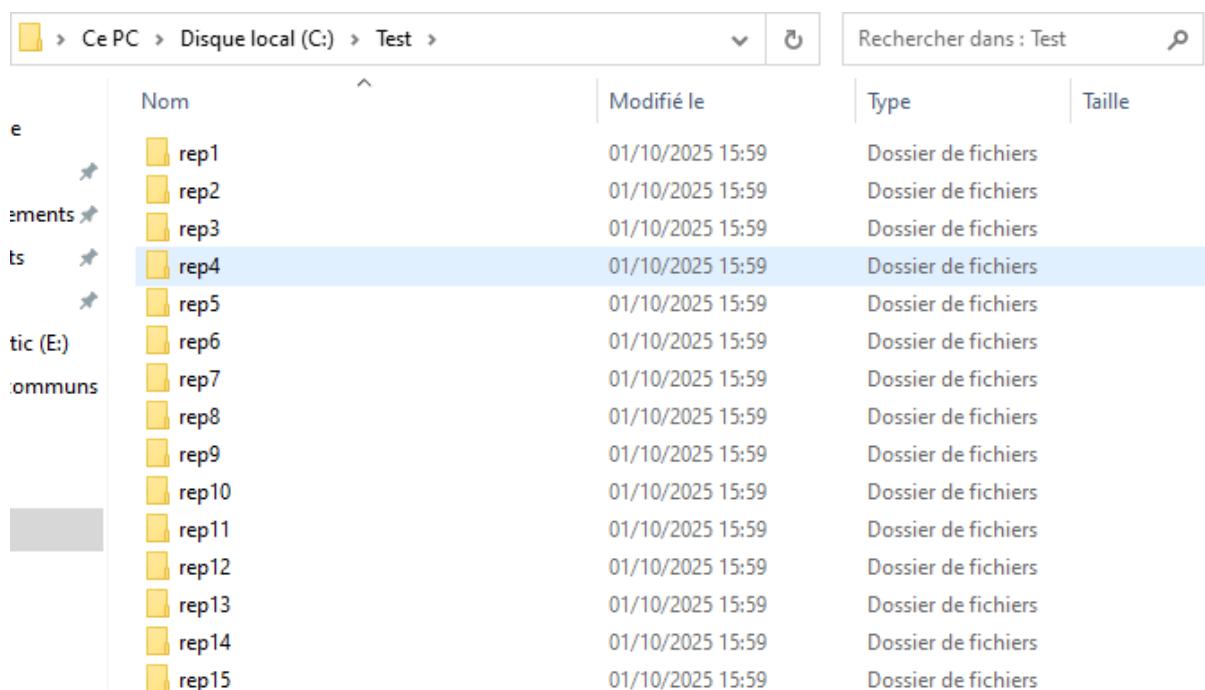


```
Répertoire : E:\Test

Mode          LastWriteTime    Length Name
----          -----
d----  01/10/2025 16:06          rep1
d----  01/10/2025 16:06          rep2
d----  01/10/2025 16:06          rep3
d----  01/10/2025 16:06          rep4
d----  01/10/2025 16:06          rep5
d----  01/10/2025 16:06          rep6
d----  01/10/2025 16:06          rep7
d----  01/10/2025 16:06          rep8
d----  01/10/2025 16:06          rep9
d----  01/10/2025 16:06          rep10
d----  01/10/2025 16:06          rep11
d----  01/10/2025 16:06          rep12
d----  01/10/2025 16:06          rep13
d----  01/10/2025 16:06          rep14
d----  01/10/2025 16:06          rep15

PS E:\Test> |
```

