

Wprowadzenie do PowerShell'a na podstawie: „Getting Started with Microsoft PowerShell”

Michał Frąckiewicz

2017

Twórcy Powershell'a:

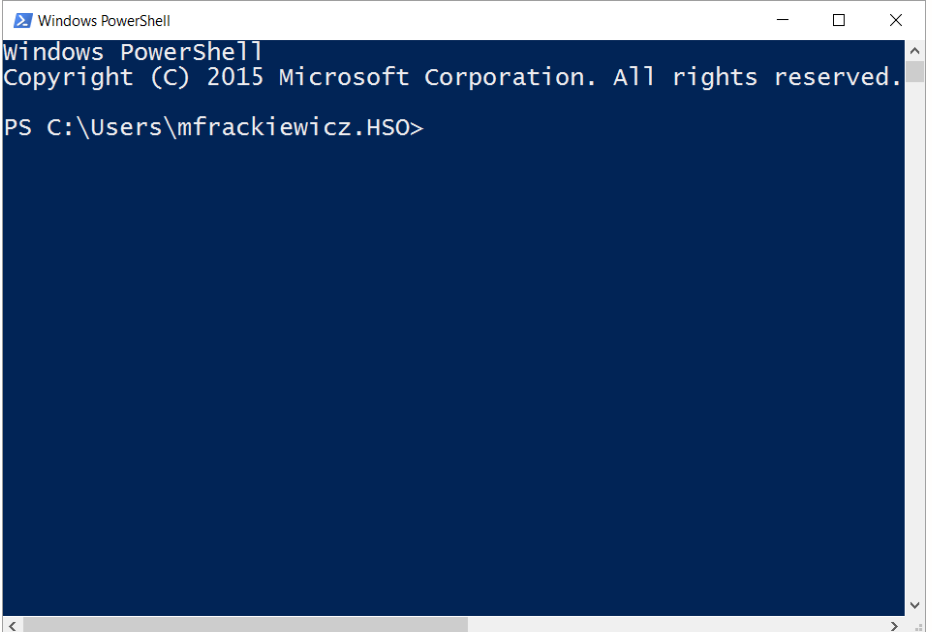
Jeffrey Snover,
Bruce Payette,
James Truher



- Autor PowerShell'a, Jeffrey Snover, zobowiązał się: „naucz się kolejnego języka programowania, a nie będziesz musiał uczyć się ponownie”. Zobowiązanie pozostaje aktualne, od pierwszej wersji: Listopad, 2006, obecnie (2017) stabilna 5.1 a wydano 6.
- Powershell {**PS**} jest językiem który w założeniu ma zachęcać do **eksploracji**, i jest mocno powiązany z konstrukcją języka angielskiego.
- Linux jest indywidualistą, Windows jest środowiskiem grupowym. Powershell jest narzędziem do automatyzacji w dużych środowiskach.
- Głównym celem jest automatyzacja środowisk, a nie środowiska.

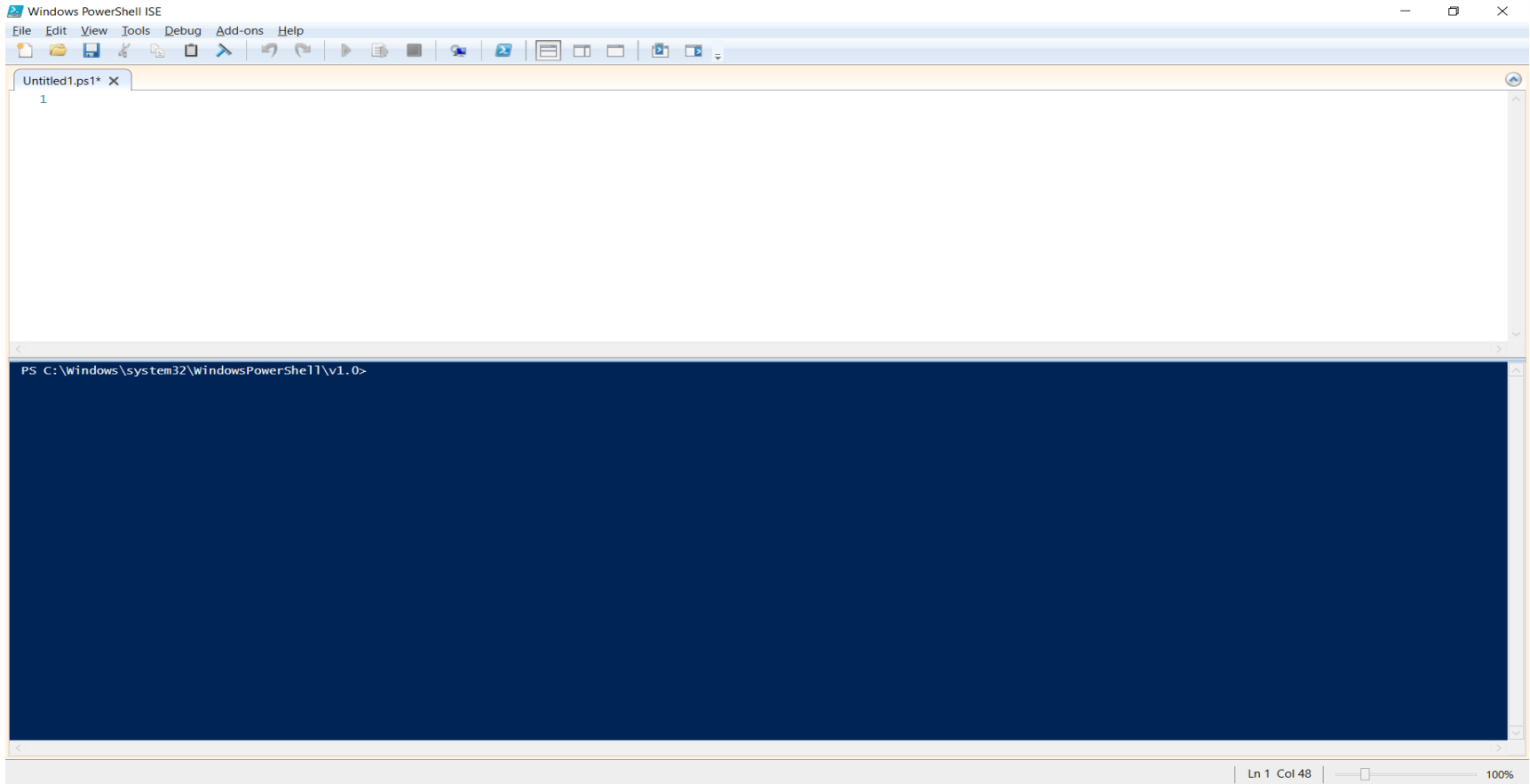
PowerShell pierwsze uruchomienie:

