Advanced Scripting   
Intro to PowerShell Modules

Last Updated: 4/18/2024 8:59 AM Version 1  
Document Prepared for: CYBER360 Student

# Name Click here to enter name ID Click here to enter id

# Instructions

Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

Modules are just another tool to keep your reusable code well organized and easily accessed. This exercise bundles the functions of the previous exercise into a simple local module.

# Setup

## Requirements

* The script from the previous exercise, containing functions Get-IPNetwork, Test-IPNetwork
* VSCode
* PowerShell

# Task 1 — Create a directory for a module

Find your module path and ensure that your module folder exists.

## Steps

1. At your PowerShell prompt, get the paths from your **PSModulePath** environment variable:  
   **$psmp = $Env:PSModulePath -split ';'** # in Linux, change semicolon **;** to colon **:** 
   1. How many paths are in your PSModulePath? Click or tap here to enter text.
   2. Most of those paths are for system modules, but one of them is for your own modules and is in a subfolder of your own home folder, such as:  
      **/home/alice/.local** # Linux  
      **C:\Users\alice\Documents** or **C:\Users\alice\OneDrive\Documents** # Windows  
      What’s the index (offset) number in **$psmp** of yours? Click or tap here to enter text. (*Hint: if your own module folder is the ninth one in the array, the expression*  
       **($Env:PSModulePath -split ';')[8]**   
      *shows its directory path, because the 9th item is eight positions after the first*.)
2. Get the path to your own module folder:  
   **$mpath = $psmp[***<index recorded in step 1.2>***]**
3. Append **MyNetworkModule** to that path, and create that subfolder:  
   **$path = $mpath + '/MyNetworkModule/'**   
   **New-Item -Type Directory -Path $path**

# Task 2 — Create and import a module

Turn the **Network.ps1** script from the previous exercise into a module named **MyNetworkModule**.

1. Move the Network.ps1 script from the previous exercise to your new module folder, changing its name to **MyNetworkModule** and its extension to **.psm1**:  
   **Move-Item /tmp/Network.ps1 $path/MyNetworkModule.psm1** # Linux  
   **Move-Item C:\Temp\Network.ps1 $path\MyNetworkModule.psm1** # Windows  
   (If your script is not in the temporary folder but is somewhere else, modify as needed.)
2. Create a manifest for your new module:  
   **New-ModuleManifest -Path $path/MyNetworkModule.psd1 -RootModule MyNetworkModule.psm1**
3. Import your new module:  
   **Import-Module MyNetworkModule**
4. Verify that it was successfully imported:  
   **Get-Module -Name MyNetworkModule**
   1. Your output: Click or tap here to enter text.
5. Test the functions imported from your new module:  
   **Get-IPNetwork "172.16.173.1" "255.255.0.0"**   
   **Test-IPNetwork "172.16.173.1" "172.17.173.1" "255.255.0.0"**   
   **Get-Help Test-IPNetwork**   
   etc.

# Deliverable

Upload this document with completed answers to I-Learn Canvas.