Level Up with PowerShell

Using PowerShell to stop global warming and increase your raw animal magnetism

Brent Stewart

@BrentEStewart

about.me/BrentStewart

Co-Founder
Alien Arc Technologies, LLC
A Modern Apps Company

What is PowerShell?

A. A task automation tool
B. A command line interpreter
C. An energy drink made from snails
D. A configuration management framework
E. A, B, and D

Where do I start?

- PowerShell console
- PowerShell ISE (Integrated Scripting Environment)
- Set the Execution Policy

demo

Some Cmdlets to remember

Get-Help – Get help for a specific command

Get-Command – Get a list of available commands

Get-Member – Get the properties of an object

Measure-Object – Gets a count of a collection

Out-GridView – Output to a graphical window

Format-Table – Format the output as a table

Format-List – Format the output as a key/value list

Set-Alias – Create your own aliases

demo

PowerShell Basics

- Variables are designated with a dollar sign $variable
- Arrays are declared using @(1,2,3) syntax
- Hash tables are declared using @{a=1; b=2; c=3} syntax
- You can filter collections with Where-Object (where or ?)
- Comparisons use -eq, -gt, -lt, -ne, -like, etc…
- You can modify what is selected with Select-Object (select)
- Sort output with Sort-Object (sort)
- Pipe the output of one command to another
- Setup common scripts using your PowerShell profile

demo

Execution control

- if else statement
- for loop
- foreach (%) loop
  - $_ is the reference to the current item
- do while loop
- while loop
- do until loop

demo

Functions

- Denoted with the keyword function
- Can have 0..N parameters
- Parameters are passed in without parentheses and commas
- The Function: PSDrive lets you view the existing functions

demo

Tips, Tricks & Gotchas

- You can use single or double quotes, but variables are only expanded when using double quotes
- The back tick (‘) is the escape character
- Use ( ) to force evaluation of an expression
- Use – -WhatIf to see what would happen if you executed something

demo

Error handling

```
try { Code to execute… }
catch [ExceptionType] { Handle the error… }
finally { This code always runs… }
```

demo

Using .NET components

- Call static methods

```
$results = [System.Math]::Sqrt(9)
```

- Create a new object

```
$stack = New-Object System.Collections.Stack($null)
```

- Create a new type

```
Add-Type –TypeDefinition  @“{C# Code }”@
```

demo

Demo Time

Learn More

- Websites
  - PowerShell.org
  - blogs.technet.com/b/heyscriptingguy
  - blogs.msdn.com/b/powershell
  - pscx.codeplex.com
- Books
  - Windows PowerShell for Developers by Douglas Finke

Thank you

Brent Stewart
Alien Arc Technologies, LLC
@BrentEStewart

About.me/brentstewart