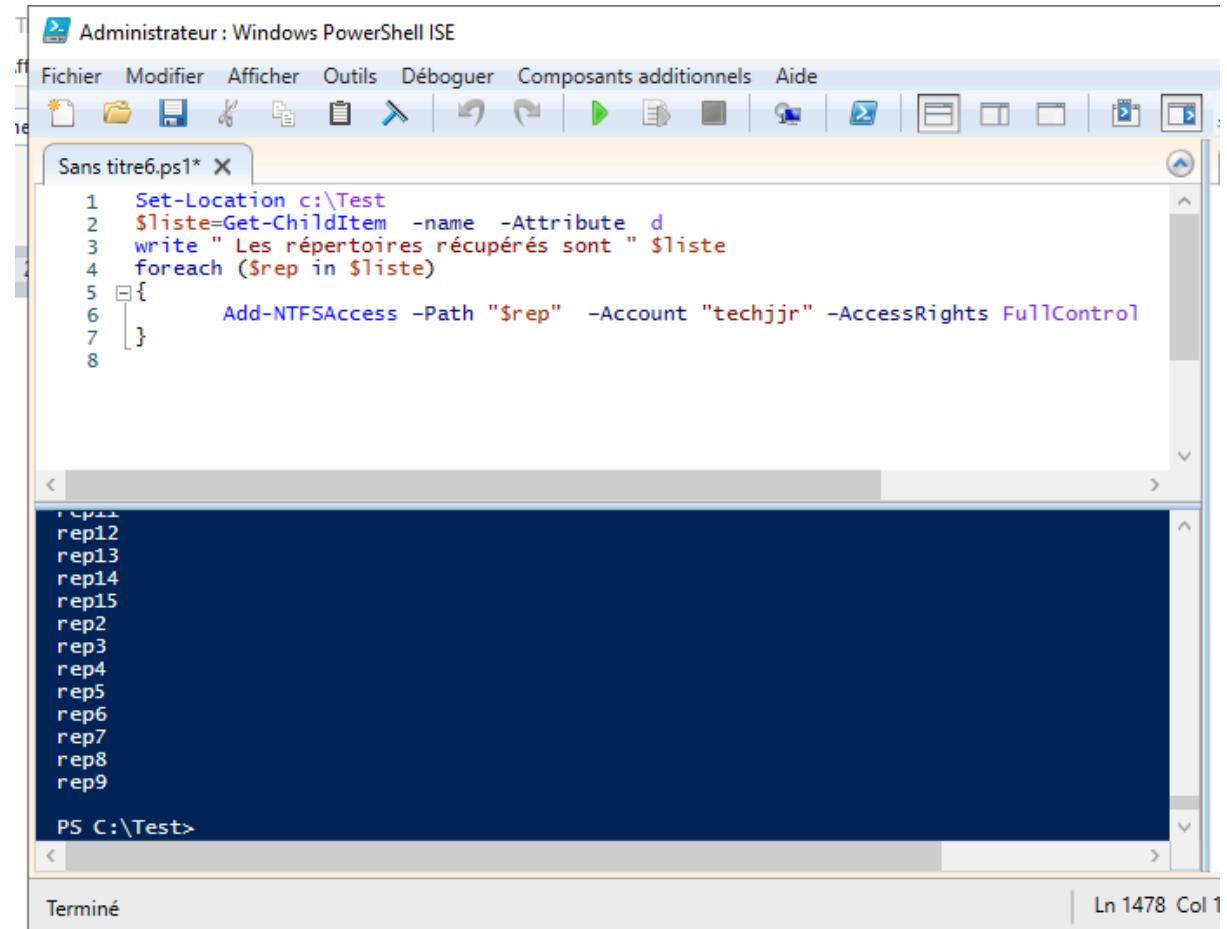
decire ce que fait lalgo ligne par ligne



	Nom	Modifié le	Type	Taille
e	rep1	01/10/2025 15:59	Dossier de fichiers	
éments	rep2	01/10/2025 15:59	Dossier de fichiers	
ts	rep3	01/10/2025 15:59	Dossier de fichiers	
tic (E:)	rep4	01/10/2025 15:59	Dossier de fichiers	
ommuns	rep5	01/10/2025 15:59	Dossier de fichiers	
	rep6	01/10/2025 15:59	Dossier de fichiers	
	rep7	01/10/2025 15:59	Dossier de fichiers	
	rep8	01/10/2025 15:59	Dossier de fichiers	
	rep9	01/10/2025 15:59	Dossier de fichiers	
	rep10	01/10/2025 15:59	Dossier de fichiers	
	rep11	01/10/2025 15:59	Dossier de fichiers	
	rep12	01/10/2025 15:59	Dossier de fichiers	
	rep13	01/10/2025 15:59	Dossier de fichiers	
	rep14	01/10/2025 15:59	Dossier de fichiers	
	rep15	01/10/2025 15:59	Dossier de fichiers	

résultat

## exo6



The screenshot shows the Windows PowerShell ISE interface. The top window is titled "Sans titre6.ps1\*" and contains the following PowerShell script:

```
1 Set-Location c:\Test
2 $liste=Get-ChildItem -name -Attribute d
3 write " Les répertoires récupérés sont " $liste
4 foreach ($rep in $liste)
5 {
6     Add-NTFSAccess -Path "$rep" -Account "techjjr" -AccessRights FullControl
7 }
8
```

