

POWERSHELL

PowerShell can help you automate tasks, streamline workflows, and improve productivity.



25 MARCH 2024

08.00 PM - 09.00 PM

DURATION - 30 ~ 45 DAYS

REGISTER NOW



Scan the QR code and fill the form



More Information :
9391448443



SYLLABUS

BASICS

Module-1

- Windows PowerShell | Core PS
- What is Windows PowerShell
- Windows PowerShell Architecture
- Overview and Background
- Finding and Running Commands
- Exploring Inbuilt Security

Module-3

- Working with the Pipeline
- Concept of Pipeline
- Advantages of Pipeline
- Exporting, Importing, and Converting Data
- Filtering Objects Out of the Pipeline

Module-5

- Using PSProviders and PSDrives
- What are PSProviders and PSDrives
- Purpose of PSProviders and PSDrives
- Using PSDrives to work with
 - Registry, Files system
- Modifying/Querying the Registry and File system

Module-2

- PowerShell Help System
- Understanding PowerShell help System
- Types of Help Available
- How to read PS Help
- Understanding the commands
- Identifying the parameters

Module-4

- How the Pipeline Works
- Passing Data in the Pipeline ByValue
- Passing Data in the Pipeline ByPropertyName

Module-6

- Formatting Output
- Using Basic Formatting
- Exploring Inbuilt formatting
- Using Advanced Formatting
- Redirecting Formatted Output to file, Screen
- Custom Formatting

SYLLABUS

INTERMITTED

Module-7

- Using WMI and CIM
- Introduction to WMI and WMI Objects
- Need of WMI in Powershell
- Querying Data with WMI/CIM
- Making Changes with WMI/CIM

Module-8

- Preparing for Scripting
- Types Of Variable
- Using Variables
- Scope of Variables
- Data types
- Operators
- Scripting Security

Module-9

- Loops Supported in Powershell
- If Statement
- If-else Statement
- Else-if Statement
- Switch Statement
- Nested If Statement
- Foreach
- For
- While Loop
- Continue and Break Statement

SYLLABUS

ADVANCED

Module-10

- Introduction to Error Handling
- Handling Errors
- Try Catch Finally statement
- Trap Exception
- Add error handling to a function
- Script to know list of exception

Module-12

- Moving From a Command to a Script to a Module
- Moving from Command to Script
- Moving From Script to Function to Module
- Implementing Basic Error Handling
- Using Basic Scripting Constructs
- Further Exploration in Scripting

Module-14

- Powershell Modules and Manifests
- Adding additional functionality in shell
- Exploring new CMDLETS
- Dynamic/Manual Module loading
- Creating your own module

Module-11

- Using Advanced PowerShell Techniques and Profiles
- Understanding Powershell Function
- Purpose Of Functions
- Creating Powershell Functions
- Test completed functions
- Adding Help information in functions
- Using Parameters in script
- Adding advanced parameters.
- Creating Profile Scripts

Module-13

- Administering Remote Computers
- What is PowerShell Remoting?
- Prerequisites for Remoting
- Using Basic Remoting 1:1 and 1: Many
- Creating interactive Remote session
- Creating Persistent Session
- Using Remoting for Delegated Administration