PowerShell Cheat Sheet

comparitech

Basic Commands

Cmdlet	Commands built into shell written in .NET
Functions	Commands written in PowerShell language
Parameter	Argument to a Cmdlet/Function/Script
Alias	Shortcut for a Cmdlet or Function
Scripts	Text files with .ps1 extension
Applications	Existing windows programs
Pipelines	Pass objects Get-process word | Stop-Process
Ctrl+c	Interrupt current command
Left/right	Navigate editing cursor
Ctrl+left/right	Navigate a word at a time
Home / End	End Move to start / end of line
Up/down	Move up and down through history
Insert	Toggles between insert/overwrite mode
F7	Command history in a window
Tab / Shift-Tab	Command line completion

Variables

$var = "string"	Assign variable
$a,$b = 0 or $a,$b = 'a','b'	Assign multiple variables
$a,$b = $b,$a	Flip variables
$var=[int]5	Strongly typed variable

Help

Get-Command	Get all commands
Get-Command -Module RGHS	Get all commands in RGHS module
Get-Command Get-p*	Get all commands starting with get-p
Get-help get-process	Get help for command
Get-Process | Get-Member	Get members of the object
Get-Process| format-list -properties *	Get-Process as list with all properties

Scripts

Set-ExecutionPolicy -ExecutionPolicy	Bypass Set execution policy to allow all scripts
."\\c-is-ts-91\c$\scripts\script.ps1"	Run Script.PS1 script in current scope
&"\\c-is-ts-91\c$\scripts\script.ps1"	Run Script.PS1 script in script scope
.\Script.ps1	Run Script.ps1 script in script scope
$profile	Your personal profile that runs at launch

Import, Export, Convert

Export-CliXML	Import-CliXML
ConvertTo-XML	ConvertTo-HTML
Export-CSV	Import-CSV
ConvertTo-CSV	ConvertFrom-CSV

Flow Control

```
If(){} Elseif(){ } Else{ }
while(){}
For($i=0; $i -lt 10; $i++){}
Foreach($file in dir C:\){$file.name}
1..10 | foreach{$_}
```

Comments, Escape Characters

#Comment	Comment
<#comment#>	Multiline Comment
"`"test`""	Escape char `
`t	Tab
`n	New line
`	Line continue

Parameters

-Confirm	Prompt whether to take action
-WhatIf	Displays what command would do

Assignment, Logical, Comparison

=, +=, -=, ++,--	Assign values to variable
-and, -or, -not,!	Connect expressions / statements
-eq, -ne	Equal, not equal
-gt, -ge	Greater than, greater than or equal
-lt, -le	Less than, less than or equal
-replace	"Hi" -replace "H", "P"
-match, -notmatch	Regular expression match
-like, -notlike	Wildcard matching
-contains, -notcontains	Check if value in array
-in, -notin	Reverse of contains, notcontains.

Common cmdlets

cd, chdir, sl	Set-Location
cat, gc, type	Get-Content
ac	Add-Content
sc	Set-Content
copy, cp, cpi	Copy-Item
del, erase, rd, ri, rm, rmdir	Remove-Item
mi, move, mv	Move-Item
si	Set-Item
ni	New-Item
sleep	Start-Sleep
sajb	Start-Job
compare, diff	Compare-Object
group	Group-Object
curl, iwr, wget	Invoke-WebRequest
measure	Measure-Object
nal	New-Alias
rvpa	Resolve-Path
rujb	Resume-Job
set, sv	Set-Variable
shcm	Show-Command
sort	Sort-Object
sasv	Start-Service
saps, start	Start-Process
sujb	Suspend-Job
wjb	Wait-Job
?, where	Where-Object
echo, write	Write-Output

Common Aliases

gcm	Get-Command
foreach,%	Foreach-Object
sort	Sort-Object
where, ?	Where-Object
diff, compare	Compare-Object
dir, ls, gci	Get-ChildItem
gi	Get-Item
copy, cp, cpi	Copy-Item
move, mv, mi	Move-Item
del, rm	Remove-Item
rni, ren	Rename-Item
fFt	Format-Table
fl	Format-List
gcim	Get-CimInstance
cat, gc, type	Get-Content
sc	Set-Content
h, history, ghy	Get-History
ihy, r	Invoke-History
gp	Get-ItemProperty
sp	Set-ItemProperty
pwd, gl	Get-Location
gm	Get-Member
sls	Select-String
cd, chdir, sl	Set-Location
cls, clear	Clear-Host

Arrays Objects

$arr = "a", "b"	Array of strings
$arr = @()	Empty array
$arr[5]	Sixth array element
$arr[-3..-1]	Last three array elements
$arr[1,4+6..9]	Elements at index 1,4, 6-9
$arr[1] += 200	Add to array item value
$z = $arA + $arB	Two arrays into single array
[pscustomobject]@{x=1;z=2}	Create custom object
(Get-Date).Date	Date property of object

Writing output and reading

"This displays a string"	String is written directly to output
Write-Host "color" -ForegroundColor Red -NoNewLine	String with colors, no new line at end
$age = Read-host "Please enter your age"	Set $age variable to input from user
$pwd = Read-host "Please enter your password" -asSecureString	Read in $pwd as secure string
Clear-Host	Clear console