



WorkshopPLUS - Windows PowerShell: Foundation Skills

Microsoft Services



Appendix

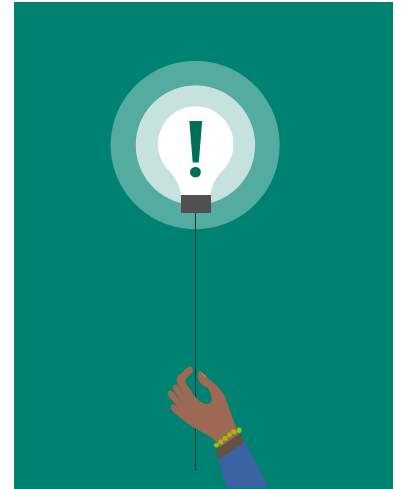


PowerShell Best Practices

Objectives

After completing PowerShell Best Practices, you will be able to:

- understand the best practises of PowerShell



Scripts

Scripts Guide

Write your code for someone else to read

Comment your code where warranted
(Regions)

Add help documentation to your script or function, with examples

Use native PowerShell, rather than equivalent external commands

One function implements one task
(Verb-Noun)

Expand all aliases

Let the ISE expand the Cmdlet (using <TAB>) for spelling and correct case

Use code-signing for production scripts

Invoke-expression can be used for code-injection, so use carefully

Declare all variables in loops before use

Use the Module concept to group related functions into a single suite

Pipeline

Pipeline Guide

Filter to the left,
format to the right

Accept input from
the pipeline and
send output to the
pipeline

Functions

Functions Guide

Use named parameters
(avoid positional
parameters)

Use Write-Verbose,
Write-Error and Write-
Debug cmdlets to
leverage Cmdlet
binding

Name your functions
using the existing
cmdlet naming
conventions.

Assign default values to
function parameters

Use Out-* and Write-*
cmdlets properly.
Write-host only emits
to the host application

Make use of switch
parameters to enact
different behaviours

Implement –WhatIf for
dangerous choices