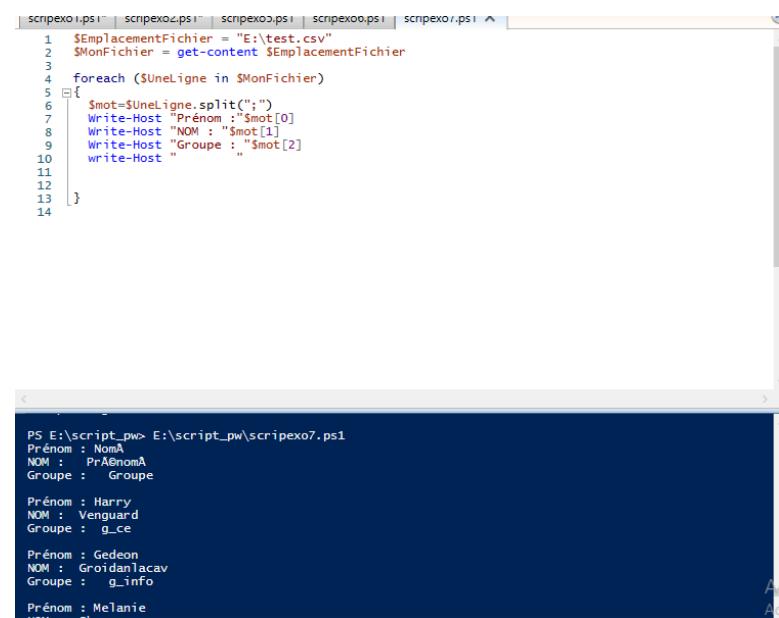
The bottom window shows the output of the script:

```
rep12
rep13
rep14
rep15
rep2
rep3
rep4
rep5
rep6
rep7
rep8
rep9
```

The status bar at the bottom right indicates "Terminé" (Completed) and "Ln 1478 Col 1".

expliquer à quoi sert le programme

## exo7



The screenshot shows the Windows PowerShell ISE interface. The top window is titled "scripexo1.ps1" and contains the following PowerShell script:

```
1 $EmplacementFichier = "E:\test.csv"
2 $MonFichier = get-content $EmplacementFichier
3
4 foreach ($UneLigne in $MonFichier)
5 {
6     $mot=$UneLigne.split(",")
7     Write-Host "Prénom :" $mot[0]
8     Write-Host "NOM : " $mot[1]
9     Write-Host "Groupe : " $mot[2]
10    write-Host ""
11
12
13 }
```

The bottom window shows the output of the script, reading data from a CSV file:

