

Exercice 1

Déplacer les fichiers modifiés il y a plus de 10 jours dans le dossier PLUS

Et les autres dans le dossier MOINS

```
PS C:\Users\ATKWE01> Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_1\TEST_Suppression" | Where-Object {$_.LastWriteTime -lt $date}

Répertoire : C:\Users\ATKWE01\TEMP\Exercice_1\TEST_Suppression

Mode                LastWriteTime      Length Name
----                -----          ---- 
-a----   11/05/2025 10:40            0 Fichier_1.txt
-a----   11/05/2025 10:40            0 Fichier_2.txt
-a----   11/05/2025 10:40            0 Fichier_3.txt
-a----   11/05/2025 10:40            0 Fichier_4.txt
-a----   11/05/2025 10:40            0 Fichier_5.txt
-a----   11/05/2025 10:40            0 Fichier_6.txt
-a----   11/05/2025 10:40            0 Fichier_7.txt

$date = (Get-Date).AddDays(-10)
$date
$liste = Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_1\TEST_Suppression" | Where-Object {$_.LastwriteTime -lt $date} #|Move-Item -Destination "C:\Users\ATKWE01\TEMP\Exercice_1\PLUS"
$liste
Foreach ($ligne in $liste)
{
    write-Host $ligne.FullName
    write-Host "-----"
    Move-Item -Path $ligne.FullName -Destination "C:\Users\ATKWE01\TEMP\Exercice_1\PLUS"
}
```

Exercice 2

- Lister tous les fichiers du répertoire Exercice_2 (et des sous-répertoires)
- Supprimer les fichiers datant de plus de 10 jours

```
Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_2\Dossiers\"  
$date = (Get-Date).AddDays(-10)  
Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_2\Dossiers\" -Recurse -File | Where-Object {$_.LastwriteTime -lt $date} | Remove-Item
```

-Recurse = chemin spécifique (dossier/fichier...)

Exercice 3

- Lister tous les fichiers du dossier "Dossiers" et des sous-dossiers dans une variable.
- Créer une boucle Foreach pour traiter chaque fichier en respectant les consignes ci-dessous :
 - Date de modification > 1 an --> Déplacer le fichier dans le dossier "PLUS_1_AN"
 - Date de modification > 6 mois --> Déplacer le fichier dans le dossier "PLUS_6_MOIS"
 - le reste --> Déplacer le fichier dans le dossier "AUTRES"

```
$date_1an = (Get-Date).AddYears(-1)
$date_6mois = (Get-Date).AddMonths(-6)
$plus_1an = "C:\Users\ATKWE01\TEMP\Exercice_3\PLUS_1_AN"
$plus_6mois = "C:\Users\ATKWE01\TEMP\Exercice_3\PLUS_6_MOIS"
$autres = "C:\Users\ATKWE01\TEMP\Exercice_3\AUTRES"

$liste| = Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_3\Dossiers\" -Recurse -File

Foreach ($ligne in $liste)
{
    $date_modification = $ligne.LastWriteTime

    If ($date_modification -lt $date_1an)
    {
        Move-Item -Path $ligne.FullName -Destination $plus_1an
    }
    Elseif ($date_modification -lt $date_6mois)
    {
        Move-Item -Path $ligne.FullName -Destination $plus_6mois
    }
    Else
    {
        Move-Item -Path $ligne.FullName -Destination $autres
    }
}
```

- Commencer par une ou plusieurs variable(s)

\$datemodification = date d'aujourd'hui

FullName = tout le chemin

Ordre : If → elseif → else

Pas confondre -lt et -gt quand valeur négative pour année/mois

Variable pour chaque valeur → plus pratique

Exercice 4

- Lister tous les fichiers du dossier "Dossiers"
- Déplacer les fichiers débutant par "PIOTR" dans le dossier Piotr
- Supprimer les autres fichiers

```
$Piotr = "C:\Users\ATKWE01\TEMP\Exercice_4\Piotr"

$liste = Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_4\Dossiers\" -Recurse -File

Foreach ($ligne in $liste)
{
    $nom_fichier = $ligne.Name

    If ($nom_fichier -like "PIOTR_*.*")
    {
        Move-Item -Path $ligne.FullName -Destination $Piotr
    }
    Else
    {
        Remove-Item -Path $ligne.FullName -Include *.txt -Exclude *PIOTR*
    }
}
```

FullName = tout le chemin

Name = que le fichier

Exercice 5

- Lister l'ensemble des fichiers.
- Déplacer et renommer les fichiers dans "Sous-dossier_1" (renomme les comme tu veux) --> Action à faire dans la même boucle

```
$liste = Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_5\Dossiers\" -Recurse -File
$fiche_de_table = "C:\Users\ATKWE01\TEMP\Exercice_5\Dossiers\Sous-dossier_1"

Foreach ($ligne in $liste)
{
    Move-Item -Path $ligne.FullName -Destination "$fiche_de_table\comme_tu_veux $($ligne.Basename).txt"
}
```

