



**ADVANCED AUTOMATED  
ADMINISTRATION WITH WINDOWS  
POWERSHELL EĞİTİMİ  
5 GÜN**



**Digital Vizyon**  
Akademi

[www.digitalvizyon.net](http://www.digitalvizyon.net)



## İçindekiler

Eğitim Hakkında.....	4
Neler Öğreneceksiniz? .....	4
Ön Koşullar .....	5
Kimler Katılmalı.....	5
Outline .....	5
Creating Advanced Functions .....	5
Lessons .....	5
Lab: Converting a Command into an Advanced Function .....	6
Lab: Creating a Script Module .....	6
Lab: Defining Parameter Attributes and Input Validation .....	6
Lab: Writing Functions that use Multiple Objects .....	6
Lab: Documenting Functions by using Content-Based Help.....	7
Lab: Supporting -Whatif and -Confirm .....	7
Lab: Writing Functions that Accept Pipeline Input.....	7
Lab: Producing Complex Function Output.....	7
Using Cmdlets and Microsoft .NET Framework in Windows PowerShell .....	8
Lessons .....	8
Lab: Using .NET Framework in Windows PowerShell.....	8
Writing Controller Scripts .....	8
Lessons .....	8
Lab: Writing Controller Scripts that Display a User Interface .....	8
Handling Script Errors Lessons .....	8
Lab: Handling Errors in a Script.....	9
Using XML Data Files .....	9
Lessons .....	9
Lab: Reading, Manipulating and Writing Data in XML.....	9
Managing Server Configurations by Using Desired State Configuration .....	9
Lessons .....	9
Lab: Creating and Deploying a DSC Configuration.....	9
Analyzing and Debugging Scripts.....	9
Lessons .....	9
Lab: Analyzing and Debugging and Existing Script .....	9



Understanding Windows PowerShell Workflow .....	10
Lessons .....	10

## Eđitim Hakkında

“Advanced Automated Administration with Windows PowerShell” eđitimi, PowerShell kullanarak daha karmařık yönetim senaryolarını ele alan ve otomasyonu daha da ileri düzeyde kullanmayı öğreten bir eđitimidir. Bu eđitim, PowerShell’un derinlemesine özelliklerini ve kullanımını kapsar ve bir önceki PowerShell eđitimine dayanır. Bu eđitimde, PowerShell script oluřturma, ileri düzey fonksiyonlar, modüller ve otomasyonu daha da geliřtirmek için araçlar gibi konular ele alınır. ” Windows PowerShell eđitimi, sistem yöneticileri, ađ yöneticileri, yazılım geliřtiricileri ve diđer IT profesyonelleri için tasarlanmıřtır.

“Windows PowerShell İleri Düzey Eđitimi”, bir önceki PowerShell eđitiminin üzerine kurulu bir programdır ve katılımcılara PowerShell’in daha da ileri düzey özelliklerini ve kullanımını öğretmektedir. Bu nedenle, eđitimde, PowerShell script oluřturma, ileri düzey fonksiyonlar, modüller ve daha fazlası gibi konulara yoğunlařılmaktadır.

Bu eđitim, katılımcılara PowerShell komut dosyaları oluřturma ve bu betiklerin karmařık otomasyon görevlerini nasıl gerçekleřtirebileceđi konusunda daha detaylı bilgi sađlar. İleri düzey fonksiyonlar ve modüllerin kullanımı, PowerShell’in otomasyon kabiliyetlerini daha da geniřletmek için vazgeçilmez araçlardır.

“Windows PowerShell İleri Düzey Eđitimi”, özellikle sistem yöneticileri, ađ yöneticileri, yazılım geliřtiricileri ve diđer IT profesyonelleri için tasarlanmıřtır. Bu kiřiler, otomasyon ve betik oluřturma yeteneklerini geliřtirerek, iř yüklerini yönetme, süreçleri otomatikleřtirme ve genel iř verimliliđini artırma konusunda daha fazla kontrol kazanabilirler.

“Advanced Automated Administration with Windows PowerShell” eđitimi, PowerShell’u kullanarak büyük ölçekli ađ yönetim görevlerini otomatikleřtirme ve yapılandırma yönetimini daha etkili hale getirme becerilerini öğretir. Bununla birlikte eđitim, PowerShell ile ileri düzey senaryoları ele alarak, öğrencilerin PowerShell’u daha iyi anlamalarını, PowerShell scriptleri yazmayı öğrenmelerini ve PowerShell modülleri oluřturup yönetmelerini sađlar. Ayrıca, bu eđitim, PowerShell komut dosyalarının yanı sıra, PowerShell Advanced Functions, Cmdletbinding, Parameter validation ve Error handling konularını da kapsar. Bu eđitim, PowerShell kullanarak otomasyonu daha etkili hale getirmek ve yönetim görevlerini daha hızlı ve daha kolay bir řekilde gerçekleřtirmek isteyen her IT profesyoneli için faydalıdır.