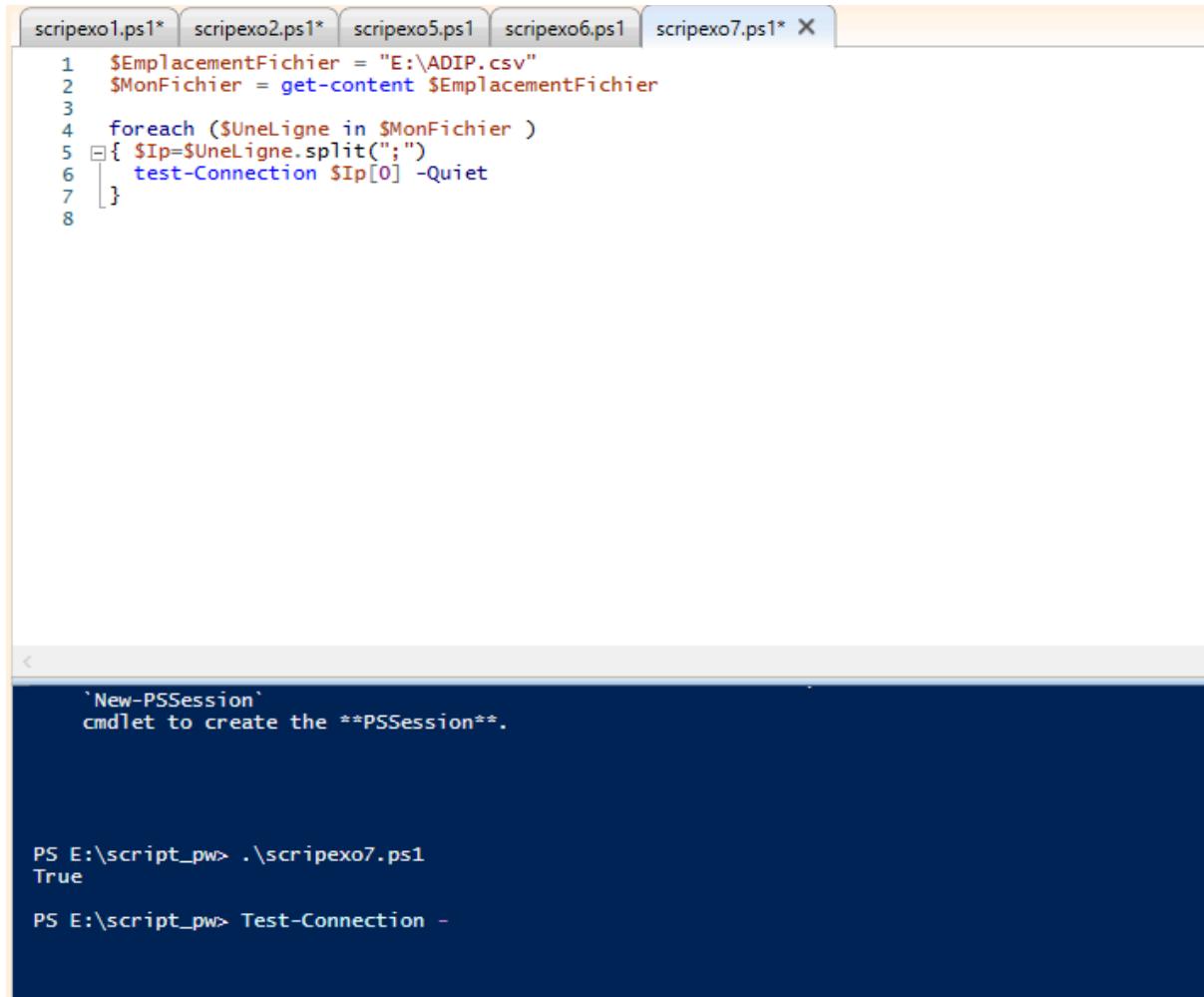
```
PS E:\script_pw> E:\script_pw\scripexo7.ps1
Prénom : NomA
NOM : PrénomA
Groupe : Groupe

Prénom : Harry
NOM : Vanguard
Groupe : g_ce

Prénom : Gédéon
NOM : Goidanlacav
Groupe : g_info

Prénom : Mélanie
```

## exo 8



The screenshot shows a Windows PowerShell window with a script editor interface. The script editor tab bar at the top includes tabs for 'scripexo1.ps1\*', 'scripexo2.ps1\*', 'scripexo5.ps1', 'scripexo6.ps1', and 'scripexo7.ps1\*'. The 'scripexo7.ps1\*' tab is active. The script content is as follows:

```
1 $EmplacementFichier = "E:\ADIP.csv"
2 $MonFichier = get-content $EmplacementFichier
3
4 foreach ($UneLigne in $MonFichier )
5 { $Ip=$UneLigne.split(";")
6   test-Connection $Ip[0] -Quiet
7 }
8
```

Below the script editor, the PowerShell command-line interface shows the following output:

