# Week 4 PowerShell

Contents

[Week 4 PowerShell 1](#_Toc176177304)

[Tutorial 1: Functions 1](#_Toc176177305)

[Tutorial 2: Getting Started Scripts 2](#_Toc176177306)

[Tutorial 3: Get Objects Methods and Properties 3](#_Toc176177307)

[Tutorial 4: Manage Processes 4](#_Toc176177308)

[Tutorial 5: Using Windows PowerShell 4](#_Toc176177309)

[Tutorial 6: Cmdlets 5](#_Toc176177310)

[Assignment Submission 6](#_Toc176177311)

Time required: 90 minutes

**NOTE:** Perform this lab on Server1.

# Tutorial 1: Functions

A PowerShell function is a reusable block of code that performs a specific task or set of tasks. Functions are used to encapsulate a series of commands and can accept input parameters and return output. They help in organizing and simplifying scripts by breaking them into smaller, manageable pieces. In essence, a PowerShell function is like a mini-program within a PowerShell script, designed to perform a specific function or operation.

Create a file named **square\_number\_function.ps1**

A computer screen shot of a program code

Description automatically generated

Example run:



# Tutorial 2: Getting Started Scripts

Let’s run a few PowerShell commands to get started.

Create a file named **sample\_script.ps1**

A screenshot of a computer program

Description automatically generated

# Tutorial 3: Get Objects Methods and Properties

Create a PowerShell program named **get\_object\_methods\_properties.ps1**

A screen shot of a computer

Description automatically generated

# Tutorial 4: Manage Processes

Create a PowerShell Program named manage\_processes.ps1

A screen shot of a computer program

Description automatically generated

# Tutorial 5: Using Windows PowerShell

Time Required: Approximately 15 minutes

Objective: Use traditional Command Prompt commands and cmdlets in Windows PowerShell.

Description: Learning how to use Windows PowerShell can be an important way to expand your server administration toolbox. In this activity, you take several traditional commands and cmdlets for a test drive.

1. Click **Start** 🡪 Type **PowerShell** 🡪 Right Click **Windows PowerShell ISE** 🡪 Click **Run as Administrator**.
2. To view the files in the current folder 🡪 **dir** 🡪 Enter.
3. **Insert a screenshot of one screen of the listing.**

Click or tap here to enter text.

1. To change to the \Users directory, enter **cd c:\users** 🡪 press Enter.
2. Type **dir** 🡪 press **Enter**.
3. **Insert a screenshot.**

Click or tap here to enter text.

1. Type **Get-Command | more** and press Enter. You see the commands one screen at a time. Press the Spacebar to advance to the next screen. Repeat this step until you’ve viewed all the screens. (Note that you can also press q to exit back to the command line, if you decide not to view all the screens of commands.)
2. Press the up arrow and notice that the last command you entered is placed on the command line so that you can repeat the command. Press the up arrow again and you’ll see the second to-last command you entered. Press Enter to run that command.
3. Type **Get-Process** 🡪 press Enter to view the processes running on the server.
4. Start Google Chrome.
5. Type **Get-Process -name chrome** 🡪 press Enter. You will see several chrome processes.
6. **Insert a screenshot**.

Click or tap here to enter text.

# Tutorial 6: Cmdlets

In this Hands-On Project, you work with common Windows PowerShell administrative cmdlets.

1. On your Windows Server 2022 host, open the Google Chrome Web browser.
2. Right-click the Start menu and choose Windows PowerShell (Admin) to open Windows PowerShell.
3. At the prompt, type **Get-Process | more** and press enter. Note that there are many processes with a ProcessName of chrome that comprise the Google Chrome Web browser.
4. Insert a screenshot.

Click or tap here to enter text.

1. Press q to quit the more command.
2. Type **Stop-Process -name chrome** and press enter. Note that the Google Chrome Web browser app was closed.
3. Type **Get-Service | ogv** and press enter. Note that the App Readiness service is called AppReadiness and is not started by default. Close the GridView window.
4. Type **Start-Service -name AppReadiness** and press enter to start the service.
5. Type **Stop-Service -name AppReadiness** and press enter to stop the App Readiness service.
6. At the prompt, type **Test-NetConnection** and press enter to test your network connectivity to internetbeacon.msedge.net.
7. Insert a screenshot.

Click or tap here to enter text.

1. Execute the following commands at the command prompt, in turn. For each one, note the network configuration information displayed.
2. **Get-NetIPConfiguration**
3. Insert a screenshot.

Click or tap here to enter text.

1. **Get-NetAdapter**
2. Insert a screenshot.

Click or tap here to enter text.

## Assignment Submission

1. Attach the program files.
2. Attach this completed document.
3. Submit in Blackboard.