Entre parenthèses : permet de déplacer et renommez les fichiers

Exercice 6

- Lister l'ensemble des fichiers du dossier DOSSIERS

- Dans une boucle Foreach

- Supprimer les fichiers débutant par "A_SUPPRIMER"

- Déplacer et renommer les fichiers débutant par "A_RENOMMER" en enlevant le "A_RENOMMER" dans le dossier DESTINATION_RENOMMAGE

- Déplacer les fichiers dont la date de modification est plus ancienne de 10 ans dans le dossier DESTINATION_ANCIEN

```
$date_10ans = (Get-Date).AddYears(-10)
$destination_renommage = "C:\Users\ATKWE01\TEMP\Exercice_6\DESTINATION_RENOMMAGE"
$destination_ancien = "C:\Users\ATKWE01\TEMP\Exercice_6\DESTINATION_ANCIEN"

$liste = Get-ChildItem -Path "C:\Users\ATKWE01\TEMP\Exercice_6\DOSSIERS\" -Recurse -File

Foreach ($ligne in $liste)
{
    $nom_du_fichier = $ligne.Name
    $date_modifier = $ligne.LastWriteTime

    If ($nom_du_fichier -like "A_SUPPRIMER_*.txt")
    {
        Remove-Item -Path $ligne.FullName
    }
    Elseif ($nom_du_fichier -like "A_RENOMMER_*.txt")
    {
        $nouveau_nom_fichier = $nom_du_fichier.replace("A_RENOMMER_", "")
        Move-Item -Path $ligne.FullName -Destination "$destination_renommage\$nouveau_nom_fichier"
    }
    Elseif ($date_modifier -lt $date_10ans)
    {
        Move-Item -Path $ligne.FullName -Destination $destination_ancien
    }
}
```

A modifier

Renommage avec .replace("caractère(s) à enlever" , "")

Dossier de destination\nouveau nom du fichier

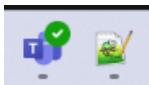
Full.Name tout le temps !

- Variables sans caractères spéciaux (par exemple : date_modifié → date_modifier/date_modifie)

Exercice 7

- Ouvrir NOTEPAD / Bloc note
- Tuer le process via PowerShell pour fermer l'application.

Avant :



```
Get-Process -Name "notepad++" | Stop-Process
```

Après :



Pour vérifier la liste de processus, il faut faire la commande Get-Process

Notes pour Get-Process

- Handles : nombre de handles que le processus a ouverts.
- NPM(K) : quantité de mémoire non paginée utilisée par le processus, en kilo-octets.
- PM(K) : quantité de mémoire paginable utilisée par le processus, en kilo-octets.
- WS(K) : taille du jeu de travail du processus, en kilo-octets. La plage de travail inclut les pages de mémoire récemment référencées par le processus.
- VM(M) : quantité de mémoire virtuelle utilisée par le processus, en mégaoctets. La mémoire virtuelle inclut le stockage dans les fichiers de pagination sur le disque.
- Processeur(s) : durée du processeur utilisée par le processus sur tous les processeurs, en secondes.
- ID : ID de processus (PID) du processus.
- ProcessName : nom du processus. Pour obtenir des informations sur les concepts liés aux processus, consultez le Glossaire du Centre d'aide et de support et l'Aide du Gestionnaire des tâches.

Exercice 8

Lister les processus en cours et mettre le contenu dans un fichier txt

```
Get-Process | Out-File -FilePath "C:\Users\ATKWE01\TEMP\Test.txt"
```

Envoie la sortie à un fichier : *Out-File*

Exercice 9

Lister les services en cours d'exécution et mettre le contenu dans un fichier txt

```
Get-Service | Where-Object { $_.Status -eq "Running" } | Out-File -FilePath "C:\Users\ATKWE01\TEMP\Tests.txt"
```

En cours d'exécution : *Running*

Exercice 10

- Vérifier si le fichier "C:\Temp\Fichier.txt" existe
- Si non, créer le fichier.
- Faire une boucle For permettant d'ajouter 10 lignes dans le fichier txt
 - Ceci est la ligne 1
 - Ceci est la ligne 2
 - ...
 - Ceci est la ligne 10

```
Test-Path "C:\temp\Fichier.txt"
New-Item -Path "C:\temp" -Name "Fichier.txt" -ItemType File
$valeur = 1..10
$fichier_texte = "C:\temp\Fichier.txt"
For ($i=0 ; $i -lt 10 ; $i++)
{
    Add-Content -Path $fichier_texte -Value "Ceci est la ligne $($valeur[$i])"
```

Add-Content = Ajouter du contenu

Set-Content = remplacer du contenu

Exercice 12

- Lister et mettre dans une variable l'ensemble des fichiers présents dans le dossier DOSSIERS
- Utiliser SWITCH pour :
 - Déplacer tous les fichiers commençant par PIOTR dans le dossier PIOTR
 - Déplacer tous les fichiers commençant par STEVEN dans le dossier STEVEN