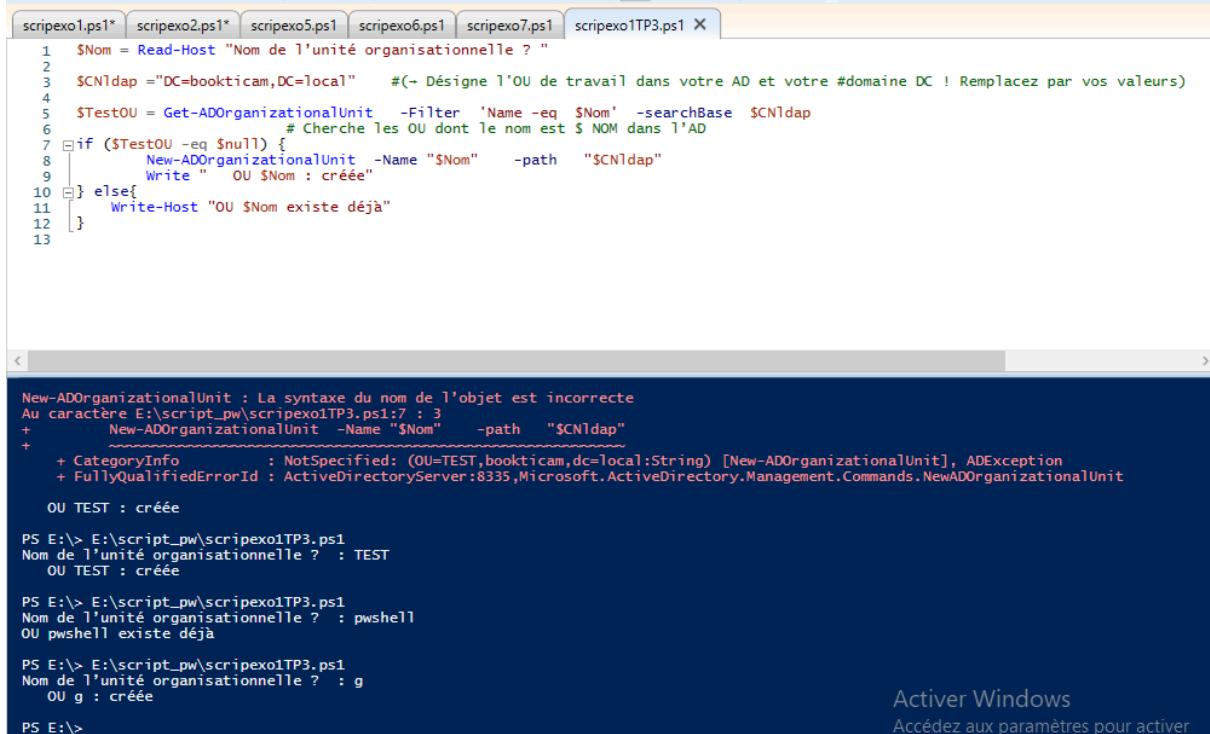
```
<
`New-PSSession`:
cmdlet to create the **PSSession**.

PS E:\script_pw> .\scripexo7.ps1
True

PS E:\script_pw> Test-Connection -
```

# TP3

## préambule 1



```
scripexo1.ps1* scripexo2.ps1* scripexo5.ps1 scripexo6.ps1 scripexo7.ps1 scripexo1TP3.ps1 X
1 $Nom = Read-Host "Nom de l'unité organisationnelle ? "
2
3 $CNldap ="DC=bookticam,DC=local" #(- Désigne l'OU de travail dans votre AD et votre #domaine DC ! Remplacez par vos valeurs)
4
5 $TestOU = Get-ADOrganizationalUnit -Filter 'Name -eq $Nom' -searchBase $CNldap
6
7 if ($TestOU -eq $null) {
8     New-ADOrganizationalUnit -Name "$Nom" -path "$CNldap"
9     Write " OU $Nom : créée"
10 } else{
11     Write-Host "OU $Nom existe déjà"
12 }
13

New-ADOrganizationalUnit : La syntaxe du nom de l'objet est incorrecte
Au caractère E:\script_pw\scripexo1TP3.ps1:7 : 3
+     New-ADOrganizationalUnit -Name "$Nom" -path "$CNldap"
+     ~~~~~
+ CategoryInfo          : NotSpecified: (OU=TEST,bookticam,dc=local:String) [New-ADOrganizationalUnit], ADErrorException
+ FullyQualifiedErrorId : ActiveDirectoryServer:8335,Microsoft.ActiveDirectory.Commands.NewADOrganizationalUnit

OU TEST : créée

PS E:\> E:\script_pw\scripexo1TP3.ps1
Nom de l'unité organisationnelle ? : TEST
OU TEST : créée

