Level Up with PowerShell

Brent Stewart
@BrentEStewart
about.me/BrentStewart

- Co-Founder Alien Arc Technologies, LLC
- Co-Organizer of the Kansas City .NET User Group
- Been using command shells since Tandy DOS 3.3

What is PowerShell?

But I’m a developer.

Why should I learn PowerShell?

Developers don’t just develop

Don’t Repeat Yourself (DRY)

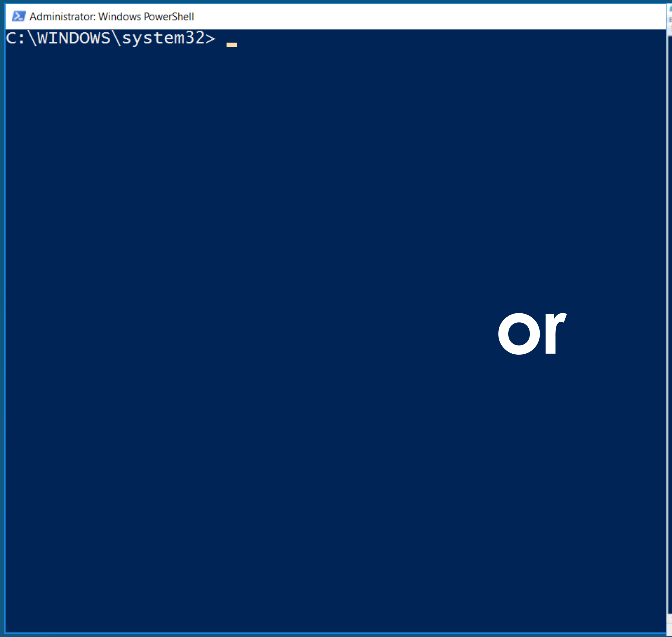
Don’t Repeat Yourself (DRY)

Leverages existing .NET skills

Sometimes quick and dirty is good enough

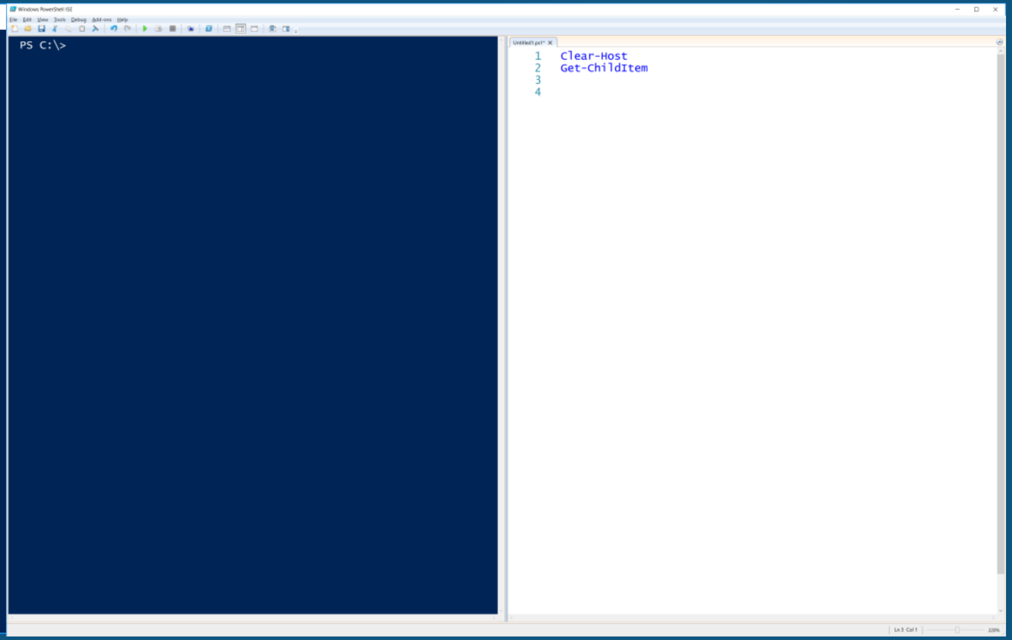
What are somethings that I can do with PowerShell?

