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

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

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

## Lister toutes les *cmdlets* disponibles.

Get-command –commandtype cmdlet

get-command -type cmdlet ............................................................................

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

Get-command –noun object

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

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

Get-service

Get-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 get-service -example...........................................................................................

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

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

Get-Service -Name 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 {$\_.Status –eq ‘’Running ‘’} Get-Service -Name "t\*" | Where-Object {$\_.Status -eq "Running"}

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

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

Write-host

Write-Host "Hello World !"

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

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

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

Write-Host "Hello World !" -ForegroundColor 4 -BackgroundColor 8 .....................

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

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

"Lundi | Mardi | Mercredi | Jeudi | Vendredi | Samedi | Dimanche"...............................

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

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

$strHelloMessage = ‘’Hello World !’’

$strHelloMessage = "Hello World!"..............................................................................

## Afficher la variable $strHelloMessage.

Write-host $strHelloMessage

Echo $strHelloMessage..................................................................

## Afficher le résultat du calcul: 143 \* 144 ; assigner le résultat à la variable $intCalc

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

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

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

$intCalc | get-member

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

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

## Donner le type de la variable $intCalc.

TypeName : System.Int32 (on peut aussi faire $intCalc.GetType() )

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

## Donner le type de la variable $strHelloMessage.

$strHelloMessage |get-member, TypeName : System.String

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

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

PowerShell assigne automatiquement le type aux variables en fonction des paramètres

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

fourni par l’utilisateur

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

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

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

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

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

## Afficher la longueur de la chaîne convertie.

$a.Length$intCalc.length

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

## Afficher le contenu du lecteur d:

Get-childitem d:\

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

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

Get-childitem –recurse d:\

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

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

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

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

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

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

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

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

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

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