

Ściąga z poleceń:

<http://aplitt.zsl.gda.pl/Powershell/Ściąga%20Paula-Januszkiewicz-Powershell-reference-PL.pdf>

Tablice:

```
$a = 1,2,3
```

```
$b = "A","B","C","D"
```

Element number	1	2	3	4
Index number	0	1	2	3
Value in the array	A	B	C	D

```
$B.GetLowerBound(0)
```

```
$B.GetUpperBound(0)
```

```
$B[2]
```

```
$B[-1]
```

```
for($i = 0; $i -le $B.GetUpperBound(0); $i++) {$B[$i]}
```

No i pipe:

```
$b | %{$_}
```

```
$B[2]=6
```

`$B[4]=12 ;-( Nie Działa dlaczego?`

`$B+=12`

`$B -contains 12`

`$b | Sort-Object -Descending`

`$rnd = Get-Random -Count 10 -InputObject (1..100000)`

`Measure-Command -Expression { $arrayA = $arrayA | Sort-Object }`

`Measure-Command -Expression { [array]::Sort($arrayB) }`

```
[array]::IndexOf($B,"B")
```

```
$B[0..($B.GetUpperBound(0)/2)]
```

- \$a = 1..5
- \$b = 6..10
- \$c = 11..15
- \$d = 16..20
- \$array = \$a,\$b,\$c,\$d

```
$comp = gwmi win32_computersystem
```

```
$os = gwmi win32_operatingsystem
```

```
$bios = gwmi win32_bios
```

```
$a = $comp.UserName, $comp.DNSHostName, $comp.Domain
```

```
$b = $os.caption, $os.buildnumber, $os.OSArchitecture
```

```
$c = $bios.name, $bios.description, $bios.manufacturer
```

```
$array = $a,$b,$c
```

```
$array[2][1]
```

```
$a = Get-Service
```

```
$b = Get-Process
```

```
$c = get-date
```

```
$array = $a,$b,$c
```

```
$hashtable = @{1="one";2="two";3="three"}
```

Hashtable jest to struktura danych podobna do tablicy, tylko ma klucze.

```
$ageList = @{}  
$key = 'Kevin'  
$value = 36
```

```
$ageList.add( $key, $value )
```

Lub:

```
$ageList.add( 'Alex', 9 )
```

```
if( $person.age ){...}
```

```
if( $person.age -ne $null ){...}
```

```
if( $person.ContainsKey('age') ){...}
```

```
$person.remove('age')
```

```
$person.clear() lub $person = @{}
```

```
$wlasne = @{  
    name = 'totalSpaceGB';  
    expression = { ($_.used + $_.free) / 1GB }  
}
```

```
$drives = Get-PSDrive | Where Used
```

```
$drives | Select-Object -Property name, $wlasne
```



NIE wykonujemy:

Jeżeli chcemy z PS skonfigurować DHCP'a:

```
Add-DhcpServerv4Scope -Name 'TestNetwork' -StartRange'10.0.0.2' -  
EndRange '10.0.0.254' -SubnetMask '255.255.255.0' -Description  
'Network for testlab A' -LeaseDuration (New-TimeSpan -Days 8) -Type  
"Both"
```

Lub bardziej elegancko (NIE wykonujemy):

```
$DHCPscope = @{  
    Name      = 'TestNetwork'  
    StartRange = '10.0.0.2'  
    EndRange   = '10.0.0.254'  
    SubnetMask = '255.255.255.0'  
    Description = 'Network for testlab A'  
    LeaseDuration = (New-TimeSpan -Days 8)  
    Type = "Both"  
}  
Add-DhcpServerv4Scope @DHCPscope
```

Przykłady pochodzą z:

<https://kevinmarquette.github.io/2016-11-06-powershell-hashtable-everything-you-wanted-to-know-about/>

|

<https://blogs.technet.microsoft.com/heyscriptingguy/2011/12/10/create-a-hash-table-in-powershell-that-contains-hash-tables/>