Where do I start?



```
C:\WINDOWS\system32> _
```

or



```
PS C:\>
```

```
Clear-Host
Get-ChildItem
```

PowerShell Console

PowerShell ISE

(Integrated Scripting Environment)

demo

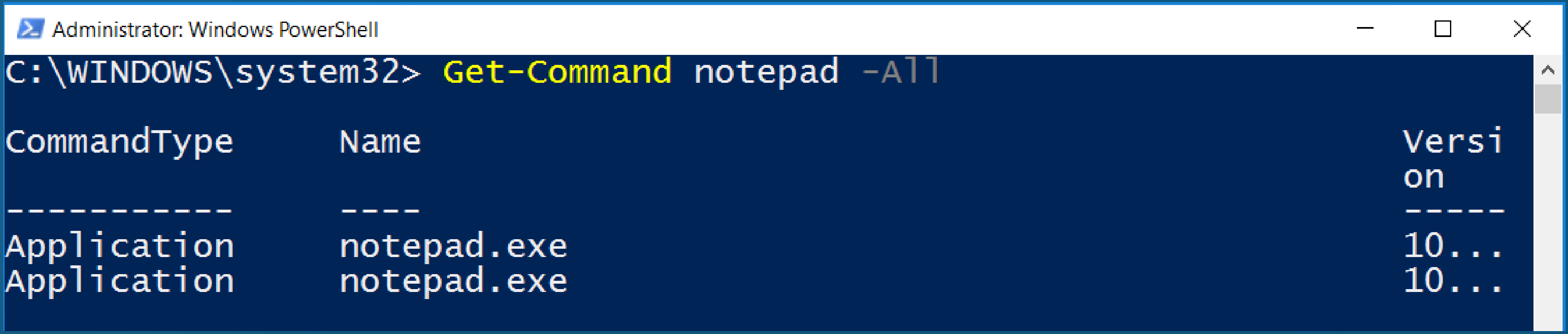
Set the Execution Policy

demo

Get-Command

Gets a list of all commands.

Notable parameters:
-all (shows every instance)
-commandtype (filters for specific types)



```
C:\WINDOWS\system32> Get-Command notepad -All

CommandType     Name                                                    Versi
                                                                        on
-----------     ----                                                    -----
Application     notepad.exe                                             10...
Application     notepad.exe                                             10...
```

Get-Help

Displays information about Windows PowerShell commands and concepts.

Notable parameters:

-examples (show examples)

-online (displays html help)

