Introduction PowerShell

# But de l'exercice

Rafraîchir les connaissances acquises lors du module MA-18 « Mode Console »

# Questions

## Dans PowerShell ISE, lancer la *cmdlet* get-command afin d'obtenir toutes les commandes disponibles:

Get-command

Get-command....................................................................................................................

..............................................................................................................................................

## Lister toutes les *cmdlets* disponibles.

Get-command -commandType cmdlet

Get-command –Type Cmdlet ou Get-command -commandType cmdlet

Get-command –CommandType Cmdlet

..............................................................................................................................................

## Lister toutes les commandes qui s’appliquent aux objets.

Get-command –noun object ou Get-command –Name \*object\*

Get-command –Name \*object\*

..............................................................................................................................................

## Trouver une commande qui permette de lister les services Windows

Get-ServiceGet-service

..............................................................................................................................................

## Trouver au moyen de l'aide sur la *cmdlet*, comment afficher les exemples d'utilisation de cette commande.

Get-help get-service, Get-help get-service -examples

Get-help –examples c*mdlet*

.............................................................................................................................................

## Afficher tous les services qui commencent par "t".

Get-service t\* (aussi : get-service –name t\*)

Get-Service \*t ..............................................................................................................................................

## Afficher tous les services en exécution

Get-service | where-object {$\_.Status –eq ‘’Running ‘’}

get-service | where-object {$\_.Status -eq "Running"}

..............................................................................................................................................

## Afficher tous les services en exécution qui commencent par t\*

Get-service t\* | where-object {$\_.Status –eq ‘’Running ‘’}

get-service t\* | where-object {$\_.Status -eq "Running"}

## Rechercher une commande qui permette l'affichage du texte *"Hello world!"*

Write-host "Hello world "

Write-host "hello world"

Echo "hello world"

..............................................................................................................................................

## Afficher le même texte en rouge sur un fond gris foncé.

Write-host –BackgroundColor DarkGray –ForegroundColor Red Hello World !

write-host -ForegroundColor red -BackgroundColor gray hello

..............................................................................................................................................

## Afficher les jours de la semaine avec comme séparateur "|".

Write-host (‘’lundi’’, ‘’Mardi’’, ‘’Mercredi’’, ‘’Jeudi’’, ‘’Vendredi’’, ‘’Samedi’’, ‘’Dimanche’’) –Separator ‘’|’’

write-host ("lu","ma","me","je","ve","sa") -Separator "| "

..............................................................................................................................................

## Stocker la chaîne *"Hello world !"* dans une variable $strHelloMessage.

$strHelloMessage = ‘’Hello World !’’

$strHelloMessage = "Hello world !". .............................................................................................................................................

## Afficher la variable $strHelloMessage.

Write-host $strHelloMessage

Write-host $strHelloMessage ..........................................................................................................................................

## Afficher le résultat du calcul: 143 \* 144

143\*144 OU write-host(143\*144) $intCalc = 143\*144, write-host $intCalc

143 \* 144

..........................................................................................................................................

## A l'aide de la *cmdlet* get-memberet d'un pipe afficher les membres de la variable $intCalc

$intCalc | get-member

$intCalc = 143 \* 144

$intCalc | get-member

..............................................................................................................................................

## Donner le type de la variable $intCalc.

$intCalc.GetType()

$intCalc.getType()

..............................................................................................................................................

## Donner le type de la variable $strHelloMessage.

$strHellopMessage.getType()

$strHelloMessage .GetType()

..............................................................................................................................................

## Que peut-on déduire suite aux réponses des deux dernières questions ?

PowerShell assigne automatiquement le type aux variables en fonction des paramètres

Les 2 variables sont de type différentes (int et string)

fourni par l’utilisateur

..............................................................................................................................................

## Convertir la variable $intCalc en TypeName: System.String. Puis vérifier la conversion.

$intCalc=$intCalc.toString() et $intCalc | get-member

$intCalc= $intCalc.ToString()

$intCalc = [String] $intCalc

..............................................................................................................................................

## Afficher la longueur de la chaîne convertie.

$intCalc.length

$strCalc.length......................................................................................................................

## Afficher le contenu du lecteur d:

Get-childitem d:\

Get-ChildItem D:

..............................................................................................................................................

## Afficher le contenu du lecteur d: de manière récursive (tous les répertoires et fichiers)

Get-childitem –recurse d:\

Get-ChildItem D: -recurse

Get-ChildItem D: -s

..............................................................................................................................................

## Utiliser la commande précédente et utiliser un affichage par table.

Get-childitem –recurse d:\ | format-table

Get-ChildItem D: -recurse | format-table

..............................................................................................................................................

## Modifier la commande précédente en utilisant un affichage en liste.

Get-childitem –recurse d:\ | format-list

Get-ChildItem D: -recurse | format-list

..............................................................................................................................................

## Modifier la commande précédente en utilisant un affichage en largeur.

Get-childitem –recurse d:\ | format-wide

Get-ChildItem D: -recurse | format-wide

..............................................................................................................................................