- Déplacer tous les fichiers commençant par GERARD dans le dossier GERARD
- Déplacer tous les fichiers commençant par ALBERT dans le dossier ALBERT
- Déplacer tous les fichiers commençant par PAUL dans le dossier PAUL
- Compter (dans PowerShell :)) le nombre de fichiers pour chaque prénom et faire afficher ceci dans la fenêtre Powershell
 - Il y a NOMBRE fichiers PIOTR
 - Il y a NOMBRE fichiers GERARD
 - Il y a NOMBRE fichiers STEVEN
 - Il y a NOMBRE fichiers PAUL
 - Il y a NOMBRE fichiers ALBERT

Qu'un fichier pour chaque nom

```
# Mettre le dossier source dans une variable
$source = "C:\Users\ATKWE01\TEMP\Exercice_12\DOSSIERS\"

# Lister tous les fichiers des dossiers et sous-dossiers
$liste = Get-ChildItem $source -Recurse -File

# Mettre les dossiers destination dans des variables
$piotr_ = "C:\Users\ATKWE01\TEMP\Exercice_12\PIOTR"
$steven = "C:\Users\ATKWE01\TEMP\Exercice_12\STEVEN"
$gerard = "C:\Users\ATKWE01\TEMP\Exercice_12\GERARD"
$albert = "C:\Users\ATKWE01\TEMP\Exercice_12\ALBERT"
$paul = "C:\Users\ATKWE01\TEMP\Exercice_12\PAUL"
$autres = "C:\Users\ATKWE01\TEMP\Exercice_12\AUTRES"

# Initialiser les compteurs
$compteur = 0
$compteurPiotr = 0
$compteurSteven = 0
$compteurGerard = 0
$compteurAlbert = 0
$compteurPaul = 0
$compteurAutres = 0
```

```

# Faire une boucle Foreach pour traiter les fichiers un à un
Foreach ($ligne in $liste) {
    # Récupérer le nom du fichier
    $name = $ligne.Name
    $compteur++

    # Déplacer le fichier dans son dossier correspondant
    Switch ($name) {
        "PIOTR.txt" {
            Move-Item -Path $ligne.FullName -Destination $Piotr_ -Force
            $compteurPiotr++
        }
        "STEVEN.txt" {
            Move-Item -Path $ligne.FullName -Destination $Steven -Force
            $compteurSteven++
        }
        "GERARD.txt" {
            Move-Item -Path $ligne.FullName -Destination $Gerard -Force
            $compteurGerard++
        }
        "ALBERT.txt" {
            Move-Item -Path $ligne.FullName -Destination $Albert -Force
            $compteurAlbert++
        }
        "PAUL.txt" {
            Move-Item -Path $ligne.FullName -Destination $Paul -Force
            $compteurPaul++
        }
        Default {
            Move-Item -Path $ligne.FullName -Destination $Autres -Force
            $compteurAutres++
        }
    }
}

# Afficher le nombre de fichiers par nom
Write-Host "Il y a $compteurPiotr fichiers PIOTR.txt"
Write-Host "Il y a $compteurSteven fichiers STEVEN.txt"
Write-Host "Il y a $compteurGerard fichiers GERARD.txt"
Write-Host "Il y a $compteurAlbert fichiers ALBERT.txt"
Write-Host "Il y a $compteurPaul fichiers PAUL.txt"
Write-Host "Il y a $compteurAutres fichiers dans AUTRES."

```

```

PS C:\Users\ATKWE01> C:\Users\ATKWE01\Documents\PowerShell\Exos PS\exo 12 11.ps1
Il y a 1 fichiers PIOTR.txt
Il y a 1 fichiers STEVEN.txt
Il y a 1 fichiers GERARD.txt
Il y a 1 fichiers ALBERT.txt
Il y a 1 fichiers PAUL.txt
Il y a 1 fichiers dans AUTRES.

```

Les commentaires servent à s'organiser dans le script et à mieux comprendre ce qu'on fait.

D'abord j'ai réinitialisé le dossier Exercice 12 avec les fichiers, puis j'ai exécuté le script.

Exercice 11

- Lister et mettre dans une variable l'ensemble des fichiers présents dans le dossier DOSSIERS

- Utiliser SWITCH pour :

- Déplacer tous les fichiers commençant par PIOTR dans le dossier PIOTR

- Déplacer tous les fichiers commençant par STEVEN dans le dossier STEVEN

- Déplacer tous les fichiers commençant par GERARD dans le dossier GERARD

- Déplacer tous les fichiers commençant par ALBERT dans le dossier ALBERT

- Déplacer tous les fichiers commençant par PAUL dans le dossier PAUL

- Compter (dans PowerShell :)) le nombre de fichiers pour chaque prénom et faire afficher ceci dans la fenêtre Powershell