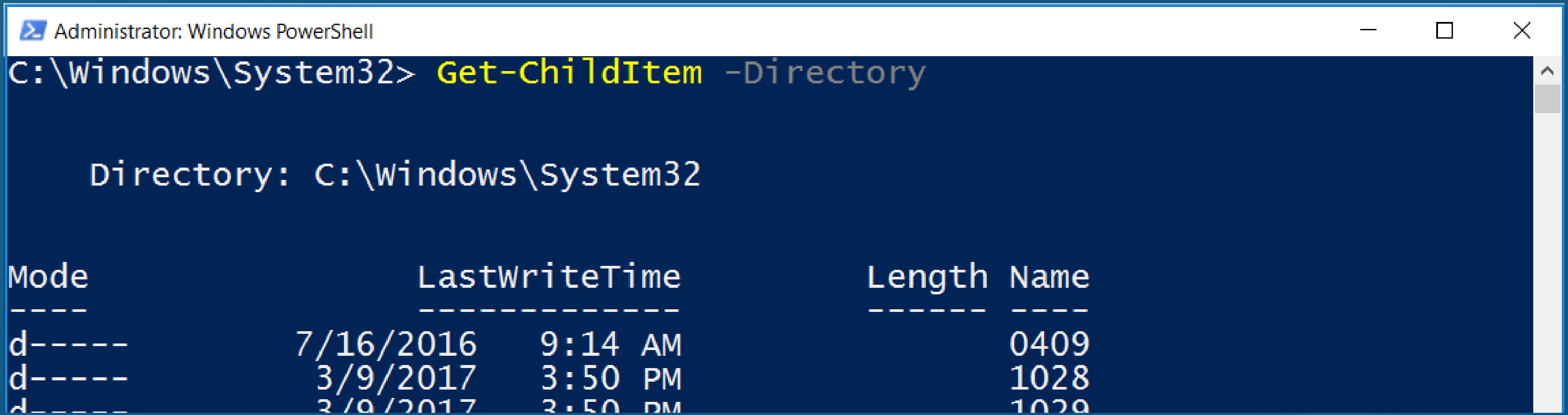
Get-ChildItem

Gets the directories and files.

Notable parameters:

-Directory (only gets directories)

-File (only gets files)



```
C:\Windows\System32> Get-ChildItem -Directory


    Directory: C:\Windows\System32


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        7/16/2016   9:14 AM                0409
d-----         3/9/2017   3:50 PM                1028
d-----         3/9/2017   3:50 PM                1029
```

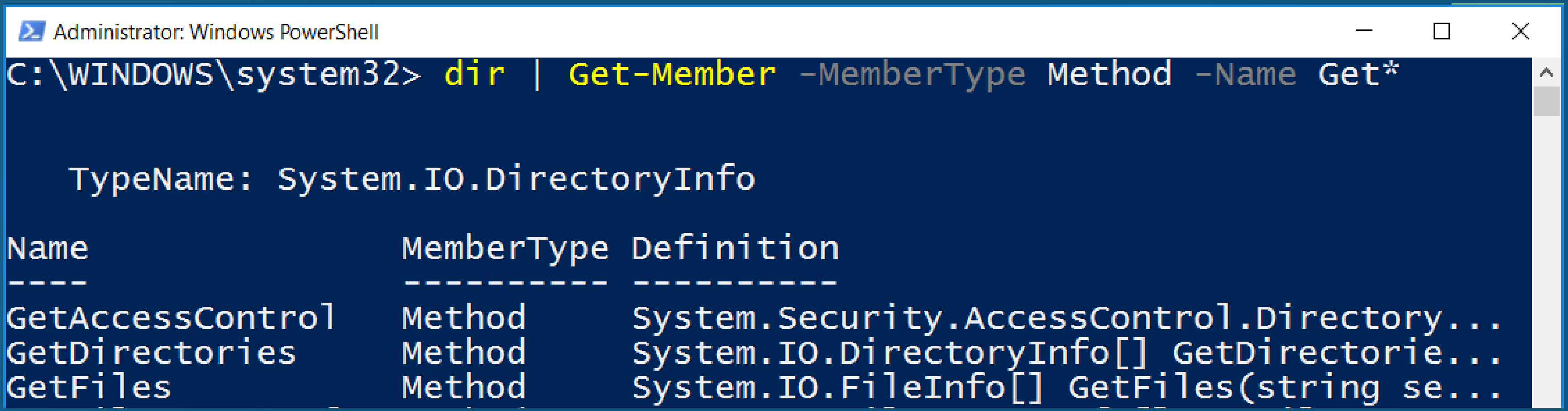
Get-Member

Gets the members, properties, and methods of objects.

