

## POTOKI 1:

### Get-Member, Format- , Select-Object

W PowerShell pipeline działa na dwóch głównych mechanizmach przekazywania danych między komendami: "By Value" i "By Property Name". Oto jak one działają:

#### By Value (Przez Wartość)

W tym przypadku, obiekt wyjściowy z jednej komendy jest przekazywany jako obiekt wejściowy do kolejnej komendy, jeśli ta komenda jest w stanie go przyjąć. Oznacza to, że druga komenda musi być zdefiniowana tak, aby akceptować obiekt tego typu.

`Get-Process | Stop-Process`

W tym przykładzie, `Get-Process` generuje obiekty typu `System.Diagnostics.Process`, które są bezpośrednio przekazywane do komendy `Stop-Process`, która akceptuje obiekty tego typu.`

#### By Property Name (Przez Nazwę Właściwości)

W tym przypadku, jeżeli obiekt wyjściowy z pierwszej komendy nie może być przekazany "By Value" do drugiej komendy, to PowerShell próbuje dopasować nazwy właściwości obiektu wyjściowego do nazw parametrów drugiej komendy.

Te dwa mechanizmy można również łączyć w jednym pipeline, co czyni go bardzo elastycznym i potężnym narzędziem.

## Get-Member

Komenda **Get-Member** w PowerShell jest używana do wyświetlania informacji o obiektach — w tym właściwości, metody i typy obiektów. Jest to bardzo użyteczne narzędzie, szczególnie kiedy pracujesz z obiektami i chcesz zrozumieć, jakie operacje możesz na nich wykonać. Pozwala to również na lepsze zrozumienie struktury obiektów, co jest kluczowe dla efektywnego wykorzystania potoku w PowerShell.

### Przykłady

#### Podstawowe informacje o procesach

```
Get-Process | Get-Member
```

Wyświetla listę właściwości i metod dostępnych dla obiektów procesów.

#### Informacje o dyskach

```
Get-Disk | Get-Member
```

Wyświetla listę właściwości i metod dostępnych dla obiektów dysków.

#### Właściwości i metody dla obiektów łańcucha znaków

```
"hello" | Get-Member
```

Pokazuje, co można zrobić z obiektami typu **String** w PowerShell.

### Właściwości i metody obiektów daty i czasu

```
Get-Date | Get-Member
```

Pokazuje właściwości i metody dostępne dla obiektów daty i czasu.

### Znajdź metody i właściwości obiektów wynikających z polecenia **Get-EventLog**

```
Get-EventLog -LogName Application -Newest 5 | Get-Member
```

### Odkryj metody i właściwości dla obiektów związanych z siecią

```
Get-NetIPAddress | Get-Member
```

### Sprawdź, co możesz zrobić z wynikami polecenia **Get-HotFix**

```
Get-HotFix | Get-Member
```

### Obejrzyj dostępne metody i właściwości dla obiektów użytkowników

```
Get-LocalUser | Get-Member
```

## Format-

W PowerShell mamy kilka podstawowych poleceń służących do formatowania wyników. Oto one:

1. **Format-Table:** Wyświetla dane w formie tabeli. Można zdefiniować szerokość kolumn, które właściwości mają być wyświetlone itd.
2. **Format-List:** Wyświetla każdy obiekt jako listę jego właściwości. Jest to szczególnie przydatne, gdy obiekt ma wiele właściwości, które nie mieszczą się w jednym wierszu w formie tabeli.
3. **Format-Wide:** Wyświetla jedną właściwość obiektu w formie listy, która zajmuje całą dostępną szerokość okna.
4. **Format-Custom:** Umożliwia zaawansowane formatowanie wyników. Jest to bardziej zaawansowane narzędzie, które można użyć do tworzenia niestandardowych widoków.
5. **Out-GridView:** Wyświetla dane w interaktywnym oknie z możliwością sortowania i filtrowania (choć to nie jest stricte polecenie formatujące, jest często używane do tych celów).

