Cheat sheet Powershell

Commands

Update-Help	Downloads and installs newest helpfiles
Get-Help	Displays information about commands and concepts
Get-Command	Gets all commands
Get-Member	Gets the properties and methods of objects
Get-Module	Gets the modules that have been imported or that can be imported

OperatorsAssignment Operators

=, +=, -=, *=, /=, %=, ++, --	Assigns one or more values to a variable

Comparison Operators

-eq	Equal
-ne	Not equal
-gt, -ge	Greater than, greater than or equal to
-lt, -le	Less than, less than or equal to
-replace	Changes the specified elements of a value
-match, -notmatch	Regular expression match
match-like, -notlike	Wildcard matching
-contains, -notcontains	Returns TRUE if the scalar value on its right is contained in the array on its left
-in, -notin	Returns TRUE only when test value exactly matches at least one of the reference values.

Handy

-Split	Splits a string
[]	Casts a object to a specified type
$env:psmodulepath -split”;”	Find the module path

Loops

$a=1 Do {$a; $a++} While ($a –lt 10)	Do While Loop
$a=1 Do {$a; $a++} Until ($a –gt 10)	Do Until Loop
For ($a=1; $a –le 10; $a++) {$a}	For Loop
Foreach ($i in Get-Childitem c:\windows) {$i.name; $i.creationtime}	ForEach - Loop Through Collection of Objects

Statements

$a = "white" if ($a -eq "red") {"The colour is red"} elseif ($a -eq "white") {"The colour is white"} else {"Another colour"}	If statement
$a = "red"	SwitchStatement

switch ($a)
{
"red" {"The colour is red"}
"white"{"The colour is white"}
default{"Another colour"} }
}	

To file:

$a = Get-Content "c:\servers.txt"
foreach ($i in $a)
{$i}
	Reading From a File

$a = "Hello world" $a | out-file test.txt	Writing to a file
$a = Get-Process $a | Convertto-Html -property Name,Path,Company > test.htm	Writing to HTML
$a = Get-Process $a| Select-Object Name,Path,Company | Export-Csv -path test.csv	Writing to a CSV File

Powershell remoting

Enable-PSRemoting	Enable remoting (must be admin)
Invoke-Command –ComputerName MyServer1 -ScriptBlock {Hostname}	Execute a script on the remote session
Invoke-Command -ComputerName MyServer1 -FilePath C:\Invoke-Example.ps1	Run a locally stored script against a remote system
Enter-PsSession –ComputerName server1.domain.com	Establish an interactive powershell console session
Exit-PsSession	Exit the session