Notable parameters:

-MemberType (filters by member type)

-Name (filters by name)



Administrator: Windows PowerShell

```
C:\WINDOWS\system32> dir | Get-Member -MemberType Method -Name Get*


   TypeName: System.IO.DirectoryInfo

Name                MemberType Definition
----                ---------- ----------
GetAccessControl    Method     System.Security.AccessControl.Directory...
GetDirectories      Method     System.IO.DirectoryInfo[] GetDirectorie...
GetFiles            Method     System.IO.FileInfo[] GetFiles(string se...
```

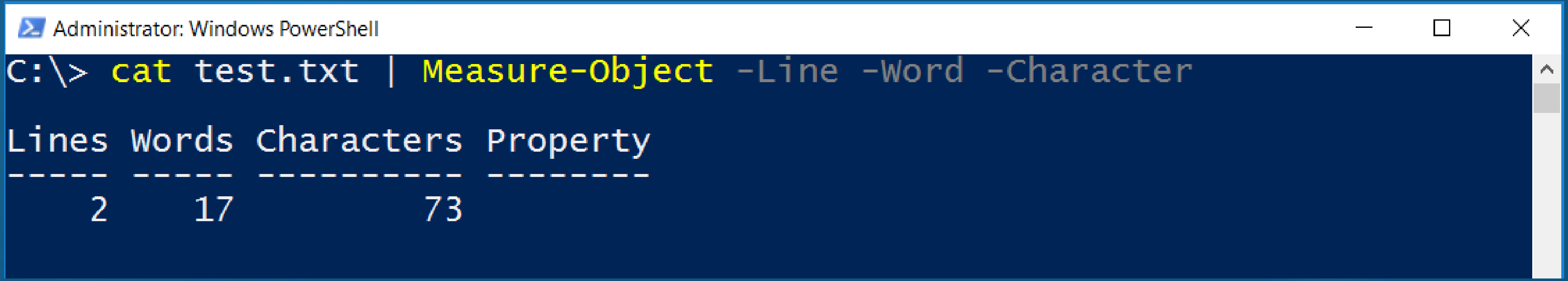
Measure-Object

Calculates the numeric properties of objects, and the characters, words, and lines in string objects, such as files of text.

Notable parameters:

-Line, -Word, -Character

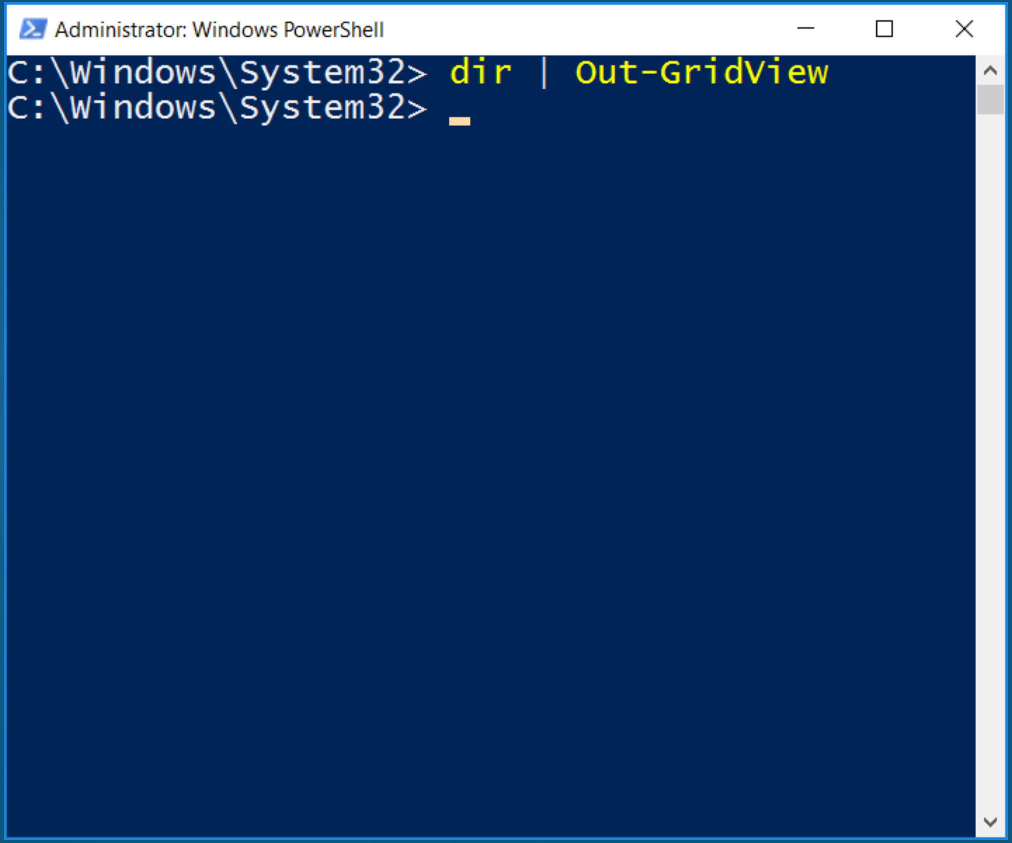
-Min, -Max, -Average, -Sum



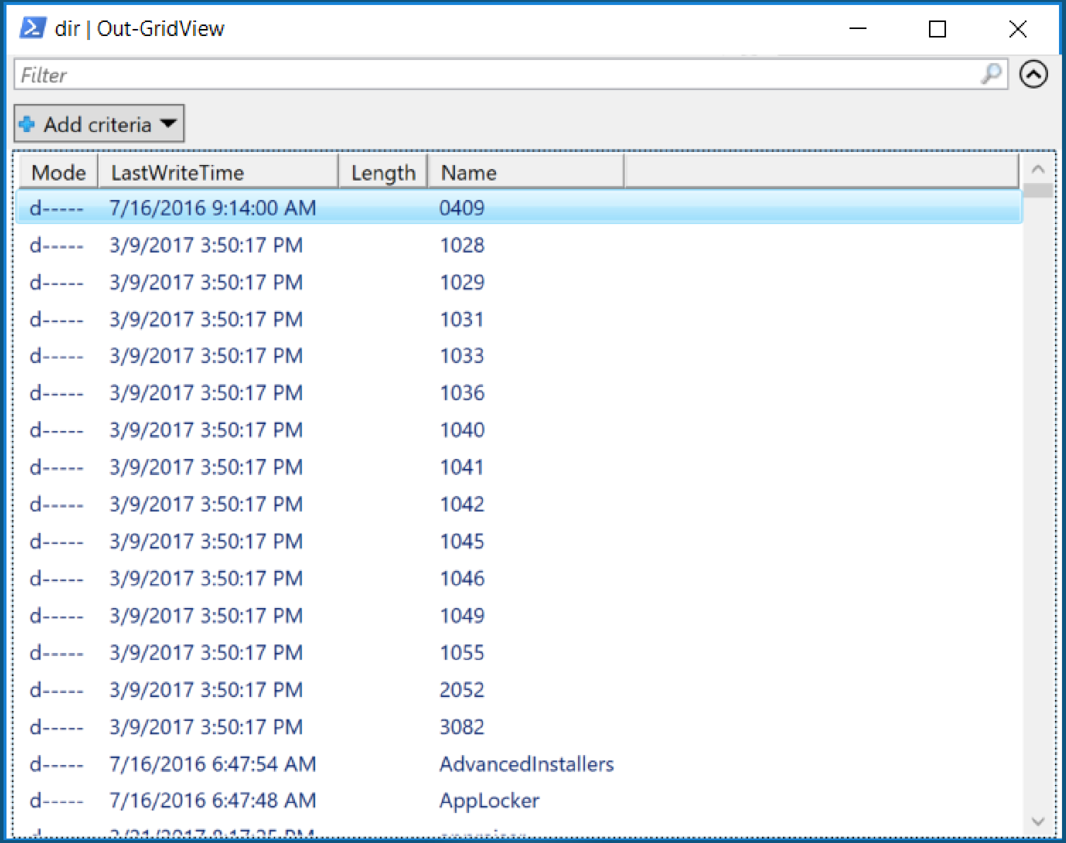
```
C:\> cat test.txt | Measure-Object -Line -Word -Character

Lines Words Characters Property
----- ----- ---------- --------
    2    17         73
```