## Przykłady

### Format-Table

Wyświetlenie listy procesów w formie tabeli, pokazując tylko nazwę procesu i zużycie pamięci:

```
Get-Process | Format-Table -Property ProcessName, WS
```

## Format-List

Wyświetlenie informacji o dyskach w formie listy właściwości:

```
Get-Disk | Format-List
```

## Format-Wide

Wyświetlenie tylko nazw procesów na pełnej szerokości okna:

```
Get-Process | Format-Wide -Property ProcessName
```

## Out-GridView

Wyświetlenie listy procesów w interaktywnym oknie:

```
Get-Process | Out-GridView
```

# Select-Object

w PowerShell `Select-Object`, jest jednym z najbardziej podstawowych i użytecznych poleceń w pracy z danymi. Polecenie `Select-Object` w PowerShell służy do wybierania określonych obiektów lub ich właściwości z potoku (pipeline). Może być używane do wielu zadań, takich jak wyodrębnianie pewnych danych z listy obiektów, wyświetlanie tylko niektórych kolumn danych w tabeli lub tworzenie nowych, uproszczonych obiektów na podstawie istniejących.

## Przykłady

**Możesz wybrać jedną właściwość z obiektu, na przykład `Name` z listy procesów.**

```
Get-Process | Select-Object -Property Name
```

**Aby wyświetlić więcej niż jedną właściwość, wystarczy je wymienić po przecinku.**

```
Get-Service | Select-Object -Property Status, Name, DisplayName
```

**Użyj parametru `-Unique` aby wyświetlić tylko unikatowe wystąpienia wybranej właściwości.**

```
Get-Process | Select-Object -Property ProcessName -Unique
```

**Parametr `-First` pozwala na wyświetlenie pierwszych `n` obiektów z listy.**

```
Get-Process | Select-Object -First 5
```

**Z kolei parametr `-Last` pozwala na wybranie ostatnich `n` obiektów.**

```
Get-Process | Select-Object -Last 5
```

**Możesz stworzyć nowy obiekt z istniejących właściwości, używając składni hasztablicy.**

```
Get-Process | Select-Object @{Name='ProcessName';Expression={$_.Name}},  
@{Name='MemoryUsage';Expression={$_.WorkingSet}}
```

**Jeśli nie znasz dokładnych nazw właściwości, możesz użyć symboli wieloznacznych do ich wyszukania.**

```
Get-Process | Select-Object -Property Peak*
```

To wybierze wszystkie właściwości, które zaczynają się od litery `Peak`.

**Pominiecie określonej liczby pierwszych obiektów z listy.**

```
Get-Process | Select-Object -Skip 10
```

Każdy z powyższych przykładów pokazuje różne sposoby wykorzystania polecenia `Select-Object`.

**Sort-Object** sortuje obiekty na podstawie wartości jednej lub kilku właściwości obiektu.

- **-Property:** Określa, według której właściwości obiektu nastąpi sortowanie.
- **-Descending:** Sortuje obiekty w kolejności malejącej.
- **-CaseSensitive:** Sortuje obiekty z uwzględnieniem wielkości liter.
- **-Unique:** Usuwa zduplikowane obiekty.

Przykład z parametrem **-Descending** i **-Property**:

```
5, 1, 9, 3, 7 | Sort-Object -Descending
```

Możesz użyć **Sort-Object** do sortowania danych uzyskanych z różnych cmdletów w PowerShell, takich jak **Get-Process**, **Get-Service** czy **Get-EventLog**.

- **Sortowanie procesów według zużycia pamięci:**

```
Get-Process | Sort-Object WS -Descending
```

- **Sortowanie usług systemowych według statusu:**

```
Get-Service | Sort-Object Status
```

**Measure-Object** wykonuje obliczenia na obiektach i zwraca wynik jako obiekt.

- **-Sum:** Sumuje wszystkie liczby.
- **-Average:** Oblicza średnią wszystkich liczb.
- **-Maximum:** Znajduje największą wartość.
- **-Minimum:** Znajduje najmniejszą wartość.
- **-Line:** Zlicza ilość linii w tekście.
- **-Word:** Zlicza ilość słów w tekście.
- **-Character:** Zlicza ilość znaków w tekście.

Przykład z parametrem **-Sum** i **-Maximum**:

5, 1, 9, 3, 7 | Measure-Object -Sum -Maximum

**Measure-Object** może być używany do analizy danych systemowych, takich jak liczba uruchomionych procesów, średnie zużycie pamięci i inne.

- **Zliczenie uruchomionych procesów:**

Get-Process | Measure-Object

- **Obliczenie średniego zużycia pamięci przez procesy:**

Get-Process | Measure-Object WS -Average