**Théorie Scripts PowerShell**

# Gestion d’un système d’exploitation

## **Définition**

**Shell**: couche logicielle qui sert d’interface au système d’exploitation. S’applique pour tous les systèmes d’exploitation.

Cette couche forme la coquille du OS, d’où son nom.

Cmd et Powershell sont des shells. Linux a... Shell (ou bash). Les interfaces graphiques le sont aussi, mais on les appelle rarement comment tel.

Powershell: interface shell sous forme de lignes de commandes et un langage de script (possibilité d’intégrer des variables, des boucles, des tests, …aux lignes de commandes)

Powershell vise à remplacer cmd. Il vise l’administration de système et fait très bien son travail dans cette optique.

**1. Les variables en Powershell**

1. Les noms de variables sont précédés d’un signe de dollar. Ex: $Variable1, $Nombre…
2. Affectation: $MaVariable = 5
3. Affichage: $MaVariable
4. Écrire du texte: Write-Host “Ma variable vaut “ $MaVariable
5. Les minuscules et les majuscules n’importent pas (insensibilité à la case).

$mavariable = $MaVariable = $MAVARIABLE

1. Les variables dans Powershell sont faiblement typées. Cela signifie que contrairement au C++, l’information que la variable contient définit son type (pas de int, float, char, etc..)

$Var1 = “Marie“

$Var1

$Var1 = $Var1 + “ Pierre“

$Var1

$Var1 = 12

$Var1

Résultat :

Marie

Marie Pierre

12

## **2. Les expressions conditionnelles**

Tableau des équivalences PowerShell avec C++

-lt <

-le <=

-gt >

-ge >=

-eq ==

-ne !=

-not !

-and &&

-or   ||

On peut mettre le résultat d’une expression conditionnelle (vrai ou faux) dans une variable :

$Comparaison = 5 -eq 5

Write-Host "Est-ce que 5 est égal à 5? Réponse:" $Comparaison

Résultat :

Est-ce que 5 est égal à 5? Réponse: True

----------------------------------------------------------------------------------------------------------------------------------------------------------------

## **3. Boucle for**

Syntaxe : get-help about\_for

Syntax

The following shows the For statement syntax.

for (<init>; <condition>; <repeat>)

{<statement list>}

Exemple1 :

for ($i=1; $i -le 10; $i=$i+2)

{

$i

}

Résultat:

1

3

5

7

9

Exemple2 : (on peut concaténer une variable dans une chaine de caractères )

for ($i=1; $i -le 5; $i++)

{

Write-Host "Tour de boucle $i"

}

Résultat:

Tour de boucle 1

Tour de boucle 2

Tour de boucle 3

Tour de boucle 4

Tour de boucle 5

## **4. Boucle while**

Syntaxe : get-help about\_while

The following shows the While statement syntax:

while (<condition>){<statement list>}

Exemple :

$i = 1

While ($i -le 10)

{

$i

$i = $i+2

}

Résultat:

1

3

5

7

9

## **5. Instruction conditionnelle if**

Syntax

The following example shows the If statement syntax:

if (<test1>)

{<statement list 1>}

[elseif (<test2>)

{<statement list 2>}]

[else

{<statement list 3>}]

## Exemple 1:

for ($i=0; $i -le 10; $i++)

{

$Parite = $i%2 -eq 0

if($Parite)

{

Write-Host "$i est pair"

}

else

{

Write-Host "$i est impair"

}

}

Résultat :

0 est pair

1 est impair

2 est pair

3 est impair

4 est pair

5 est impair

6 est pair

7 est impair

8 est pair

9 est impair

10 est pair

## Exemple 2:

for ($i=10; $i -ge 0; $i=$i-1)

{

$Parite = $i%2 -eq 0

if($i -eq 0)

{

Write-Host "On est arrivé à zéro"

}

elseif($Parite)

{

Write-Host "$i est pair"

}

else

{

Write-Host "$i est impair"

}

}

Résultat :

10 est pair

9 est impair

8 est pair

7 est impair

6 est pair

5 est impair

4 est pair

3 est impair

2 est pair

1 est impair

On est arrivé à zéro

## **6. Branchement switch**

A basic Switch statement has the following format:

Switch (<test-value>)

{

<condition> {<action>}

<condition> {<action>}

}

# selon la valeur de la variable $test

clear-host

switch ($test)

{

{$test -eq 1}

{write-host "la valeur est 1"}

{$test -eq 2}

{write-host "la valeur est 2"}

{$test -eq 3}

{write-host "la valeur est 3"}

}

## **7. Boucle foreach**

La boucle foreach est utilisée pour traiter les collections.

On peut affecter un ensemble d’items à une variable. C’est ce qu’on appelle une collection.

Exemples :

$Nombres = 2,4,6,8 (Ensemble d’entiers)

$Cara = "a","b","c","d","e" (Ensemble de caractères)

$Mots = "Émilie","Marie","Gabrielle" (Ensemble de caractères)

$Fichier = Get-Content C:\Temp\FichierPSH.txt (Ensemble d’enregistrements, (de lignes) du fichier)

La boucle foreach parcourt l’ensemble (la collection) et traite chacun de ses éléments. Elle gère le début et la fin de la collection.

Exemple 1 :

# Exemple avec une collection qui contient des entiers

Clear-Host

$Nombres = 4,7,8,2,3,6,5

$i = 0

foreach ($Nombre in $Nombres)

{

$i++

Write-host "Le nombre en position $i est $Nombre"

}

Résultat :

Le nombre en position 1 est 4

Le nombre en position 2 est 7

Le nombre en position 3 est 8

Le nombre en position 4 est 2

Le nombre en position 5 est 3

Le nombre en position 6 est 6

Le nombre en position 7 est 5

Exemple 2 :

Clear-Host

# Exemple avec une collection qui contient des caractères

$Mots = "Émilie", "Marie", "Gabrielle"

$i = 0

foreach ($Mot in $Mots)

{

$i++

Write-host "Le mot en position $i est $Mot"

}

Résultat :

Le mot en position 1 est Émilie

Le mot en position 2 est Marie

Le mot en position 3 est Gabrielle

Exemple 3 :

Clear-Host

# Exemple avec un fichier (collection de lignes)

$fichier = Get-Content C:\Temp\FichierPSH.txt

$i = 0

foreach ($ligne in $fichier)

{

$i++

Write-host "La ligne numéro $i du fichier est: $ligne"

}

Résultat :

La ligne numéro 1 du fichier est: Germinal Émile ZOla

La ligne numéro 2 du fichier est: Cent ans de solitude Gabriel Garcia Marquez

La ligne numéro 3 du fichier est: Les cerfs-volants Romain Gary

La ligne numéro 4 du fichier est: L'exil et le royaume Albert Camus

Notes :

1. pour pouvoir utiliser la commande ICACLS dans une fenêtre PSH, il faut encadrer chaque attribution de droits par des guillemets :

ex : ICACLS C:\Rep1 /grant "System:(oi)(ci)(F)" "Administrateurs:(oi)(ci)(F)"

1. Commentaire : # au début d’une ligne

<# et #> pour le début et la fin d’un paragraphe

1. Accent grave à la fin de la ligne pour changer de ligne
2. Clear-host au début pour ne pas avoir le code dans la fenêtre de résultats
3. Création de partages en PowerShell : New-SMBShare
4. Aide : get-help, get-help about, documentation, Internet

# PowerShell pour les scripts