Out-GridView

Sends output to an interactive table in a separate window.



```
C:\Windows\System32> dir | Out-GridView
C:\Windows\System32> _
```

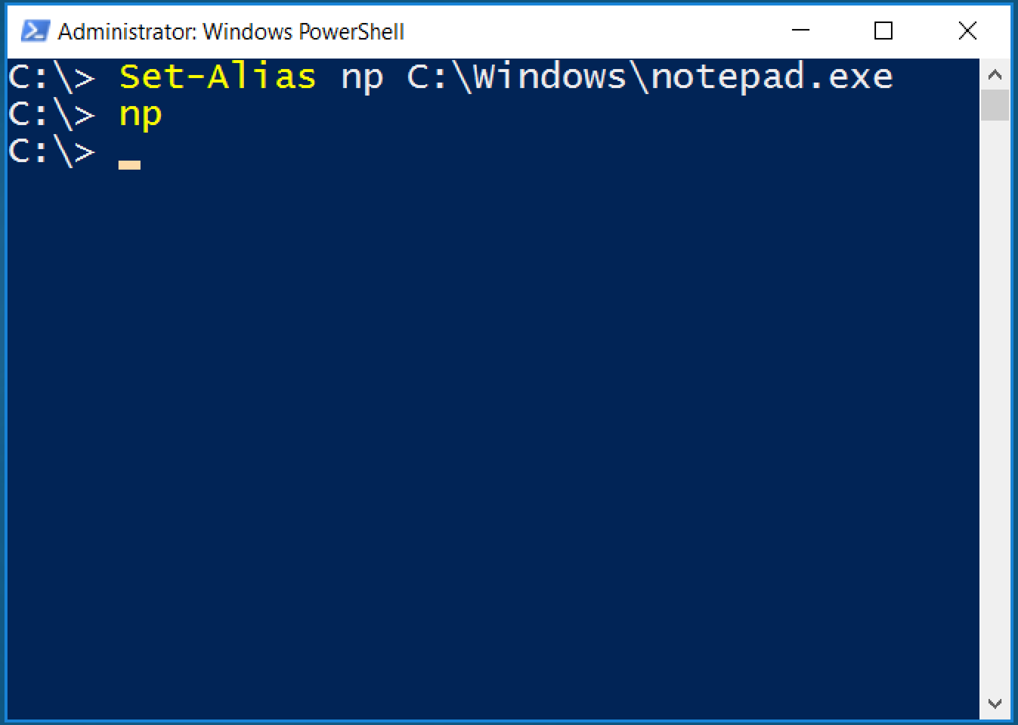


Mode	LastWriteTime	Length	Name
d-----	7/16/2016 9:14:00 AM		0409
d-----	3/9/2017 3:50:17 PM		1028
d-----	3/9/2017 3:50:17 PM		1029
d-----	3/9/2017 3:50:17 PM		1031
d-----	3/9/2017 3:50:17 PM		1033
d-----	3/9/2017 3:50:17 PM		1036
d-----	3/9/2017 3:50:17 PM		1040
d-----	3/9/2017 3:50:17 PM		1041
d-----	3/9/2017 3:50:17 PM		1042
d-----	3/9/2017 3:50:17 PM		1045
d-----	3/9/2017 3:50:17 PM		1046
d-----	3/9/2017 3:50:17 PM		1049
d-----	3/9/2017 3:50:17 PM		1055
d-----	3/9/2017 3:50:17 PM		2052
d-----	3/9/2017 3:50:17 PM		3082
d-----	7/16/2016 6:47:54 AM		AdvancedInstallers
d-----	7/16/2016 6:47:48 AM		AppLocker

