

Operatory relacji

- **-eq**
Porównuje, czy dwie wartości są równe.
5 -eq 5 zwraca **\$true**.
- **-ne**: różność
Porównuje, czy dwie wartości są różne.
5 -ne 6 zwraca **\$true**.
- **-gt**: większe niż
Sprawdza, czy lewa wartość jest większa niż prawa.
7 -gt 5 zwraca **\$true**.
- **-ge**: większe lub równe
Sprawdza, czy lewa wartość jest większa lub równa prawej.
5 -ge 5 zwraca **\$true**.
- **-lt**: mniejsze niż
Sprawdza, czy lewa wartość jest mniejsza niż prawa.
3 -lt 4 zwraca **\$true**.
- **-le**: mniejsze lub równe
Sprawdza, czy lewa wartość jest mniejsza lub równa prawej.
4 -le 4 zwraca **\$true**.
- **-like**: dopasowanie wzorca
Porównuje ciągi znaków przy użyciu znaków wieloznacznych (*, ?).
'Hello' -like 'H*' zwraca **\$true**.
- **-notlike**: brak dopasowania wzorca
Porównuje ciągi znaków, zwracając **\$true**, jeśli nie pasują do wzorca.
'Hello' -notlike 'W*' zwraca **\$true**.
- **-match**: dopasowanie wyrażenia regularnego
Porównuje ciągi znaków przy użyciu wyrażeń regularnych.
'Hello' -match 'ell' zwraca **\$true**.

- **-notmatch**: brak dopasowania wyrażenia regularnego
Porównuje ciągi znaków, zwracając **\$true**, jeśli nie pasują do wyrażenia regularnego.
'Hello' -notmatch 'world' zwraca **\$true**.
- **-contains**: zawiera
Sprawdza, czy kolekcja zawiera daną wartość.
@(1, 2, 3) -contains 2 zwraca **\$true**.
- **-notcontains**: nie zawiera
Sprawdza, czy kolekcja nie zawiera danej wartości.
@(1, 2, 3) -notcontains 4 zwraca **\$true**.
- **-in**: jest w kolekcji
Sprawdza, czy wartość jest elementem kolekcji.
2 -in @(1, 2, 3) zwraca **\$true**.
- **-notin**: nie jest w kolekcji
Sprawdza, czy wartość nie jest elementem kolekcji.
4 -notin @(1, 2, 3) zwraca **\$true**.

-AND , - OR

-or: Jest to operator logiczny 'LUB'. Zwraca wartość True, jeśli przynajmniej jeden z operandów (lewy lub prawy) jest True. Jeśli oba operandy są False, wtedy wynik operacji również jest False. Przykład użycia:

```
$false -or $true
```

W tym przypadku całe wyrażenie zwróci **True**

-and: Jest to operator logiczny 'I'. Zwraca wartość True tylko wtedy, gdy oba operandy (lewy i prawy) są True. Jeśli którykolwiek z operandów jest False, wynik operacji również jest False. Przykład użycia:

```
$false -and $true
```

W tym przypadku, wyrażenie zwróci False

Operatory Arytmetyczne

+ (**Dodawanie**) - Dodaje dwie liczby.

```
5 + 3 # Wynik: 8
```

- (**Odejmowanie**) - Odejmuje prawą liczbę od lewej.

```
5 - 3 # Wynik: 2
```

***** (**Mnożenie**) - Mnoży dwie liczby.

```
5 * 3 # Wynik: 15
```

/ (**Dzielenie**) - Dzieli lewą liczbę przez prawą.

```
5 / 2 # Wynik: 2.5
```

% (**Reszta z dzielenia**) - Zwraca resztę z dzielenia lewej liczby przez prawą.

```
5 % 3 # Wynik: 2
```

++ (**Inkrementacja**) - Zwiększa wartość zmiennej o 1.

```
$x = 5
$x++ # $x teraz wynosi 6
```

-- (**Dekrementacja**) - Zmniejsza wartość zmiennej o 1.

```
$x = 5
$x-- # $x teraz wynosi 4
```

-= (**Odejmowanie i przypisanie**) - Odejmuje wartość od zmiennej i przypisuje nową wartość do tej samej zmiennej.

```
$x = 5
$x -= 2 # $x teraz wynosi 3
```

+= (**Dodawanie i przypisanie**) - Dodaje wartość do zmiennej i przypisuje nową wartość do tej samej zmiennej.

```
$x = 5
$x += 2 # $x teraz wynosi 7
```

***=** (**Mnożenie i przypisanie**) - Mnoży zmienną przez wartość i przypisuje nową wartość do tej samej zmiennej.

```
$x = 5
$x *= 2 # $x teraz wynosi 10
```

/= (**Dzielenie i przypisanie**) - Dzieli zmienną przez wartość i przypisuje nową wartość do tej samej zmiennej.

```
$x = 10
$x /= 2 # $x teraz wynosi 5
```

%= (**Reszta z dzielenia i przypisanie**) - Oblicza resztę z dzielenia zmiennej przez wartość i przypisuje nową wartość do tej samej zmiennej.

```
$x = 10
$x %= 3 # $x teraz wynosi 1
```