# Lab-4-1: Powershell Modules

# Objectives

* Gather a basic understanding of and experience with Microsoft PowerShell



# Preparation

1. Open the PowerShell ISE (To do that type ‘powershell’ in the search field on your workstation
2. Select Windows **PowerShell ISE**). You can also perform this lab in the basic PowerShell environment (Windows **PowerShell**), but you may be missing the code completion facilities of the Interactive Scripting Environment (ISE).
3. You can also pin it to your taskbar for ease of access in future.

# Exploring PowerShell

Solve the following tasks based on the **integrated help** (get-help), **lecture slides** and PowerShell **online reference**.

**Write down the commands you used to achieve the tasks.**

1. Find all installed modules in your computer

PS C:\Users\leggtc1> Get-module -ListAvailable | select Name

Name

----

Microsoft.PowerShell.Operation.Validation

PackageManagement

Pester

PowerShellGet

PSReadline

AppBackgroundTask

AppLocker

AppvClient

Appx

AssignedAccess

BitLocker

1. find all modules available on powershell gallery with name starting with “Azure”
2. PS C:\Users\leggtc1> find-module -Name "Azure\*" -Repository psgallery | Format-Wide -Property Name
3. Azure.Storage AzureRM.profile
4. Azure Azure.AnalysisServices
5. AzureRM.KeyVault AzureRM.OperationalInsights
6. AzureRM.Automation AzureRM.Compute
7. AzureRM.AnalysisServices AzureRM.Resources
8. AzureRM.ApiManagement AzureRM.Network
9. AzureTableEntity AzurePipelinesPS
10. AzureServicePrincipalAccount AzureDevOps
11. AzureIoT AzureArtifactsPowerShellModuleHelper
12. AzureExt AzureADPreview
13. AzureRM.SiteRecovery AzureRM.ServerManagement
14. AzureRmStorageTable AzureRM.AzureStackAdmin
15. AzureRM.AzureStackStorage azure.databricks.cicd.tools
16. AzureStack AzureHelpers
17. AzureRM.BootStrapper AzureAutomationAuthoringToolkit
18. azure.datafactory.tools AzureRmStorageQueue
19. Install terraform, import it and find all its commands

PS C:\Users\leggtc1> Install-Module "Terraform" -Repository psgallery

Import-Module Terraform

PS C:\Users\leggtc1> get-command -Module Terraform

CommandType Name Version Source

----------- ---- ------- ------

Function Connect-Terraform 1.0.5 Terraform

Function Get-Terraform 1.0.5 Terraform

Function Get-TFModule 1.0.5 Terraform

Function Get-TFOAuthToken 1.0.5 Terraform

Function Get-TFRun 1.0.5 Terraform

Function Get-TFState 1.0.5 Terraform

Function Get-TFWorkspace 1.0.5 Terraform

Function Get-TFWorkspaceVariable 1.0.5 Terraform

Function Lock-TFWorkspace 1.0.5 Terraform

Function New-TFWorkspace 1.0.5 Terraform

Function New-TFWorkspaceVariable 1.0.5 Terraform

Function Publish-TFModule 1.0.5 Terraform

Function Remove-TFModule 1.0.5 Terraform

Function Remove-TFWorkspace 1.0.5 Terraform

Function Remove-TFWorkspaceVariable ………………………………………

1. Write a function that will take month, year and day as input and print the corresponding day in the following format “Monday 10/08/2020”
2. PS C:\Users\leggtc1> function datestring {
3. param([string]$day, [string]$month, [string]$year)
4. $newDate = Get-Date -Date "$day-$month-$year" -UFormat "%A, %d/%m/%Y"
5. Write-Host "Date is: $newDate"
6. }
7. datestring -day "1" -month "8" -year "2021"
8. Date is: Sunday, 01/08/2021
9. Save the above function as module (hint: save as .psm1) and import it. Then call the function

PS C:\Users\leggtc1> datestring -day "1" -month "8" -year "2021"

Date is: Sunday, 01/08/2021

1. Write a module that includes a backup function which takes a source folder and destination folder as arguments and using the robocopy command copies everything from source to destination in backup mode. Write the logs on a file named backuplogs

function BackupDirectory{

param([string]$source, [string]$destination, [string]$file, [string]$logs)

robocopy $source $destination $file /b /mon:1 /mot:1 /tee /fp /e /log+:$logs

}

BackupDirectory -source "C:\Users\leggtc1\Documents\mydocs" -destination "C:\Users\leggtc1\Desktop\test" -file "\*.txt" -logs "C:\Users\leggtc1\Desktop\backuplogs.txt"

1. Create a folder named in609/files/ on c drive. Run the filemodifier (can be found on teams->assignment1).
2. Schedule a periodic (everyone minute) backup of the files directory you created above.

function BackupDirectory{

param([string]$source, [string]$destination, [string]$file, [string]$logs, [int]$timer, [int]$changes)

robocopy $source $destination $file /b /mon:$timer /tee /fp /e /v /im /mir /log+:$logs /xf "FileModifier.jar"

}

#BackupDirectory -source "C:\Users\leggtc1\Documents\mydocs" -destination "C:\Users\leggtc1\Desktop\test" -file "\*.txt" -logs "C:\Users\leggtc1\Desktop\backuplogs.txt"

BackupDirectory -source "C:\IN609" -destination "C:\Users\leggtc1\Desktop\filemodifier" -file "\*" -logs "C:\Users\leggtc1\Desktop\backuplogs2.txt" -timer 1