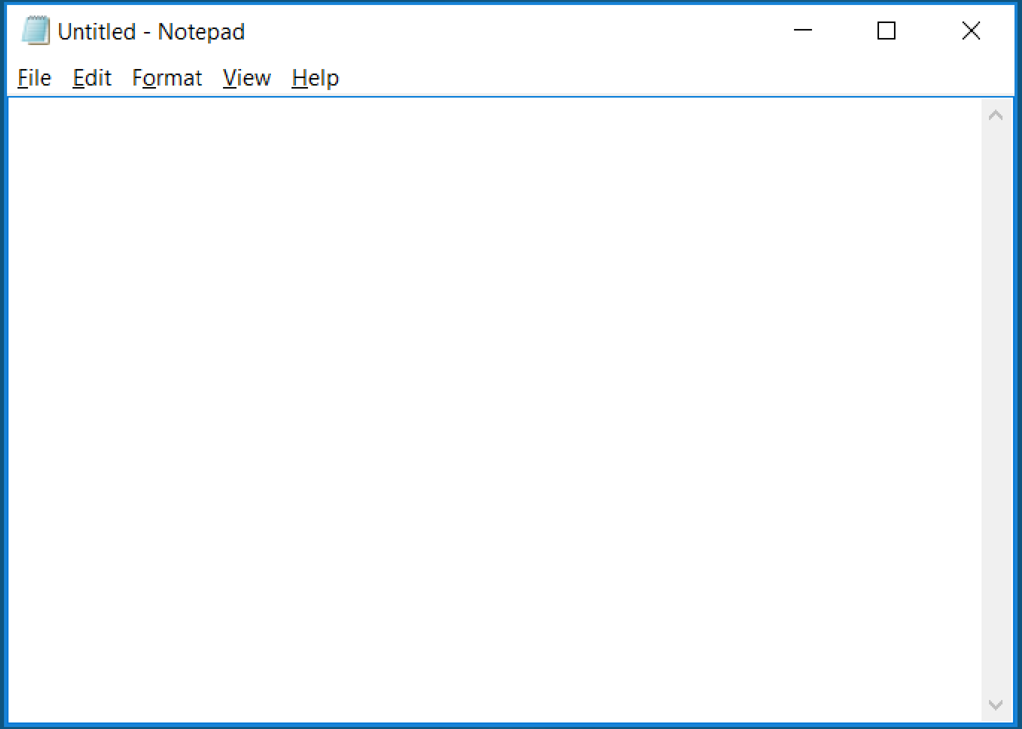
Out-GridView Demo

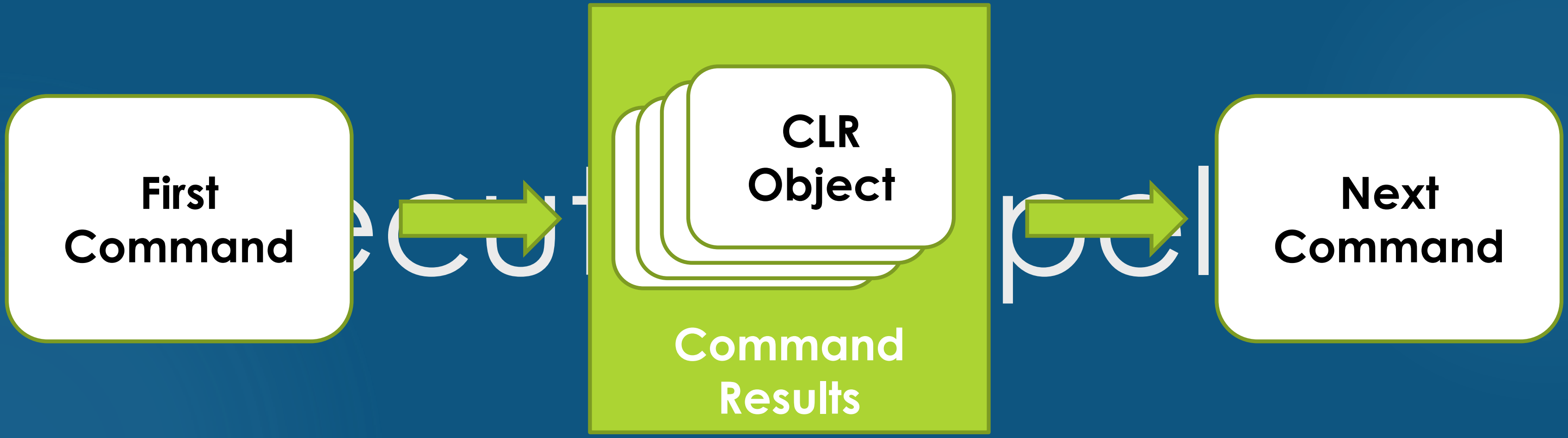
Set-Alias

Creates or changes an alias for a cmdlet or other command element in the current Windows PowerShell session.



```
C:\> Set-Alias np C:\Windows\notepad.exe
C:\> np
C:\> _
```




First Command
CLR Object
Command Results
Next Command

Variables

Variable are declared with a dollar sign

```
C:\> $dev = “C:\development”
```

Arrays

Arrays are declared with an @() syntax or just using commas separated items.

```
C:\> $pets = @(“dog”, “cat”, “hamster”)

C:\> $pets = “snake”, “gorilla”, “donkey”
```

Hash Tables

Hash tables are declared using
@{name1="val1"; name2="val2"; …} syntax.

```
C:\> $myPets = @{dog="Dakota"; cat="Cooper"}
```

Other notable items

Comparisons use -eq, -gt, -lt, -like, etc..
Where-Object (alias where)
Select-Object (alias) select
Sort-Object (alia sort)

if/else

```
C:\> if (‘’ -eq 0) {‘equal’} else {‘not equal’}
not equal
C:\>
```

for loop

```
C:\> for ($i=0; $i -lt 2; $i++) { echo $i }
0
1
C:\>
```

foreach loop

(alias for ForEach-Object)

```
C:\> dir *.txt | foreach {$_.name, $_.length}
File1.txt
1002
File2.txt
245
File3.txt
100
C:\>
```

while, do/while, do/until

```
C:\> $i = 0
C:\> while ($i -lt 2) {echo $i; $i++}