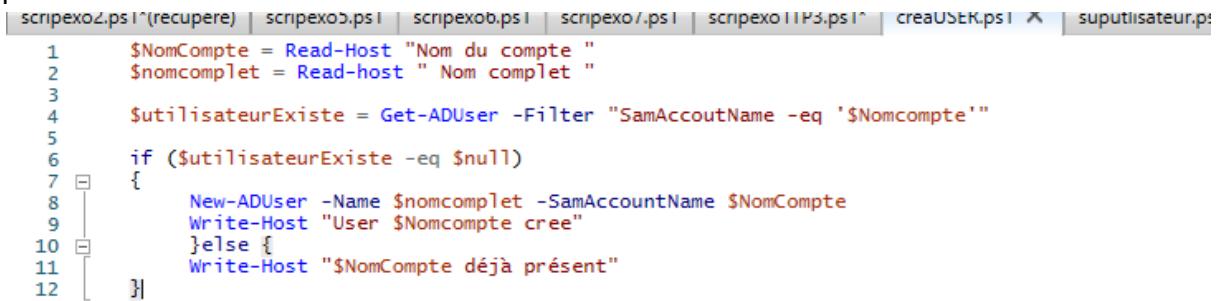
PS E:\> E:\script_pw\scripexo1TP3.ps1
Nom de l'unité organisationnelle ? : pwshell
OU pwshell existe déjà

PS E:\> E:\script_pw\scripexo1TP3.ps1
Nom de l'unité organisationnelle ? : g
OU g : créée

PS E:\>
```

Activer Windows  
Accédez aux paramètres pour activer.

## part. A



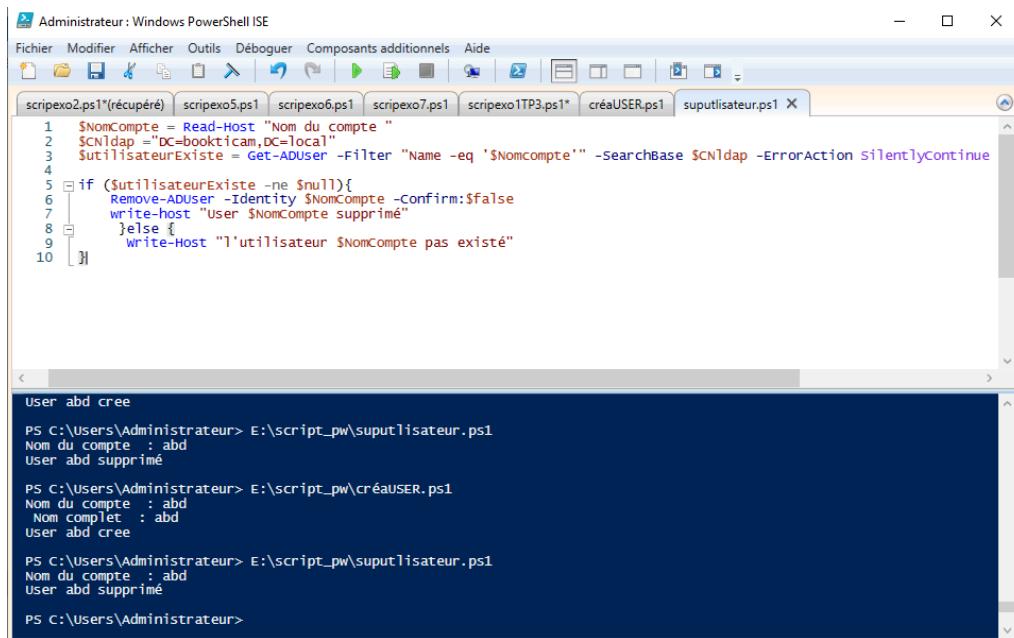
```
scripexo2.ps1*(recupere) scripexo5.ps1 scripexo6.ps1 scripexo7.ps1 scripexo1TP3.ps1* creaUSER.ps1 X suputilisateur.ps1
1 $NomCompte = Read-Host "Nom du compte "
2 $nomcomplet = Read-host " Nom complet "
3
4 $utilisateurExiste = Get-ADUser -Filter "SamAccountName -eq '$Nomcompte'"
5
6 if ($utilisateurExiste -eq $null)
7 {
8     New-ADUser -Name $nomcomplet -SamAccountName $NomCompte
9     Write-Host "User $Nomcompte cree"
10 } else {
11     Write-Host "$NomCompte déjà présent"
12 }
```

