

TP 1 PowerShell : Fichiers Textes

Table of Contents

Travail à faire	1
Exercice 1	1
Exercice 2	1
Exercice 3	2
Exercice 4	3
Exercice 5	3

- **Objectif** : Exercices sur les fichiers textes

Travail à faire

Exercice 1

Écrire un script nommé :`exo1.ps1`

Ce script doit afficher les données de ce fichier sous cette forme (avec la commande `Write-Host`) :

Nom : Regnier Prénom : Daniel

Nom : Ravrat Prénom : Emmanuel

Nombre de nom(s) : 2



Le nombre de noms doit être affiché à la fin.

Exercice 2

Reprendre le script `exercice1_et_2.ps1` de l'**exercice 1** en ajoutant un test d'existence du fichier avant la lecture.

Si le fichier n'existe pas, le traitement affiche le message suivant : *Fichier exo1.txt non trouvé.*

Tester les deux cas.

Corrigé 1 et 2 :

```
# Script qui affiche les données du fichier exo1.txt
```

```
# Le nombre de noms doit être affiché à la fin
# Avec test d'existence du fichier

$fichier="exo1.txt"

if (Test-Path $fichier){
    $collignes=Get-Content $fichier
    $cpt=0
    foreach($ligne in $collignes){
        $tabCompte=$ligne.Split(";")
        Write-Host "Nom : " $tabCompte[0] " Prénom : " $tabCompte[1]
        $cpt++
    }

    Write-Host "Nombre de nom(s) : $cpt"
}

else{
    Write-Host "Fichier $fichier non trouvé"
}
}
```

Exercice 3

En utilisant les tests précédents, écrire un script nommé **exercice3.ps1**, qui demande à l'utilisateur une information et qui enregistre cette information et la date de saisie dans un fichier nommé **test2.log**.



La date doit être placée avant l'information.

Corrigé 3 :

```
# script qui demande à l'utilisateur une information
# enregistre cette information et la date de saisie dans un fichier nommé
test2.log

$fichier = "C:\TpPowerShell\test2.log"
$date=get-date -Format "dd/MM/yyyy"

$reponse=Read-Host "Saisir une information : "

Add-Content $fichier $date
Add-Content $fichier $reponse

# autre solution sur 1 ligne : Add-Content $fichier "$date $reponse"
```

Exercice 4

Écrire un nouveau script nommé `exercice4.ps1`.

Ce script recherche le mot erreur dans le fichier `audit.log` et affiche un message à chaque fois qu'il le trouve.

Corrigé 4 :

```
#recherche ligne par ligne d'une chaine dans un fichier texte
#affiche un message à chaque fois

$fichier="c:\TpPowershell\audit.log"

#$recherche=Read-Host "Saisir une information : "

$recherche = "erreur"

if (Test-Path $fichier){
    $collignes=Get-Content $fichier

    foreach($ligne in $collignes){

        if ($ligne.Contains($recherche)){
            Write-Host "$recherche est trouvé"
        }
    }
}

else{
    Write-Host "Fichier $fichier non trouvé"
}
```

Exercice 5

Écrire un nouveau script nommé `exercice5.ps1`.

Ce script fait la même recherche que le script `exercice4.ps1`, mais il affiche seulement à la fin du traitement le nombre de fois où le mot erreur est trouvé dans le fichier (mot : 1 par ligne).

Corrigé 5 :

```
#recherche ligne par ligne d'une chaine dans un fichier texte
#affiche un message à chaque fois
```

```
$fichier="c:\TpPowershell\audit.log"

#$recherche=Read-Host "Saisir une information : "

$recherche = "erreur"

if (Test-Path $fichier){
    $collignes=Get-Content $fichier

    foreach($ligne in $collignes){

        if ($ligne.Contains($recherche)){
            Write-Host "$recherche est trouvé"
        }
    }
}

else{
    Write-Host "Fichier $fichier non trouvé"
}
```