



**OSTRAVSKÁ UNIVERZITA**  
PŘÍRODOVĚDECKÁ FAKULTA

# PowerShell

Ing. Pavel Smolka, Ph.D.

# PowerShell

- PowerShell je skriptovací jazyk a shell od Microsoftu.
- Určen pro administrátory a programátory.
- Umožňuje automatizaci procesů.
- Většina příkazů CMD je do tohoto prostředí integrována.
- Existuje v 64bitové a 32bitové formě.
- Existuje ve verzi konzole nebo ve verzi PowerShell ISE – vhodnější pro tvorbu skriptů a jejich testování.



# Syntaxe PowerShellu

- Výstup PowerShellu je založen na objektech.
- Objekty zde chápeme jako strukturované informace.
- Výstup jednoho příkazu může být vstupem dalšího příkazu.
- Lze vytvářet aliasy příkazů.
- PowerShell pracuje s kanály – předávání výstupů na pozici vstupů následujících příkazů.



# Syntaxe PowerShellu

- Cmdlet - základní příkaz PowerShellu.
- Casesenzitivní
- Jmenná konvence - Verb-Noun
  - Get-Date
  - Get-Location
  - Get-Culture
  - Set-Date
- Výpis dostupných sloves pořídíme pomocí Get-Verb, plné příkazy Get-Command



# Příklady filtrovaného výpisu

Get-Command -Verb get

Get-Command -Verb get -Noun \*tcp\*

Get-Command -Module NetTCPIP -Verb set



# Informace k plné syntaxi

Get-Help Get-Member

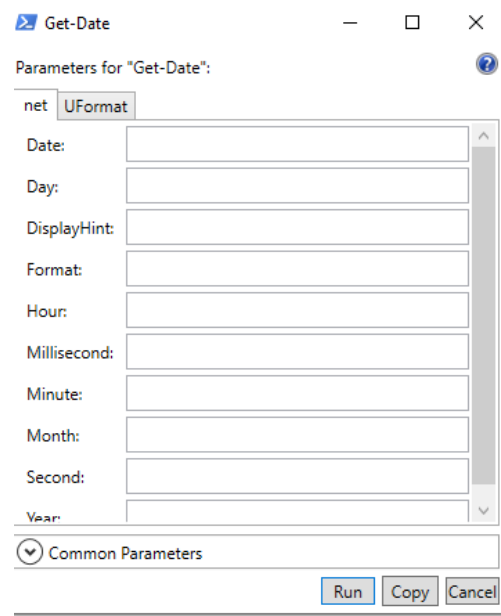
Get-Help Get-Member –Examples

Get-Help Get-Member -Full



# Grafický vstupní ovladač rutiny

## Show-Command Get-Date



# Skripty a bezpečnost

- Jednotlivé rutiny můžeme seskupovat do komplexních skriptů.
- Skripty jsou ukládány s příponou .ps1, .psd1, .psm1.
- Standardně nelze spouštět skripty, ale jen příkazy – otázka bezpečnostních pravidel.
- Bezpečnostní pravidla nastavujeme pomocí **Set-ExecutionPolicy**





# Úrovně bezpečnostních pravidel skriptů PowerShellu

- **Restricted** – prostředí nenačítá konfigurační soubory ani nespouští skripty. Tato úroveň je výchozí.
- **AllSigned** – vyžaduje, aby všechny skripty a konfigurační soubory podepsal důvěryhodný vydavatel.
- **RemoteSigned** – vyžaduje, aby důvěryhodný vydavatel podepsal všechny skripty a konfigurační soubory stažené z Internetu.
- **Unrestricted** – načítá všechny konfigurační soubory a spouští všechny skripty. Spouštíme-li nepodepsaný skript, zobrazí se před spuštěním takového skriptu žádost o povolení.
- **Bypass** – nic není blokováno a nezobrazují se žádná upozornění ani výzvy.
- **Undefined** – odebere aktuálně přiřazené pravidlo spuštění z aktuálního oboru.



# Úrovně bezpečnostních pravidel skriptů PowerShellu II.

Kromě šesti úrovní pravidel spouštění existují tři různé obory pravidel spouštění:

- **Process** – pravidlo spouštění se vztahuje pouze na aktuální proces prostředí PowerShell.
- **CurrentUser** – pravidlo spouštění se vztahuje pouze na aktuálního uživatele.
- **LocalMachine** – pravidlo spouštění ovlivňuje všechny uživatele počítače.

```
Set-ExecutionPolicy -Scope LocalMachine -ExecutionPolicy unrestricted
```



# Syntaxe spuštění skriptu

```
./skript.ps1
```

```
powershell C:\skripty\skript.ps1
```



# Roura - pipeline

- Vezmi výstup jednoho příkazu a pomocí roury jej pošli na vstup příkazu dalšího.

Get-ChildItem | Sort -Property Length





# Zobrazení informací o objektu - procesu

Get-Process | Get-Member | Out-Host -  
Paging



# Příklad filtrování a zobrazení

Get-ChildItem | Sort-Object -Property  
LastWriteTime, Name | Format-Table -  
Property LastWriteTime, Name



# Změna stavu počítače

- Odhlášení:
  - shutdown.exe –l
- Vypnutí:
  - Stop-Computer
- Restart:
  - Restart-Computer (-force)



## Příklady k vyřešení

- Vypište v libovolném adresáři (složce) soubory podle času poslední úpravy.
- Vypište v libovolném adresáři (složce) pět největších souborů.
- Vypište celková velikost souborů v adresáři (složce).





Děkuji za pozornost