- Il y a NOMBRE fichiers PIOTR

- Il y a NOMBRE fichiers GERARD

- Il y a NOMBRE fichiers STEVEN

- Il y a NOMBRE fichiers PAUL

- Il y a NOMBRE fichiers ALBERT

Ici c'est la même chose que l'exercice 12, mais il y a plusieurs fichiers

```
# Mettre le dossier source dans une variable
$source_ = "C:\Users\ATKWE01\TEMP\Exercice_11\DOSSIERS\"

# Lister tous les fichiers des dossiers et sous-dossiers
$liste_ = Get-ChildItem $source_ -Recurse -File

# Mettre les dossiers destination dans des variables
$piotr_ = "C:\Users\ATKWE01\TEMP\Exercice_11\PIOTR"
$steven_ = "C:\Users\ATKWE01\TEMP\Exercice_11\STEVEN"
$gerard_ = "C:\Users\ATKWE01\TEMP\Exercice_11\GERARD"
$albert_ = "C:\Users\ATKWE01\TEMP\Exercice_11\ALBERT"
$paul_ = "C:\Users\ATKWE01\TEMP\Exercice_11\PAUL"

# Initialiser les compteurs
$compteur_ = 0
$compteurPiotr_ = 0
$compteurSteven_ = 0
$compteurGerard_ = 0
$compteurAlbert_ = 0
$compteurPaul_ = 0
```

```

# Faire une boucle Foreach pour traiter les fichiers un à un
Foreach ($lignes in $liste) {
    # Récupérer le nom du fichier
    $nom = $lignes.Name
    $compteur_++

    # Déplacer le fichier dans son dossier correspondant avec Switch
    Switch -Wildcard ($nom) {
        { $nom -like "PIOTR_*.txt" } {
            Move-Item -Path $lignes.FullName -Destination $Piotr_ -Force
            $compteurPiotr_++
        }
        { $nom -like "STEVEN_*.txt" } {
            Move-Item -Path $lignes.FullName -Destination $Steven_ -Force
            $compteurSteven_++
        }
        { $nom -like "GERARD_*.txt" } {
            Move-Item -Path $lignes.FullName -Destination $Gerard_ -Force
            $compteurGerard_++
        }
        { $nom -like "ALBERT_*.txt" } {
            Move-Item -Path $lignes.FullName -Destination $Albert_ -Force
            $compteurAlbert_++
        }
        { $nom -like "PAUL_*.txt" } {
            Move-Item -Path $lignes.FullName -Destination $Paul_ -Force
            $compteurPaul_++
        }
        Default {
            Write-Host "Fichier non déplacé : $nom" # Afficher les fichiers non traités
        }
    }
}

# Afficher le nombre de fichiers par nom
Write-Host "Il y a $compteurPiotr_ fichiers PIOTR_*.txt"
Write-Host "Il y a $compteurSteven_ fichiers STEVEN_*.txt"
Write-Host "Il y a $compteurGerard_ fichiers GERARD_*.txt"
Write-Host "Il y a $compteurAlbert_ fichiers ALBERT_*.txt"
Write-Host "Il y a $compteurPaul_ fichiers PAUL_*.txt"

```

```

PS C:\Users\ATKWE01> C:\Users\ATKWE01\Documents\PowerShell\Exos PS\exo 12 12.ps1
Il y a 100 fichiers PIOTR_*.txt
Il y a 100 fichiers STEVEN_*.txt
Il y a 100 fichiers GERARD_*.txt
Il y a 100 fichiers ALBERT_*.txt
Il y a 100 fichiers PAUL_*.txt

```

On utilise l'opérateur `-like` qui permet de rechercher une ressemblance entre deux valeurs.

« * » = utilisé pour n'importe quel caractère

Default : s'il n'y a pas le fichier demandé

Exercice 13

- Des fichiers logs sont dans le dossier "Fichiers".

Dans chaque fichier il peut y avoir des lignes "INFORMATION", des lignes "ERROR" ou des lignes "WARNING".

Je souhaite faire afficher pour chaque fichier log l'information suivante :

(Texte à afficher)

- Analyse du fichier NOM_FICHIER (Ecriture en jaune dans PowerShell)

- Il y a NOMBRE ligne(s) INFORMATION
- Il y a NOMBRE ligne(s) ERROR (Ecriture en Rouge)
 - S'il y a une ou plusieurs lignes ERROR, il faut l'afficher.
- Il y a NOMBRE ligne(s) WARNING