## Neler Öğreneceksiniz?

“Advanced Automated Administration with Windows PowerShell” eđitiminde, ařađdaki konular dahil olmak üzere PowerShell ile ilgili daha geliřmiř konuları öğrenebilirsiniz:

- PowerShell script yazma ve geliřtirme
- PowerShell Advanced Functions oluřturma
- Cmdletbinding kullanarak parametre yönetimi



- PowerShell modülleri oluşturma ve yönetme
- Script debugging ve Error handling konuları
- PowerShell Desired State Configuration (DSC)
- PowerShell Workflow ve Jobs yönetimi
- PowerShell ile Azure ve Office 365 gibi bulut hizmetleri yönetimi

Bu eğitim, PowerShell kullanarak daha karmaşık otomasyon görevleri için gerekli olan becerileri öğretir ve PowerShell'un ileri düzey özelliklerini kullanarak yapılandırma yönetimini daha etkili hale getirmenizi sağlar.

## Ön Koşullar

“Advanced Automated Administration with Windows PowerShell” eğitiminde, katılımcıların “Automating Administration with Windows PowerShell” veya PowerShell temelleri hakkında bir ön bilgiye sahip olmaları önerilir. Ayrıca, PowerShell ile ileri düzey senaryoları ele almak için, PowerShell script yazma ve PowerShell modülleri oluşturma konularında da biraz deneyimli olmak faydalı olabilir. Bununla birlikte, bu konulara tamamen yabancı olanlar da bu eğitimden faydalanabilirler, ancak konuları daha iyi anlamak için ek öğrenme kaynaklarına başvurmaları gerekebilir.

## Kimler Katılmalı

“Advanced Automated Administration with Windows PowerShell” eğitimi, PowerShell kullanarak daha karmaşık yönetim senaryolarını ele almak isteyen sistem yöneticileri, ağ yöneticileri, yazılım geliştiricileri ve diğer IT profesyonelleri için tasarlanmıştır. Bu eğitim, PowerShell'u daha gelişmiş ve ileri düzeyde kullanmak isteyen herkes için faydalıdır. Ayrıca, büyük ölçekli yapılandırma yönetimi ve otomasyonu gerektiren işlerle uğraşanlar, özellikle bu eğitimden faydalanabilirler.

## Outline

### Creating Advanced Functions

#### Lessons

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm



### **Lab: Converting a Command into an Advanced Function**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

### **Lab: Creating a Script Module**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

### **Lab: Defining Parameter Attributes and Input Validation**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

### **Lab: Writing Functions that use Multiple Objects**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm



### **Lab: Documenting Functions by using Content-Based Help**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

### **Lab: Supporting -Whatif and -Confirm**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

### **Lab: Writing Functions that Accept Pipeline Input**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm

### **Lab: Producing Complex Function Output**

- Converting a Command into an Advanced Function
- Creating a Script Module
- Defining Parameter Attributes and Input Validation
- Writing Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Producing Complex Function Output
- Documenting Functions by using Content-Based Help
- Supporting -Whatif and -Confirm



- After completing this module, students will be able to:
- Convert a Command into an Advanced Function
- Create a Script Module
- Define Parameter Attributes and Input Validation
- Write Functions that use Multiple Objects
- Writing Functions that Accept Pipeline Input
- Produce Complex Function Output
- Document Functions by using Content-Based Help
- Support -Whatif and -Confirm

## Using Cmdlets and Microsoft .NET Framework in Windows PowerShell

### Lessons

- Running Windows PowerShell Commands
- Using Microsoft .NET Framework in Windows PowerShell

### Lab: Using .NET Framework in Windows PowerShell

- Using Static and Instance Members
- After completing this module, students will be able to:
- Run Windows PowerShell commands
- Use Microsoft .NET Framework in Windows PowerShell

## Writing Controller Scripts

### Lessons

- Understanding Controller Scripts
- Writing Controller Scripts that Show a User Interface

### Lab: Writing Controller Scripts that Display a User Interface

- Write Functions to be Used in the Controller Script
- Write a Controller Script that Implements a Text-Based Menu
- After completing this module, students will be able to:
- Understand Controller Scripts
- Write Controller Scripts that show a User Interface

## Handling Script Errors Lessons

- Understanding Error Handling
- Handling Errors in a Script





## **Lab: Handling Errors in a Script**

- Handling Errors in a Script
- After completing this module, students will be able to:
- Understand Error Handling
- Handle Errors in a Script

## Using XML Data Files

### **Lessons**

- Reading, Manipulating and Writing Data in XML

## **Lab: Reading, Manipulating and Writing Data in XML**

- Test the Provided Tools
- Update an XML Inventory Document
- After completing this module, students will be able to:
- Reading, Manipulating and Writing Data in XML

## Managing Server Configurations by Using Desired State Configuration

### **Lessons**

- Understanding Desired State Configuration
- Creating and Deploying a DSC Configuration

## **Lab: Creating and Deploying a DSC Configuration**

- Write, Run and Push a DSC Configuration
- After completing this module, students will be able to:
- Understand Desired State Configuration
- Create and Deploy a DSC Configuration

## Analyzing and Debugging Scripts

### **Lessons**

- Debugging in Windows PowerShell
- Analyzing and Debugging and Existing Script

## **Lab: Analyzing and Debugging and Existing Script**



- Analyzing and Debugging and Existing Script
- After completing this module, students will be able to:
- Analyze and Debug a script

## Understanding Windows PowerShell Workflow

### Lessons

- Understanding Windows PowerShell Workflow
- After completing this module, students will be able to:
- Understanding Windows PowerShell Workflow