

Where-Object

Komenda **Where-Object** w PowerShell pozwala na filtrowanie obiektów przesyłanych przez potok. Działa na zasadzie logicznego testu, który wykonuje na każdym obiekcie przesyłanym do niej. Obiekty, które spełniają warunek, są przesyłane dalej przez potok.

Parametry

- **-FilterScript { script_block }**: Blok skryptu, który zawiera warunek logiczny do oceny. Obiekty, które spełniają ten warunek, są przesyłane dalej. W bloku skryptu można odwoływać się do obiektu przesyłanego przez potok za pomocą zmiennej `$_`.
- **-InputObject <PSObject>**: Określa obiekty, które mają być filtrowane. Parametr jest rzadko używany, ponieważ zwykle obiekty są przesyłane do **Where-Object** przez potok.

Przykłady zastosowania

1. **Filtrowanie liczb**: Wybiera tylko liczby większe od 5.

```
1..10 | Where-Object { $_ -gt 5 }
```

2. **Filtrowanie procesów**: Wybiera tylko procesy, które zużywają więcej niż 100 MB pamięci.

```
Get-Process | Where-Object { $_.WorkingSet -gt 100MB }
```

3. **Filtrowanie plików**: Wybiera tylko pliki tekstowe.

```
Get-ChildItem | Where-Object { $_.Extension -eq '.txt' }
```

Foreach-Object

Komenda **Foreach-Object** w PowerShellu jest używana do wykonania określonej akcji na każdym obiekcie przesyłanym przez potok. Jest to szczególnie przydatne, gdy chcemy zautomatyzować zadania dla wielu obiektów jednocześnie.

Parametry

- **-Process**: Blok kodu, który będzie wykonywany na każdym obiekcie w potoku.
- **-Begin**: Blok kodu wykonywany przed rozpoczęciem przetwarzania obiektów w potoku.
- **-End**: Blok kodu wykonywany po zakończeniu przetwarzania wszystkich obiektów w potoku.

Przykłady zastosowania

Wyświetlenie liczb od 1 do 5

```
1..5 | Foreach-Object { Write-Host $_ }
```

Wyświetlenie kwadratów liczb od 1 do 5

```
1..5 | Foreach-Object { Write-Host ($_ * $_) }
```

Zsumowanie liczb od 1 do 5

```
$suma = 0 1..5 | Foreach-Object { $suma += $_ } Write-Host $suma
```