Avec Foreach :

```
# Mettre le dossier dans une variable
$logs = "C:\Users\ATKWE01\TEMP\Exercice_13\Fichiers"
# Lister tous les fichiers du dossier
$liste = Get-ChildItem $logs

# Faire une boucle Foreach pour traiter chaque fichier
Foreach ($fichier in $liste) {
    # Lire le contenu du fichier
    $contenu = Get-Content -Path $fichier.FullName

    # Initialiser les compteurs
    $compteurInfo = 0
    $compteurError = 0
    $compteurWarning = 0

    # Analyser chaque ligne du contenu
    foreach ($ligne in $contenu) {
        if ($ligne -like "*INFO*") {
            $compteurInfo++
        }
        elseif ($ligne -like "*ERROR*") {
            $compteurError++
        }
        elseif ($ligne -like "*WARNING*") {
            $compteurWarning++
        }
    }

    # Afficher les résultats
    Write-Host "Analyse du fichier $($fichier.Name)" -ForegroundColor Yellow
    Write-Host "Il y a $compteurInfo ligne(s) INFORMATION"
    Write-Host "Il y a $compteurError ligne(s) ERROR" -ForegroundColor Red
    Write-Host "Il y a $compteurWarning ligne(s) WARNING"
    Write-Host "" # Ligne vide pour séparer les résultats des fichiers
}
```

Résultat :

```
Analyse du fichier Log1.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du Fichier Log10.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log11.txt
Il y a 3 ligne(s) INFORMATION
Il y a 1 ligne(s) ERROR
Il y a 2 ligne(s) WARNING

Analyse du fichier Log12.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log13.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du Fichier Log14.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log15.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du fichier Log16.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log17.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du fichier Log18.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du Fichier Log19.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log2.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du fichier Log20.txt
Il y a 3 ligne(s) INFORMATION
Il y a 1 ligne(s) ERROR
Il y a 2 ligne(s) WARNING

Analyse du fichier Log3.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log4.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du Fichier Log5.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du Fichier Log6.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log7.txt
Il y a 3 ligne(s) INFORMATION
Il y a 3 ligne(s) ERROR
Il y a 0 ligne(s) WARNING

Analyse du fichier Log8.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING

Analyse du fichier Log9.txt
Il y a 3 ligne(s) INFORMATION
Il y a 2 ligne(s) ERROR
Il y a 1 ligne(s) WARNING
```

Log : fichier de script avec les erreurs

Exercice 14

- Lister les fichiers du dossier "Dossiers" et des sous-dossiers
- Déplacer les fichiers dans leur répertoires respectifs (extension .xlsx dans le répertoire XLSX,...)

```
# Lister tous les fichiers du dossier
$liste = Get-ChildItem "C:\Users\ATKWE01\TEMP\Exercice_14\Dossiers" -Recurse -File

# Mettre les dossiers destination dans des variables
$docx = "C:\Users\ATKWE01\TEMP\Exercice_14\DOCX"
$img = "C:\Users\ATKWE01\TEMP\Exercice_14\JPG"
$pdf = "C:\Users\ATKWE01\TEMP\Exercice_14\PDF"
$txt = "C:\Users\ATKWE01\TEMP\Exercice_14\TXT"
$xlsx = "C:\Users\ATKWE01\TEMP\Exercice_14\XLSX"

# Créer une boucle Foreach
foreach ($ligne in $liste) {
    # Extraire l'extension du fichier sans le point
    $ext = $ligne.Extension.ToLower() # Convertir en minuscules pour assurer la correspondance

    # Déterminer le dossier de destination en fonction de l'extension
    switch ($ext) {
        ".docx" { Move-Item -Path $ligne.FullName -Destination $docx -Force }
        ".jpg" { Move-Item -Path $ligne.FullName -Destination $img -Force }
        ".pdf" { Move-Item -Path $ligne.FullName -Destination $pdf -Force }
        ".txt" { Move-Item -Path $ligne.FullName -Destination $txt -Force }
        ".xlsx" { Move-Item -Path $ligne.FullName -Destination $xlsx -Force }
        default { write-Host "Aucun dossier défini pour l'extension : $ext" -ForegroundColor Yellow } # Afficher un message si l'extension n'a pas de dossier de destination
    }
}

# Afficher un message de confirmation à la fin
write-Host "Opération terminée. Consultez le fichier log à : $log" -ForegroundColor Green
```

Créer des fichiers log :

D'abord créer le chemin du fichier log, ensuite créer le fichier log avec New-Item

