

Zmienne tablicowe – info

Tworzenie zmiennych tablicowych

Wczytywanie tablic z pliku

Odwoływanie się do elementów tablicy

Właściwości tablic

Przyjrzymy się zmiennym tablicowym.

Czasami jest potrzeba pracy na obiekcie, który składa się z kilku pozycji (rekordów).

```
$compNames="DESKTOP007, DESKTOP008, DESKTOP009"
```

```
$compNames
```

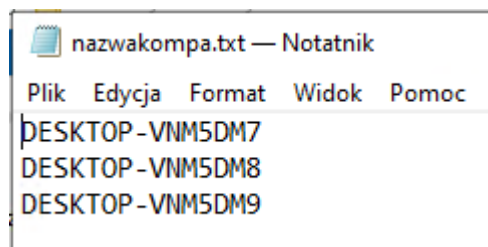
```
1 $compNames="DESKTOP007, DESKTOP008, DESKTOP009"
2 $compNames
```

```
DESKTOP007, DESKTOP008, DESKTOP009
```

Mamy tekst rozdzielony przecinkami, to nie jest lista.

Gdyby napisy znajdujące się w cudzysłowach porozdzielać przecinkami to była by lista napisów.

Utworzymy plik umieścimy w nim nazwy komputerów



```
Get-Content C:\Users\admin\nazwakompa.txt
```

```
3 Get-Content C:\Users\admin\nazwakompa.txt
```

```
DESKTOP007
DESKTOP008
DESKTOP009
```

Wczytamy zawartość pliku do zmiennej i dostaniemy jakby listę, nazwy komputerów są w trzech oddzielnych liniijkach

Wczytamy zawartość pliku do zmiennej

```
$compNames = Get-Content C:\Users\admin\nazwakompa.txt
```

```
4 $compNames = Get-Content C:\Users\admin\nazwakompa.txt
```

```
PS C:\Windows\system32> $compNames = Get-Content C:\Users\admin\nazwakompa.txt
```

Sprawdzamy wartość zmiennej

```
2 $compNames  
DESKTOP007  
DESKTOP008  
DESKTOP009
```

Tak wygląda tablica

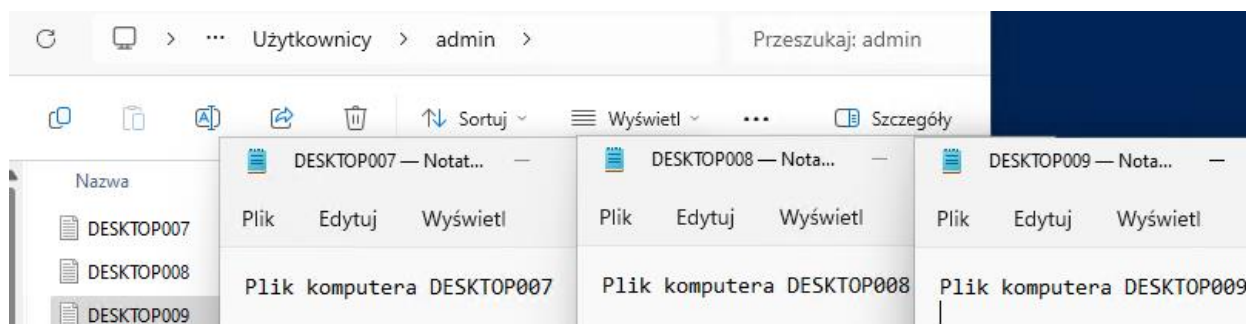
Tablice można przesłać do innej komendy przez potok (\$_ oznacza to co przyszło na wejściu do potoku)

```
$compNames | ForEach-Object { "Działą $_" }
```

```
5 $compNames | ForEach-Object { "Działą $_" }  
Działą DESKTOP007  
Działą DESKTOP008  
Działą DESKTOP009
```

