

# Welcome to PowerShell Basics

By - Radion Barakov

# Radion Barakov - Background



# Objectives & Expectations

- Familiarize with PowerShell & its flows
- Fit to read and write simple queries
- Develop interest for further use

# Topics

- History
- Core logics
- Language Basics
- Best Practice
- Conditions & Loops
- Summary

# History

- PS1.0 - 2006
- PS2.0 - 2009 – Most seen, part of Win7
- PS3.0 - 2012 – ISE & Tab Completion
- PS4.0 - 2013 – DSC
- PS5.0 - 2014 – Package Management
- PS6.0 - 2016 – Core, Cross Platform
- PS7.0 - 2019 – TBD - <https://bit.ly/2JEI3Wo>

# Core Logics

- Verb-Noun
- Verbs are categorised and nouns are descriptive.

I.e.: Add-Computer

Install-Module

Connect-AzureAD

Block-FileShareAccess

Test-Connection

# Language Basics

## **How to start:**

---

- Get-Help – About anything
- Get-Command – Command Specific
- Get-Member – For Objects

# Language Basics

---

- Command line
- Parameters – Explicit, Named, Switch

The diagram illustrates the components of a PowerShell command line. A horizontal line separates the command from its parameters. Above the line, the word "Parameters" is centered, with a bracket spanning from the start of the first parameter to the end of the last. Below the line, the word "Values" is centered, with a bracket spanning from the start of the first parameter to the end of the last. The command line itself is split into two sections by a vertical line. The left section is labeled "Command Line" and contains the command "Get-ChildItem". The right section is labeled "Parameters" and contains the parameters "-Path 'C:\Users\PcMasterRace\Documents' -Exclude \*.txt -Force". The parameters are further categorized: "-Path" is labeled "Explicit", "-Exclude" is labeled "Named", and "-Force" is labeled "Switch".

```
Command Line | Parameters
```

```
C:\> Get-ChildItem -Path 'C:\Users\PcMasterRace\Documents' -Exclude *.txt -Force
```

Parameters

Explicit Named Switch

Values



# Language Basics

## **Cmdlets:**

---

- PowerShell built-in (Core)
- Easily found with the help commands
- Command description & Examples

# Language Basics

## **Modules:**

---

- Integrated with existing Cmdlets
- Do not always follow the standards
- Output modification

# Language Basics

## **Functions:**

---

- Cmdlet like
- Customisable Parameters
- Can contain personalised Variables etc.

# Language Basics

## Objects:

- Output of a Cmdlet
- Structured Data
- Contains properties
- Can be manually created
- Easily convertible & sorted

# Language Basics

## **Properties & Methods:**

- Object Attributes (Characteristics)
- Property – Information
- Method – Action

# Language Basics

## **Variables:**

---

- Storing Objects & Values
- Fine-graining
- Types: Manual – Auto – Preference

# Language Basics

## **Aliases:**

---

- Customisable
- Understandable
- Built-in Aliases

# Language Basics

## **Pipelines:**

---

- Output Transfer
- Object Transfer
- Output Modification



# Language Basics

## **Arrays:**

- Collection of items
- Contains all types
- Expandable

# Language Basics

## **Hash Tables:**

- Key = Value
- Value can be anything
- Not ordered (Unless [Ordered])
- Convertible to Custom Objects

# Conditions & Loops

## **ForEach:**

---

- Variable assignment
- No Pipeline support
- Refinable (If,Else,Elseif - conditions)

# Conditions & Loops

## **ForEach-Object:**

- Different than ForEach
- ScriptBlock
- Preferred with Arrays, (I/O pipeline)

# Conditions & Loops

## **Until. While - Do:**

- Time conditions
- Script pausing
- Used when waiting for something to start

# Best Practice

## **Hands-on:**

---

- PowerShell Profile
- Modules
- Repositories

# Best Practice

- Nalias – NO ALIAS!
- Filter | Format
- `$DefaultParameters = $DEF:Parameters`
- `#Comment`  
`#[1 minute writing] > [3 minutes figuring] x [X amount]`
- Remember-Rule (Verb-Noun)
- Readable?

# Summary

## **Key points:**

- PowerShell is Object Oriented
- Starting PowerShell 6 it works on all Platforms
- Beneficial for lazy people



Summary

Q&A

---