Et avec Add-Content on ajoute du contenu (comme on veut) dans le fichier log

```
# Lister tous les fichiers du dossier
$liste = Get-ChildItem "C:\Users\ATKWE01\TEMP\Exercice_14\Dossiers" -Recurse -File

# Mettre les dossiers destination dans des variables
$docx = "C:\Users\ATKWE01\TEMP\Exercice_14\DOCX"
$img = "C:\Users\ATKWE01\TEMP\Exercice_14\JPG"
$pdf = "C:\Users\ATKWE01\TEMP\Exercice_14\PDF"
$txt = "C:\Users\ATKWE01\TEMP\Exercice_14\TXT"
$xlsx = "C:\Users\ATKWE01\TEMP\Exercice_14\XLSX"

# Créer le chemin du fichier log
$log = "C:\Users\ATKWE01\TEMP\Exercice_14\log_exo14.txt"

# Créer le fichier log s'il n'existe pas déjà
New-Item -ItemType File -Path $log | Out-Null

# Créer une boucle Foreach
foreach ($ligne in $liste) {
    # Extraire l'extension du fichier sans le point
    $ext = $ligne.Extension.ToLower() # Convertir en minuscules pour assurer la correspondance

    # Déterminer le dossier de destination en fonction de l'extension
    switch ($ext) {
        ".docx" {
            Move-Item -Path $ligne.FullName -Destination $docx -Force
            Add-Content -Path $log -Value "Déplacé : $($ligne.FullName) vers $docx"
        }
        ".jpg" {
            Move-Item -Path $ligne.FullName -Destination $img -Force
            Add-Content -Path $log -Value "Déplacé : $($ligne.FullName) vers $img"
        }
        ".pdf" {
            Move-Item -Path $ligne.FullName -Destination $pdf -Force
            Add-Content -Path $log -Value "Déplacé : $($ligne.FullName) vers $pdf"
        }
        ".txt" {
            Move-Item -Path $ligne.FullName -Destination $txt -Force
            Add-Content -Path $log -Value "Déplacé : $($ligne.FullName) vers $txt"
        }
        ".xlsx" {
            Move-Item -Path $ligne.FullName -Destination $xlsx -Force
            Add-Content -Path $log -Value "Déplacé : $($ligne.FullName) vers $xlsx"
        }
        default {
            Write-Host "Aucun dossier défini pour l'extension : $ext" -ForegroundColor Yellow
            Add-Content -Path $log -Value "Aucun dossier défini pour l'extension : $ext pour le fichier $($ligne.FullName)"
        }
    }
}

# Afficher un message de confirmation à la fin
write-Host "Opération terminée. Consultez le fichier log à : $log" -ForegroundColor Green
```

```
Début du script : 06/12/2025 10:05:23
Opération terminée. Consultez le fichier log à : C:\Users\ATKWE01\TEMP\Exercice_14\log_exo14.txt
Fin du script : 06/12/2025 10:05:24
Durée d'exécution : 00:00:00.3573202
```

Out-Null sert à masquer la sortie de la commande et à ne pas afficher les éléments

Mission 1

Dans le dossier "Dossiers", nous avons plusieurs sous-dossiers.

Nous avons aussi un fichier csv recensant 10 dossiers parmi les sous-dossiers.

L'objectif est de purger les sous-dossiers de la façon suivante :

- Si le sous-dossier est référencé dans le fichier csv, il faut alors supprimer tous les fichiers de ce sous-dossier qui ont une date de modification supérieur à (Aujourd'hui - la colonne date de la ligne).
- Si le dossier n'est pas référencé dans le fichier CSV, il faut supprimer tous les fichiers de ce sous-dossier qui ont une date de modification supérieure à 1 an.

Ecrire des logs pour chaque traitement pour pouvoir suivre et analyser le script

Tableau CSV:

Chemin	Filtre	Date
C:\Temp\Exercice_15\Dossiers\Dossier52\	*.csv	40
C:\Temp\Exercice_15\Dossiers\Dossier24\	*.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier18\	*.csv	15
C:\Temp\Exercice_15\Dossiers\Dossier49\	*.txt	40
C:\Temp\Exercice_15\Dossiers\Dossier35\	*.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier13\	*.*	15
C:\Temp\Exercice_15\Dossiers\Dossier25\	*.csv	40
C:\Temp\Exercice_15\Dossiers\Dossier73\	*.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier93\	*.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier90\	*.csv	15

Contenu du fichier CSV :

```
PS C:\Users\ATKWE01> $folder
Chemin                               Filtre Date
----- -----
```

C:\Temp\Exercice_15\Dossiers\Dossier52\ *.csv	40
C:\Temp\Exercice_15\Dossiers\Dossier24\ *.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier18\ *.csv	15
C:\Temp\Exercice_15\Dossiers\Dossier49\ *.txt	40
C:\Temp\Exercice_15\Dossiers\Dossier35\ *.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier13\ *.*	15
C:\Temp\Exercice_15\Dossiers\Dossier25\ *.csv	40
C:\Temp\Exercice_15\Dossiers\Dossier73\ *.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier93\ *.txt	15
C:\Temp\Exercice_15\Dossiers\Dossier90\ *.csv	15

Script pour afficher le contenu du fichier CSV :

```
# Chemin du fichier CSV
$csv = "C:\temp\Exercice_15\ListeDossiers.csv"

# Lire le contenu du fichier CSV
$dossiers = Import-Csv -Path $csv -Delimiter ";"

Foreach ($ligne in $dossiers) {
    # Accéder au chemin du dossier pour chaque ligne
    $dossiers_chemin = $ligne.Chemin

    If (Test-Path $dossiers_chemin) {
        Write-Host "Contenu du dossier : $dossiers_chemin"

        # Obtenir et afficher les fichiers du dossier
        $fichiers = Get-ChildItem -Path $dossiers_chemin

        If ($fichiers.Count -eq 0) {
            Write-Host "Le dossier est vide."
        } Else {
            Foreach ($fichier in $fichiers) {
                Write-Host "Fichier trouvé : $($fichier.FullName)"
            }
        }
    } Else {
        Write-Host "Le dossier n'existe pas : $dossiers_chemin"
    }
}
```