Coś bardziej pożytecznego

Mamy 3 pliki jak poniżej o podobnej zawartości



Potokiem przesyłasz nazwy komputerów do ForEach który wywołuje komendę Get-Content odczytującą zawartość pliku, ścieżka do pliku jest budowana dynamicznie, bo tekst zamknięty w cudzysłowie będzie interpretowany i PowerShell zamiast \$_ umieści w wbudowanej ścieżce nazwy komputerów.

```
$compNames | ForEach-Object { Get-Content "C:\Users\admin\$_txt" }
```

```
6 $compNames | ForEach-Object { Get-Content "C:\Users\admin\$_txt" }  
Plik komputera DESKTOP007  
Plik komputera DESKTOP008  
Plik komputera DESKTOP009  
Get-Content : Cannot find path 'C:\Users\admin\_.txt' because it does not exist.  
At line:1 char:31  
+ $compNames | ForEach-Object { Get-Content "C:\Users\admin\_.txt" }  
+ ~~~~~  
+ CategoryInfo          : ObjectNotFound: (C:\Users\admin\_.txt:String) [Get-Content], ItemNotFoundException  
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand
```

podczas próby odczytania zawartości plików, zmienna \$compNames zawiera pusty element, co prowadzi do błędu, ponieważ PowerShell próbuje odczytać plik o nazwie .txt, który nie istnieje.

Można zbudować polecenie, które połączy się z tymi komputerami i wykona polecenia, np. wyświetli katalog, wyświetli usługi, założy konta, zainstaluje komponenty Windows, sprawdzi łączność z tymi komputerami itd.

Wróćmy do tematu tablicy.

Programiści mogą się odwołać do elementów tablicy

PowerShell indeksowanie tablic zaczyna się od zera

`$compNames[0]`

`$compNames[1]`

```
PS C:\Windows\system32> $compNames[0]
DESKTOP007
PS C:\Windows\system32> $compNames[1]
DESKTOP008
```

Informacja o długości tablicy

`$compNames.Count`

```
9 $compNames.Count
3
```

W przypadku tablic zwraca to samo co Count ale w przypadku innych typów może być inaczej

`$compNames.Length`

```
10 $compNames.Length
3
```

Właściwość Length można wywołać też dla elementów tablicy, tu pytamy o długość pierwszego elementu listy

`$compNames[0].Length`

```
7 $compNames[0].Length
10
```

Dziesięć literek

Odwołanie do ostatniego elementu tablicy, Count która określa, ile jest elementów tablicy, ale 0 oznacza pierwszy element więc należy odjąć 1

`$compNames[$compNames.Count-1]`

```
9 $compNames[$compNames.Count-1]
DESKTOP009
```

Można się odwołać bezpośredni do elementu -1

`$compNames[-1]`

```
9 $compNames[$compNames.Count-1]
DESKTOP009
```

Można się odwołać bezpośredni do elementu -2

`$compNames[-2]`

```
PS C:\Windows\system32> $compNames[-2]  
DESKTOP008
```

Można się odwołać bezpośredni do elementu -10 – pusty obiekt

`$compNames[-10]`

```
PS C:\Windows\system32> $compNames[-10]  
  
PS C:\Windows\system32>
```

Praktyczny przypadek korzystania z usług

Pobieramy listę wszystkich usług na komputerze

`$services = Get-Service`

```
PS C:\Windows\system32> $services = Get-Service
```

Mamy tablicę, która zawiera 251 elementów – ilość usług zainstalowanych na komputerze

`$services.Count`

```
PS C:\Windows\system32> $services.Count  
270
```

Aby w tablicy umieścić wyłącznie informacje o uruchomionych usługach

`$runningServices = Get-Service | Where { $_.Status -eq "Running" }`

```
PS C:\Windows\system32> $runningServices = Get-Service | Where { $_.Status -eq "Running" }
```

Ile ich jest

`$runningServices.Count`

```
PS C:\Windows\system32> $runningServices.Count  
93
```

Ile było innych o statusie innym niż uruchomiona

`$services.Count - $runningServices.Count`

```
PS C:\Windows\system32> $services.Count - $runningServices.Count  
177
```

Omówiłem:

Tworzenie zmiennych tablicowych

Wczytywanie tablic z pliku

Odwoływania się do elementów tablicy

Właściwość tablic