- Jak głosi oficjalna strona MS: „PowerShell comes installed by default in every Windows, starting with Windows 7 SP1 and Windows Server 2008 R2 SP1”
- I jeżeli jednak mamy potrzebę instalacji, to proponuje ściągnąć „**Windows Management Framework**” z: <https://www.microsoft.com/en-us/download/details.aspx?id=54616>
- Weryfikacja wersji, start-> wpisujemy power -> „wybieramy Windows PowerShell”
- Wpisujemy **Get-ho** + wciskamy tabulator, następuje auto-uzupełnienie do **Get-host**.
- Get-ExecutionPolicy – W powershellu został położony duży nacisk na bezpieczeństwo, jednym z jego elementów jest polityka która określa poziom bezpieczeństwa. O bezpieczeństwie porozmawiamy jednak później ..., jednak:
 - „Run as Administrator” jeżeli chcemy wykonywać pracę administratora to trzeba PS uruchomić jako administrator



```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\mfrackiewicz.HSO>
```



PowerShell ISE- dojrzałe narzędzie do pisania skryptów, również ma tryb pracy Administratora.
(Można korzystać jeszcze z Visual Studio...)

Powershell konwencja:

- **Linux** jest systemem ukierunkowanym na tekst, adaptacja tego podejścia w Windowsie nie powiodła się (inna filozofia, orientacja na **API** {Application Programming Interface}). Spowodowało to nową potrzebę, by móc skutecznie automatyzować zadania jak w Linuxie, wymyślono więc Command-let'y: Cmdlets.
- **Cmdlets** : Verb – Noun „Czasownik – Rzeczownik”, tak się porozumiewają ludzie i jest to naturalny sposób komunikacji. Czasownik informuje nas co będzie robione, oraz rzeczownikiem która określa na czym operacja będzie robiona. Rozdzielone są myślnikiem, przykłady, wraz z **aliasami**:
 - Cls – Clear-Host.
 - Cd – Set-Location
 - Dir / ls – Get-Children
 - Type / cat – Get-Content
 - Copy / cp – Copy-Item
 - Get-Alias - gal
- **Natywne** komendy działają, np. ping
- **Cmdlets** nie są osobnymi aplikacjami, są kompilowane w ramach **PS**, co powoduje że wynikiem nie jest tekst a **obiekt**.

Uwagi ogólne:

- Windows zyskuje namiastkę bash'a w Linuxie, uwaga haczyk doświadczenie z basha / Pythona nie zawsze są dobrym rozwiązaniem, da się tak pisać skrypty, ale po co?
- Powershell jest obiektowy, oznacza że ma bardzo duży apetyt na pamięć i jest wolny.
- Można pisać aplikacje okienkowe, ale po co?
- Powershell jest narzędziem które umożliwia zarządzanie i automatyzowanie dużych środowisk, i tam pokazuje swoją siłę, jest to narzędzie dla DevOps'ów.

Ćwiczenie:

Odwiedzić strony:

- <https://blogs.technet.microsoft.com/heyscriptingguy/>.
- <https://www.powershellgallery.com>
- <https://mva.microsoft.com/>
- Google ;-) poszukać źródeł wiedzy o PS.
- Uruchomić Powershella, ISE w trybie administratora i normalnie.
- Kto to jest DevOps