Résultat :

```
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier52\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier52\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier52\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier52\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier24\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier24\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier24\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier24\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier18\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier18\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier18\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier18\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier49\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier49\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier49\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier49\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier35\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier35\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier35\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier35\FichierUnMois.txt

Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier13\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier13\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier13\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier13\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier25\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier25\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier25\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier25\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier73\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier73\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier73\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier73\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier93\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier93\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier93\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier93\FichierUnMois.txt
Contenu du dossier : C:\Temp\Exercice_15\Dossiers\Dossier90\
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier90\FichierAujourdHui.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier90\FichierUnAn.txt
Fichier trouvé : C:\Temp\Exercice_15\Dossiers\Dossier90\FichierUnMois.txt
```

Initialiser les variables dès l'entrée du foreach

Variables pour chaque élément du tableau

Script avec le fichier CSV :

```
# Lister les fichiers du dossier
$liste = Get-ChildItem "C:\temp\Exercice_15\Dossiers" -Recurse -File

# Créer des variables pour chaque date
$date_1_an = (Get-Date).AddYears(-1)
$date_ = Get-Date

# Créer le chemin du fichier log
$logFile = "C:\temp\Exercice_15\log_exo15.log"

# Créer le fichier log s'il n'existe pas
New-Item -ItemType File -Path $LogFile | Out-Null

# Chemin du fichier CSV
$csv = "C:\temp\Exercice_15>ListeDossiers.csv"

# Lire le contenu du fichier CSV
$dossiers = Import-Csv -Path $csv -Delimiter ";" 

# Début du script
Add-Content -Path $LogFile -Value "Début du script : $(Get-Date)"

Foreach ($ligne in $dossiers) {
    # Accéder au chemin du dossier pour chaque ligne
    $dossiers_chemin = $ligne.Chemin
    $filtre = $ligne.Filtre
    $date = $ligne.Date

    If (Test-Path $dossiers_chemin) {
        $fichiers = Get-ChildItem -File -Path $dossiers_chemin -Filter $filtre | Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-$date) }

        If ($fichiers) {
            Foreach ($fichier in $fichiers) {
                Remove-Item $fichier.FullName
                Add-Content -Path $LogFile -Value "Suppression du fichier $($fichier.FullName)"
            }
        }
        <#
        Write-Host "Contenu du dossier : $dossiers_chemin"
        # Obtenir et afficher les fichiers du dossier
        Get-ChildItem -File -Path $dossiers_chemin -Filter $filtre | Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-$date) } | Out-File -FilePath $LogFile | Remove-Item
        Write-Host "Fichier supprimé : $($fichier.FullName)"#
    }
    else {
        Write-Host "Le dossier n'existe pas : $dossiers_chemin"
    }
}

write-Host "Purge des fichiers terminée. Consultez le fichier log à : $LogFile" -ForegroundColor Magenta
# Fin du script
Add-Content -Path $LogFile -Value "Fin du script : $(Get-Date)"
```

Script avec les fichiers non référencés :

```

# Lister les dossiers
$liste_ = Get-ChildItem "C:\temp\Exercice_15\Dossiers"

# Créer le chemin du fichier log
$logFile = "C:\temp\Exercice_15\log_exo15.log"

# Chemin du fichier CSV
$csv = Import-Csv -Path "C:\temp\Exercice_15\ListeDossiers.csv" -Delimiter ";"

# Début du script
Add-Content -Path $LogFile -Value "Début du script : $(Get-Date)"

#Foreach ($dossier in $liste_) {
#    # Accéder au chemin du dossier pour chaque ligne
#    $dossiers_chemin = $dossier.FullName

#    # Vérifier si le dossier est dans le CSV
#    $b = ""
#    $b = $csv | Where-Object { $_.Chemin -eq "$dossiers_chemin" }

#    # Identifier si le dossier est présent dans le CSV
#    $presentDansCSV = If ($b) { "OUI" } else { "NON" }

#    # Journaliser l'analyse du dossier
#    Add-Content -Path $LogFile -Value "---- Analyse du dossier $($dossier.Name) ----"
#    Add-Content -Path $LogFile -Value "Présent dans le fichier CSV : $presentDansCSV"
#    Write-Host "---- Analyse du dossier $($dossier.Name) ----"
#    Write-Host "Présent dans le fichier CSV : $presentDansCSV"

#    If ($dossiers_chemin -in $csv) {
#        $filtre = $b.Filtre
#        $date = $b.Date

#        # Obtenir les fichiers correspondant au filtre et vérifiant la date
#        $fichiers = Get-ChildItem -File -Path $dossiers_chemin -Filter $filtre | Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-$date) }

#        If ($fichiers.Count -gt 0) {
#            foreach ($abc in $fichiers) {
#                Remove-Item -Path $abc.FullName -Force
#                Add-Content -Path $LogFile -Value "Fichier supprimé : $($abc.Name)"
#            }
#        }
#        Else {
#            # Obtenir et supprimer les fichiers vieux d'un an
#            $fichiersASupprimer = Get-ChildItem -Path $dossiers_chemin -File | Where-Object { $_.LastWriteTime -lt (Get-Date).AddYears(-1) }

#            If ($fichiersASupprimer.Count -gt 0) {
#                foreach ($fichier in $fichiersASupprimer) {
#                    Remove-Item -Path $fichier.FullName -Force
#                    Add-Content -Path $LogFile -Value "Fichier supprimé : $($fichier.Name)"
#                }
#            }
#        }
#    }
#    # Journaliser la fin de l'analyse
#    Add-Content -Path $LogFile -Value "Suppression terminée"
#}

Write-Host "Purge des fichiers terminée. Consultez le fichier log à : $LogFile" -ForegroundColor Magenta

# Fin du script
Add-Content -Path $LogFile -Value "Fin du script : $(Get-Date)"

