

SISR4 05 Administration des systèmes

02 Aide Mémoire PowerShell

Liste des commandes	Get-commande	
Récupérer l'aide sur un cmdLet	Get-help get-service	
Récupérer toutes propriétés et méthodes	Get-service get-member	
Paramètres de sécurité	Get-executionPolicy set-executionPolicy remoteSigned set-executionPolicy unrestricted	
Exécuter un script depuis commande MS-DOS	Powershell.exe " c:\monScript.ps1 "	
Variable (doit commencer par \$) variable typée	\$maVariable = 13 [int] maVariable =13	
Tableaux - initialisation - requete	\$tableau=@(1,2,3,4) \$valeur = \$tableau[1]	
Les constantes	Set-variable -name maConstante -value 192.168.1.32 -option constant	
Ecrire dans la console	Write-host "Montexte : « +\$variable	
Saisir une entrée utilisateur	\$reponse = read-host 'Tapez votre réponse : ' <i>la valeur saisie est dans la variable \$reponse.</i>	
Commentaire	#Ceci est mon commentaire	
Plusieurs commandes sur une ligne	\$v1=198;\$v2='Mon Texte';\$v3=\$V1*3,14	
Une commande sur plusieurs lignes	Mettre le caractère '`' à la fin de chaque ligne intermédiaire.	
Envoyer la sortie vers une autre commande (pipeline ou tube)	Get-service get-member	
Utiliser le filtrage intégré	get_service where-object {\$_.Status -eq 'stopped'} on peut remplacer where-object par where ou par ?	
Trier les colonnes de sortie	get-process sortId get-process sort-objectId	
Variables prédéfinies	\$\$	Contient le dernier mot de la dernière commande frappée dans la console.
	\$?	Contient <i>true</i> si la dernière commande a réussi, <i>false</i> sinon.
	\$^	Contient le premier mot de la dernière commande frappée dans la console.
	\$_	Contient l'objet courant transmis par le pipe ().
	\$Args	Contient le tableau des arguments passés à un script.
	\$Error	Contient, sous forme de tableau, la liste des erreurs de la session. \$Error[0] contient la dernière erreur.
	\$False	Contient la valeur <i>False</i>

	<table> <tr> <td>\$Home</td><td>Contient le chemin d'accès au répertoire de l'utilisateur</td></tr> <tr> <td>\$Host</td><td>Contient des informations sur l'ordinateur</td></tr> <tr> <td>\$Null</td><td>Variable vide</td></tr> <tr> <td>\$Pwd</td><td>Répertoire en cours</td></tr> <tr> <td>\$True</td><td>Contient la valeur <i>true</i></td></tr> </table>	\$Home	Contient le chemin d'accès au répertoire de l'utilisateur	\$Host	Contient des informations sur l'ordinateur	\$Null	Variable vide	\$Pwd	Répertoire en cours	\$True	Contient la valeur <i>true</i>										
\$Home	Contient le chemin d'accès au répertoire de l'utilisateur																				
\$Host	Contient des informations sur l'ordinateur																				
\$Null	Variable vide																				
\$Pwd	Répertoire en cours																				
\$True	Contient la valeur <i>true</i>																				
Les opérateurs arithmétiques	<table> <tr> <td>+</td><td>Addition</td></tr> <tr> <td>-</td><td>Soustraction</td></tr> <tr> <td>*</td><td>multiplication</td></tr> <tr> <td>/</td><td>Division</td></tr> <tr> <td>%</td><td>Modulo : calcul du reste de la division</td></tr> </table>	+	Addition	-	Soustraction	*	multiplication	/	Division	%	Modulo : calcul du reste de la division										
+	Addition																				
-	Soustraction																				
*	multiplication																				
/	Division																				
%	Modulo : calcul du reste de la division																				
Les opérateurs de comparaison	<table> <tr> <td>-eq</td><td>Égal</td></tr> <tr> <td>-ne</td><td>Non égal</td></tr> <tr> <td>-gt</td><td>Plus grand strictement</td></tr> <tr> <td>-ge</td><td>Plus grand ou égal</td></tr> <tr> <td>-lt</td><td>Plus petit strictement</td></tr> <tr> <td>-le</td><td>Plus petit ou égal</td></tr> <tr> <td>-like</td><td>Comparaison d'égalité générique (le caractère générique est * et non pas % comme en SQL)</td></tr> <tr> <td>-notlike</td><td>Test l'inégalité générique.</td></tr> <tr> <td>-match</td><td>Équivalent à like, mais pas de caractère générique possible.</td></tr> <tr> <td>-nomatch</td><td></td></tr> </table>	-eq	Égal	-ne	Non égal	-gt	Plus grand strictement	-ge	Plus grand ou égal	-lt	Plus petit strictement	-le	Plus petit ou égal	-like	Comparaison d'égalité générique (le caractère générique est * et non pas % comme en SQL)	-notlike	Test l'inégalité générique.	-match	Équivalent à like, mais pas de caractère générique possible.	-nomatch	
-eq	Égal																				
-ne	Non égal																				
-gt	Plus grand strictement																				
-ge	Plus grand ou égal																				
-lt	Plus petit strictement																				
-le	Plus petit ou égal																				
-like	Comparaison d'égalité générique (le caractère générique est * et non pas % comme en SQL)																				
-notlike	Test l'inégalité générique.																				
-match	Équivalent à like, mais pas de caractère générique possible.																				
-nomatch																					
Les conditions	<pre>\$maVariable = 12 if (\$maVariable -eq 12) { write-host 'ma variable est = 12' } else if (\$maVariable -gt 12) { write-host 'ma Variable est plus grande' }</pre>																				
Les boucles DO...WHILE (répète une suite d'instruction jusqu'à ce qu'une condition soit remplie)	<pre>\$i=0 do { write-host 'ma Valeur = '+\$i \$i++ } (while (\$i -ge 20))</pre>																				
Les boucles FOR (répète une suite d'instructions un certain nombre de fois)	<pre>For (\$iv = 0; \$i lte 20;\$i++) { write-host 'ma valeur = '+\$i }</pre>																				
Les fonctions	<pre>Function surface { param (\$longueur,\$largeur) return \$longueur*\$largeur }</pre>																				
Lire à partir d'un fichier	<pre>\$mesLignes = get-content 'c:\mesvaleurs.txt'</pre>																				

	<pre>foreach (\$uneLigne in \$ mesLignes) { write-host \$uneLigne } \$mesLignes[x] contient la ligne n° X+1 du fichier.</pre> <p>Autre solution :</p> <pre>Get-Content '.\mesValeurs.txt foreach { \$ligne = \$_ Write-Host \$ligne }</pre>
Écrire un fichier Texte	<pre>\$mV = "TOTO" \$mV out-file monfichier.txt</pre>