Title: Beginning PowerShell Workshop

 Abstract:

In this 3 hour workshop, you’ll learn how PowerShell works and how to make PowerShell work for you—and you’ll learn it from the experts! Join [Jeffrey Snover](http://twitter.com/jsnover), the inventor of PowerShell, together with [Jason Helmick](http://twitter.com/theJasonHelmick), for this intense, script-filled experience. IT pros, admins, and help desk persons will learn how to improve your management capabilities, automate redundant tasks and manage your environment in scale.

# Introduction

## Introduction

## Why PowerShell Matters

## Why DevOps Matters

## DevOps Report and numbers

# Getting Started

## Launching PowerShell for the administrator

### Caution – 32-bit versus 64-bit

### Launch an Administrative console – Run As Administrator

## Customize the Shell for comfort

### Font selection and size

### Width and Height requirements

### Color options

## Getting familiar with the shell

### Native commands

### Cmdlet – Verb-Noun

### Aliases – shortcuts

### Scripts

# The Help System

## Updatable Help

## Get-Help, Help, Man

## Discoverability with the Help system

### The Process – Discover then Dig

### Examples:

#### Get-Help \*<noun>\*

#### Get-Help \*<verb>\*

#### Get-Help cmdlet –detailed

#### Get-Help cmdlet –Examples

#### Get-Help cmdlet –Full

#### Get-Help cmdlet –Online

#### Get-Help cmdlet –ShowWindow

### The other help system About\_

## Understanding Syntax

### Parameter Sets

### What does all this syntax mean?

## Real-world

### Examples of discovering and solving

#### Get-Service –Name bits

#### Stop-Service

#### Start-service

#### Get-AdComputer

# The Pipeline: Connecting commands

## What’s the pipe and what does it do?

### How things are passed

### Examples

#### Get-service –name bits

#### Get-Service –Name bits | Stop-Service

#### Get-Service | Stop-Service

## Exporting/Importing CSV’s

## Exporting/Importing XML

### Example: Compare-Object on Processes

## Printers and files

## Displaying information in a GUI

### Out-Gridview

## Making a webpage of Information

## Cmdlets that kill

### –Whatif

### –Confirm

# Extending the shell

## Finding and Adding Modules

### Version 2 – Import-Module

### Version 3 - Dynamic importing

## Discovering new commands

### Get-Command –Module <Module>

### Get-Command –Module <module> | Measure-Object

### Get-Help \*<noun>\*

# Objects for the Admin

## Object across the pipeline

### How they flow

### Get-Member

## Getting the information you need

### Type Name

### Methods

### Properties

## Sorting Objects

### Examples

#### Get-ChildItem | Get-Member #Property list

#### Get-ChildItem –Path c:\ | Sort-Object –property Length –Descending

#### Get-Process | Sort-Object –Property cpu –Descending

## Selecting Objects

### Examples

#### Get-Service | Select-Object –Property Name, Status

#### Get-Process | Select-Object –Property Name, ID, VM, PM

#### Trick: Get-Command –Module <module> | Sort-Object –property noun | Get-Help | Select-Object –Property Name, Synopsis

## Custom Properties

### Examples

#### Get-Service | Select-Object –Property Name, status , @{n=’MyState’;e={$\_.Status}}

#### Get-WmiObject –Class Win32\_LogicalDisk –Filter “DeviceID=’c:’” | Select-Object –Property Freespace

#### Get-WmiObject –Class Win32\_LogicalDisk –Filter “DeviceID=’c:’” | Select-Object –Property @{n=’FreeGB’;e={$\_.Freespace /1GB as [int]}}

## Filtering data

### Comparison operators

#### Examples

### Where-Object

#### Examples

### –Filter

### Filter left!

## Methods – when no cmdlet exists

### Brief introduction into Methods

### What if Stop-Service and Start-Service didn’t exist?

### Examples

#### Get-Service –Name bits | Get-Member #Use GM

#### Get-Service –Name bits | Foreach{$\_.stop()}

#### $var=Get-Service bits #Vars in more detail later

#### $var.Start()

# The pipeline: Deeper

## How the pipeline really works

## 4 Step solution

## 1- ByValue

### Example – Get-Service | Stop-Service

## 2 – ByPropertyName

### Example – Get-Service | Stop-Process

## 3- What if my property doesn’t match

### Custom property

### Example:

#### Get-Adcomputer –Filter \* | Select –ExpandProperty | Get-Service

## 4- Parenthetical – when all else fails

### Example – Get-WmiObject –ComputerName ()

# The Power in PowerShell – Remoting

## Overview of Remoting

## Enabling Remoting

### Enable-PSRemote and demonstration

### Group Policy

## One-To-One

### Enter-PSSession

## One-To-Many

### Invoke-Command

## Not the end yet!

### We still have the cool part – Sessions – but first let’s start to automate.

# Automation in Scale – Remoting

## Reusable sessions

### New-PSSession

## Using sessions with Invoke-Command

## Real-World Web server deployment

### Example:

#### Deploy web-server to multiple computers

#### Deploy a website

## Creating automation scripts

## Getting command from anywhere – Implicit Remoting

## Introduction to JEA

### Jason – have an endpoint already configured

### Show concept then demo

# Introducing scripting and toolmaking

## Execution Policy

## Variables: A place to store stuff

## Making commands repeatable

### Example – Get-WmiObject –class Win32\_LogicalDisk

### Example – GWMI –class win32\_Bios

## Adding Parameters to your script

## Documenting your script

## Turning your script into a function

### Example

#### Adding the properties to a hashtable

#### Creating your own object

## How about a Module?

### A place to store your commands

# Introducing DSC

## Architecture/ How it works

## Configurations

## Resources

## Mofs

## PSGallery Resources

## Quick Demo

## PSAutoLab