```

Try-Catch → intéressant pour le Remove-Item par exemple

Exercice 16

- Avec PowerShell, utiliser 7z pour décompresser le fichier "Fichiers.zip" --> Le mot de passe est "123456789".
- Une fois ok, rechercher les fichiers dont le nom est dans le fichier "Liste.csv"
- Déplacer les fichiers concernés dans leur dossier respectif (fichier 00020 dans le dossier 00020).

Première version du script :

```
# Chemin du programme
$sevenZip = "C:\Program Files\7-zip\7z.exe"

# Chemin du fichier ZIP
$chemin = "C:\Users\ATKWE01\TEMP\Exercice_16\Fichiers.zip"

# chemin vers le dossier destination
$destination = "C:\Users\ATKWE01\TEMP\Exercice_16"

# Mot de passe
$password = "123456789"

# Décompresser le dossier
Expand-Archive -LiteralPath $chemin -DestinationPath $destination

# Lister les fichiers
$liste = Get-ChildItem "C:\Users\ATKWE01\TEMP\Exercice_16\Fichiers" -Recurse -File

# Mettre les dossiers destination
$20 = "C:\Users\ATKWE01\TEMP\Exercice_16\00020"
$40 = "C:\Users\ATKWE01\TEMP\Exercice_16\00040"
$60 = "C:\Users\ATKWE01\TEMP\Exercice_16\00060"
$80 = "C:\Users\ATKWE01\TEMP\Exercice_16\00080"
$100 = "C:\Users\ATKWE01\TEMP\Exercice_16\00100"

# créer le chemin du fichier log
$logFile = "C:\Users\ATKWE01\TEMP\Exercice_16\log16.log"

# créer le fichier log
New-Item -ItemType File -Path $LogFile | Out-Null

# chemin vers le fichier CSV
$csv = "C:\Users\ATKWE01\TEMP\Exercice_16>Liste.csv"

# Lire le contenu du fichier CSV
$dossiers = Import-Csv -Path $csv -Delimiter ";"

# Début du script
Add-Content -Path $LogFile -Value "Début du script : $(Get-Date)"

foreach ($ligne in $dossiers) {
    # Récupérer le nom du fichier
    $nom = $ligne.Fichiers

    switch ($nom) {
        "00020.pdf" {
            Move-Item -Path $liste.FullName -Destination $20 -Force
            Add-Content -Path $LogFile -Value "Déplacé : $($ligne.Fichiers) vers $20"
        }
        "00040.pdf" {
            Move-Item -Path $liste.FullName -Destination $40 -Force
            Add-Content -Path $LogFile -Value "Déplacé : $($ligne.Fichiers) vers $40"
        }
        "00060.pdf" {
            Move-Item -Path $liste.FullName -Destination $60 -Force
            Add-Content -Path $LogFile -Value "Déplacé : $($ligne.Fichiers) vers $60"
        }
        "00080.pdf" {
            Move-Item -Path $liste.FullName -Destination $80 -Force
            Add-Content -Path $LogFile -Value "Déplacé : $($ligne.Fichiers) vers $80"
        }
        "00100.pdf" {
            Move-Item -Path $liste.FullName -Destination $100 -Force
            Add-Content -Path $LogFile -Value "Déplacé : $($ligne.Fichiers) vers $100"
        }
        default {
            Write-Host "Aucun fichier défini" -ForegroundColor Yellow
            Add-Content -Path $LogFile -Value "Aucun dossier défini pour le nom : $nom"
        }
    }
}

write-Host "Déplacement des fichiers terminé. Voir le fichier log à : $LogFile" -ForegroundColor Green

# Fin du script
Add-Content -Path $LogFile -Value "Fin du script : $(Get-Date)"
```