0
1

C:\> do {echo $i; $i--} while($i -gt 0)
2
1

C:\> do {echo $i; $i++} until($i -eq 3)
1
2
```

PowerShell Providers

Functions

```
C:\> function add($x, $y)
>> {
>> $x + $y
>> }
C:\> add 2 3
5
C:\>
```

Writing a Cmdlet in C#

Profile

A profile is a PowerShell script used to setup the execution environment

Current User, Current Host	Current User, All Hosts
`$Home\Documents\WindowsPowerShell\Profile.ps1`	`$Home\[My ]Documents\Profile.ps1`

All Users, Current Host (Console)	All Users, All Hosts
`$PsHome\Microsoft.PowerShell_profile.ps1`	`$PsHome\Profile.ps1`

Current User, Current Host (ISE)	All Users, Current Host (ISE)
`$Home\Documents\WindowsPowerShell\Microsoft.PowerShellISE_profile.ps1`	`$PsHome\Microsoft.PowerShellISE_profile.ps1`

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

Learn More

- Websites
  - PowerShell.org
  - blogs.technet.microsoft.com/heyscriptingguy
  - blogs.msdn.microsoft.com/powershell
- Books
  - Windows PowerShell for Developers by Douglas Finke
- Puralsight
  - Beginning PowerShell Scripting for Developers by Robert Cain

Thank you

Slides available on my github site

https://github.com/brentestewart/PowerShellTalk

Brent Stewart

Alien Arc Technologies, LLC

@BrentEStewart

About.me/brentstewart