```

scripexo2.ps1*(récupéré) scripexo5.ps1 scripexo6.ps1 scripexo7.ps1 scripexo1TP3.ps1* créUSER.ps1 suputilisateur.ps1 exo3TP3.ps1 X exo2TP3.ps1
1 $Ficusers= 'E:\script_pw\user.csv'
2 $Entete="Prenom","Nom","Groupe"
3 $utilisateurs=import-csv $Ficusers -Delimiter ';' -header $($Entete)
4 $CNldap="ou=test,dc=bookticAM,dc=local"
5 foreach ($o in $utilisateurs)
6 {
7     $Exists="existe"
8     $Grp=$o.Groupe
9     $TestOU=Get-ADOrganizationalUnit -Filter 'Name -eq $Grp' -searchBase $CNldap
10    if ($($TestOU) -eq $null) {
11        $Exists="ajoutée"
12        New-ADOrganizationalUnit -Name "$Grp" -path "$CNldap"
13    }
14    write-Host "$exists"
15 }
16
17

```

## part. B



```

Administrator : Windows PowerShell ISE
Fichier Modifier Afficher Outils Déboguer Composants additionnels Aide
scripexo2.ps1*(récupéré) scripexo5.ps1 scripexo6.ps1 scripexo7.ps1 scripexo1TP3.ps1* créUSER.ps1 suputilisateur.ps1 X
1 $Nomcompte = Read-Host "Nom du compte"
2 $CNldap = "DC=bookticAM,DC=local"
3 $utilisateurExiste = Get-ADUser -Filter "Name -eq '$Nomcompte'" -searchBase $CNldap -ErrorAction silentlyContinue
4
5 if ($utilisateurExiste -ne $null){
6     Remove-ADUser -Identity $Nomcompte -Confirm:$false
7     write-host "User $Nomcompte supprimé"
8 }else {
9     write-Host "l'utilisateur $Nomcompte pas existé"
10 }

User abd cree
PS C:\Users\Administrateur> E:\script_pw\suputilisateur.ps1
Nom du compte : abd
User abd supprimé

PS C:\Users\Administrateur> E:\script_pw\créUSER.ps1
Nom du compte : abd
Nom complet : abd
User abd cree

PS C:\Users\Administrateur> E:\script_pw\suputilisateur.ps1
Nom du compte : abd
User abd supprimé

PS C:\Users\Administrateur>

```

## Exo1 tp3

\$o fonctionne d'une manière spéciale quand la première boucle s'effectue \$o représente chaque élément du tableau \$utilisateurs à chaque tour de boucle. il vas prendre comme délimiteur le point virgule

```
1 $Ficusers = 'E:\script_pw\user.csv'
2 $Entete = "prenom", "Nom", "Groupe"
3 $Utilisateurs =import-csv $Ficusers -Delimiter ';' -header $($Entete)
4 $CNldap = "ou=Test,dc=bookticAM,dc=local"
5
6 foreach ($o in $Utilisateurs)
7 {
8     $Exists = "existante"
9     $Grp = $o.Groupe
10    $TestOU=Get-ADOrganizationalUnit -Filter 'Name -eq $Grp' -searchBase $CNldap
11
12    if ($TestOU -eq $null) {
13        $Exists = "ajoutée"
14        New-ADOrganizationalUnit -Name "$Grp" -path "$CNldap" -ProtectedFromAccidentalDeletion $false
15        New-ADGroup -Name "$Grp" -GroupScope Global -GroupCategory Security -path "OU=$Grp,$CNldap"
16    }
17
18    # CRÉATION DE L'UTILISATEUR
19    $Prenom = $o.Prenom
20    $Nom = $o.Nom
21    $Login = $Prenom.Substring(0,1)+$Nom
22    $OUPath = "OU=$Grp,$CNldap"
23
24    New-ADUser -Name "$Prenom $Nom" `
25        -GivenName $Prenom `
26        -Surname $Nom `
27        -SamAccountName $Login `
28        -UserPrincipalName "$Login@bookticaw.local" `
29        -Path $OUPath `
30        -AccountPassword (ConvertTo-SecureString "$Prenom.Substring(0,1)+$Nom" -AsPlainText -Force) `
31        -Enabled $true
32
33    Add-ADGroupMember -Identity $Grp -Members $Login
34
35    Write-Host "OU $Grp : $exists - Utilisateur $Login créé et ajoutée dans le groupe $Grp"
36 }
```