

PowerShell - exercices

1. Listez tous les dossiers (et uniquement les dossiers) à partir de C:\.

```
gci -Path C:\ -Recurse -Directory -ErrorAction SilentlyContinue
```

2. Listez les fichiers de Windows à partir de C:\, triés par taille croissante.

```
gci -Path C:\ -Recurse -ErrorAction SilentlyContinue | Sort-Object length
```

3. Listez les fichiers de Windows à partir de C:\, triés par taille décroissante,

en affichant uniquement la taille et le nom du fichier.

```
gci -Path C:\ -Recurse -ErrorAction SilentlyContinue | Sort-Object length
```

```
-Descending | Select length, name
```

4. Listez tous les fichiers jpg sur le disque C:\

```
gci -Path C:\ -Recurse -filter *.jpg -ErrorAction SilentlyContinue
```

5. Créez l'arborescence suivante dans le dossier Documents: Toto > Musiques

```
..> Images > Original
```

```
..> > Travail
```

```
..> > Jpg
```

```
..> Documents > Perso ..> > Divers
```

```
ni -path c:\users\azerty\documents\toto\images\original -itemtype directory
```

```
ni -path c:\users\azerty\documents\toto\images\travail -itemtype directory
```

```
ni -path c:\users\azerty\documents\toto\images\jpg -itemtype directory
```

```
ni -path c:\users\azerty\documents\toto\documents\perso -itemtype directory ni -path
```

```
c:\users\azerty\documents\toto\documents\divers -itemtype directory ni -path
```

```
c:\users\azerty\documents\toto\musique -itemtype directory
```

6. Supprimez toute l'arborescence précédente en une seule commande. Demandez la confirmation à l'utilisateur.

```
ri -path c:\users\azerty\documents\toto -confirm
```

7. Ecrivez un fichier *.log* avec comme contenu tous les fichiers avec

l'extension *.log* se trouvant sur le disque dur.

```
gci -path c:\ -recurse -filter *.log -erroraction silentlycontinue | out-file -filepath  
c:\users\azerty\desktop\liste.log
```

8. Ajoutez à ce même fichier la liste de tous les fichiers *.exe* se trouvant sur le disque dur.

```
gci -path c:\ -recurse -filter *.exe -erroraction silentlycontinue | add-content -path  
c:\users\azerty\desktop\liste.log
```

9. Comptez le nombre de fichiers .exe présent sur votre disque dur
gci -path c:\ -recurse -filter *.exe -erroraction silentlycontinue | measure-object

10. Copiez les image jpg de votre disque dur dans un dossier nommé *Backup/Images*.

```
gci c:\ -recurse -filter *.jpg -erroraction silentlycontinue | foreach-object {copy-item  
"$_.fullname" -destination c:\users\azerty\desktop\backup\images}  
=>fonctionne mais je dois créer avant tout les dossiers backup\images
```

11. Déplacer le dossier backup\images précédemment créé dans le dossier documents

```
move-item c:\users\...\desktop\backup\ -destination c:\users\...\documents\
```

12. renommez les dossiers backup/images en Bkp\imgs

```
rename-item c:\users\...\documents\backup -newname Bkp  
rename-item c:\users\...\documents\Bkp\images -newname imgs
```

13. Supprimez le dossier Bkp et tous son contenu.

```
Remove-item c:\users\...\documents\Bkp
```

14. Ecrivez "Vive POWERSHELL" en noir sur fond jaune

```
Write-host « Vive POWERSHELL » -backgroundcolor Yellow -foregroundcolor Black
```

15. Affichez une liste des processus actifs commençant par la letter s

```
Get-process | where name -like s*
```

16. indiquez uniquement le displayname et status du service fhsvc

```
Get-service | where name -like fhsvc | select displayname, status
```

17. Ecrivez un script qui demande un nom de fichier à l'utilisateur. Le script doit indiquer si le fichier existe ou non

```
Clear-host  
$chemin = read-host -prompt « Veuillez entrer le chemin de votre fichier »  
$existe = test-path $chemin  
If($existe){  
    "Votre fichier existe"  
}  
Else{  
    "Votre fichier est inexistant"  
}
```

18 . Ecrivez un script qui demande à l'utilisateur un fichier à copier, ainsi que l'endroit où le copier. Verifier que le fichier existe et qu'il n'est pas déjà présent dans le dossier de destination

```
$chemininitial = read-host -prompt « veuillez entrer le chemin dy fichier à copier »
$cheminfinal = read-host -prompt « veuillez entrer le chemin de destination du fichier »
$existeinit = test-path $chemininitial
$existefinal = test-path $cheminfinal
$filename = split-path $chemininitial -leaf
```

```
If ($existeinit) {
    If($existefinal) {
        If(test-path $cheminfinal\$filename){
            "votre fichier existe déjà dans le dossier de destination »
        }
        Else{
            « votre fichier va être déplacé... »
            Move-item -path $chemininitial -destination $cheminfinal
            "votre fichier a été déplacé ! »
        }
    }
    Else{
        « votre dossier de destination n'existe pas »
    }
}
Else{
    « votre fichier n'existe pas »
}
```

19. Écrivez un script qui demande à l'utilisateur de deviner un nombre compris entre 0 et 100. L'ordinateur choisira ce nombre au hasard. L'ordinateur indiquera en vert **Plus grand** et en rouge **Plus petit**. Quand le nombre est trouvé, l'ordinateur affichera en couleur (au choix) le message *Bravo, vous avez trouvé n en x coups*. n et x sont à remplacer par leur valeur.

```
$alea = get-random -maximum 100 -minimum 0
$nbr = 101
$nbrcoup = 1
While($nbr -ne $alea){
    $nbr = read-host -prompt "quel est le nombre choisi ? »
    If($nbr -gt $alea){
        Write-host « plus petit » -foregroundcolor Red
        $nbrcoup += 1
    }
    ElseIf($nbr -lt -alea){
        Write-host "plus grand" -foregroundcolor Green
        $nbrcoup += 1
    }
}
```

```

    }
Else{
    Write-host "bravo ! vous avez trouvé $alea en $nbrcoup coup(s) » -
foregroundcolor darkmagenta -backgroundcolor Green
    }
}

```

20. Améliorez le script précédent pour rejouer tant que l'utilisateur le souhaite et limiter le nombre d'essais à 10 coups.

```

$choix = "oui"
While($choix -ieq "oui"){
    $alea = get-random -maximum 100 -minimum 0
    $nbr = 101
    $nbrcoup = 1
    While(($nbr -ne $alea) -and ($nbrcoup -ne 11)){
        $nbr = read-host -prompt "quel est le nombre choisi ? »
        If($nbr -gt $alea){
            Write-host « plus petit » -foregroundcolor Red
            $nbrcoup += 1
        }
        ElseIf($nbr -lt -alea){
            Write-host "plus grand" -foregroundcolor Green
            $nbrcoup += 1
        }
        Else{
            Write-host "bravo ! vous avez trouvé $alea en $nbrcoup coup(s) » -
foregroundcolor darkmagenta -backgroundcolor Green
        }
    }
    If($nbrcoup -ge 10){« vous avez perdu ! »}
    $choix = read-host -prompt « voulez-vous rejouer ? (oui/non) »
}

```