**PowerShell Exercises (Part I & Part II)**

**Exercise 1: File System Navigation and Manipulation**

**Objective:** Learn to navigate directories, list files, and create new directories using PowerShell.

**Instructions:**

1. **Open PowerShell:** Press the Windows key, type "powershell", and press Enter.
2. **Get current directory:** Type Get-Location and press Enter. This shows your current directory path in the file system.
3. **List files and folders:** Type Get-ChildItem and press Enter. This lists all files and folders in your current directory. You can use Get-ChildItem -Force to also show hidden items.
4. **Change directory:** Use the Set-Location cmdlet to navigate to a different directory. For example, to go to your "Documents" folder, type:

PowerShell

Set-Location "C:\Users\[YourUserName]\Documents"

(Replace [YourUserName] with your actual username.)

1. **Create a new directory:** Use the New-Item cmdlet to create a new folder. For example, to create a folder named "PowerShell\_Exercises" in your Documents folder, type:

PowerShell

New-Item -ItemType Directory -Path "C:\Users\[YourUserName]\Documents\PowerShell\_Exercises"

**Details:**

* **Get-Location:** This cmdlet displays the current working directory.
* **Get-ChildItem:** This cmdlet lists the files and folders in a specified location (or the current location if none is specified).
* **Set-Location:** This cmdlet changes the current working directory. You can provide a full path or a relative path (e.g., ..\ to go up one level).
* **New-Item:** This cmdlet creates new items in the file system, such as files, directories, or registry keys. The -ItemType parameter specifies the type of item to create.

**Exercise 2: Working with Services**

**Objective:** Learn to get information about services, start and stop services, and change service startup types.

**Instructions:**

1. **Open PowerShell:** Press the Windows key, type "powershell", and press Enter.
2. **List all services:** Type Get-Service and press Enter. This displays a list of all services installed on your computer, along with their status (Running, Stopped, etc.).
3. **Get a specific service:** To get information about a specific service, use the -Name parameter. For example, to get the status of the "Windows Audio" service, type:

PowerShell

Get-Service -Name "Audiosrv"

1. **Start a service:** To start a stopped service, use the Start-Service cmdlet. For example, to start the "Windows Audio" service, type:

PowerShell

Start-Service -Name "Audiosrv"

1. **Stop a service:** To stop a running service, use the Stop-Service cmdlet. For example, to stop the "Windows Audio" service, type:

PowerShell

Stop-Service -Name "Audiosrv"

1. **Change startup type:** To change how a service starts (e.g., Automatic, Manual, Disabled), use the Set-Service cmdlet. For example, to set the "Windows Audio" service to start automatically, type:

PowerShell

Set-Service -Name "Audiosrv" -StartupType 'Automatic'

**Details:**

* **Get-Service:** This cmdlet retrieves information about services.
* **Start-Service:** This cmdlet starts a service.
* **Stop-Service:** This cmdlet stops a service.
* **Set-Service:** This cmdlet modifies the configuration of a service, including its startup type.

**Exercise 3: Working with Variables and User Input**

**Objective:** Learn to create variables, assign values to them, and get user input.

**Instructions:**

1. **Open PowerShell:** Press the Windows key, type "powershell", and press Enter.
2. **Create a variable:** To create a variable, use the $ symbol followed by a name. For example, to create a variable named "username", type:

PowerShell

$username = "JohnDoe"

1. **Display the variable value:** To display the value of a variable, simply type its name and press Enter:

PowerShell

$username

1. **Get user input:** Use the Read-Host cmdlet to get input from the user. For example, to ask the user for their name and store it in the $username variable, type:

PowerShell

$username = Read-Host "Please enter your name"

1. **Use the variable in a command:** You can use variables in commands. For example, to display a personalized greeting, type:

PowerShell

Write-Host "Hello, $username!"

**Details:**

* **Variables:** Variables are used to store data in PowerShell. They are defined using the $ symbol.
* **Read-Host:** This cmdlet prompts the user for input and returns the entered value.
* **Write-Host:** This cmdlet displays information to the console.