

Skrypty, funkcje, moduły - KLUCZ

1.

```
# Przykładowe daty modyfikacji plików
```

```
$date1 = Get-Date
```

```
$date2 = (Get-Date).AddDays(-25)
```

```
$date3 = (Get-Date).AddDays(-55)
```

```
# Utworzenie przykładowych plików
```

```
New-Item -Path C:\temp\File1.txt -ItemType File -Value "To jest File1" -Force
```

```
New-Item -Path C:\temp\File2.txt -ItemType File -Value "To jest File2" -Force
```

```
New-Item -Path C:\temp\File3.txt -ItemType File -Value "To jest File3" -Force
```

```
# Ustawienie różnych dat modyfikacji dla plików
```

```
(Get-Item C:\temp\File1.txt).LastWriteTime = $date1
```

```
(Get-Item C:\temp\File2.txt).LastWriteTime = $date2
```

```
(Get-Item C:\temp\File3.txt).LastWriteTime = $date3
```

2. `Get-ChildItem c:\temp -Recurse -File | Where-Object { $_.LastWriteTime -lt '2024-09-16' }`

3. `[DateTime]$OlderThan = (Get-Date).AddDays(-1)`

```
[string]$BaseFolder = 'C:\temp\'
```

4. `Get-ChildItem $BaseFolder -Recurse -File | Where-Object { $_.LastWriteTime -lt $OlderThan }`

5. #Zapisz w pliku `C:\temp\FilesOlderThan.ps1`

```
[CmdletBinding()]
```

```
Param(
```

```
[Parameter(mandatory=$true)]
```

```
[DateTime]$OlderThan = (Get-Date).AddDays(-1),
```

```
[Parameter(mandatory=$true)]
```

```
[string]$BaseFolder
```

)

```
Get-ChildItem $BaseFolder -Recurse -File | Where-Object { $_.LastWriteTime -lt $OlderThan }
```

Parametry skryptu:

1. [CmdletBinding()]:

- Jest to atrybut używany na początku skryptu, który umożliwia użycie zaawansowanych funkcji powiązanych z cmdletami, takich jak obsługa parametrów, potoków (pipeline) danych, logowania i innych.

2. Param():

- Jest to sekcja definiująca parametry, które skrypt może przyjmować.

3. Parameter:

- Każdy parametr jest definiowany w kwadratowych nawiasach [Parameter()]. W Twoim przypadku masz dwa parametry:
 - **OlderThan:**
 - Jest to wymagany parametr typu DateTime, który domyślnie jest ustawiony na datę sprzed 1 dnia ((Get-Date).AddDays(-1)). Oznacza to, że domyślnie skrypt będzie szukać plików starszych niż 1 dzień od bieżącej daty.
 - **BaseFolder:**
 - Jest to wymagany parametr typu string, który określa podstawowy folder, w którym skrypt będzie szukać plików.

Skrypt:

Get-ChildItem \$BaseFolder -Recurse -File:

- Pobiera wszystkie pliki (-File) z podanego \$BaseFolder (i jego podkatalogów, dzięki opcji -Recurse).

Where-Object { \$_.LastWriteTime -lt \$OlderThan }:

- Jest to warunek filtrujący przy użyciu Where-Object, który sprawdza, czy data modyfikacji (LastWriteTime) każdego pliku (\$) jest wcześniejsza (-lt) niż wartość \$OlderThan (czyli daty określonej przez użytkownika).

Jak używać tego skryptu?

```
.\FindOldFiles.ps1 -OlderThan "2025-01-01" -BaseFolder "C:\temp"
```

Powyższe polecenie spowoduje wyszukanie plików w katalogu C:\temp, które mają datę modyfikacji starszą niż 1 stycznia 2025 roku.

6. #W nowym oknie Powershell:

```
C:\Scripts\FilesOlderThan.ps1 -OlderThan '2024-08-16' -BaseFolder 'C:\temp'
```

7. #Zapisz w pliku `c:\temp\usitilies.psm1`

```
function Get-FilesOlderThan
```

```
{
```

```
[CmdletBinding()]
```

```
Param(
```

```
[Parameter(mandatory=$true)]
```

```
[DateTime]$OlderThan = (Get-Date).AddDays(-1),
```

```
[Parameter(mandatory=$true)]
```

```
[string]$BaseFolder
```

```
)
```

```
Get-ChildItem $BaseFolder -Recurse -File | Where-Object { $_.LastWriteTime -lt  
$OlderThan }
```

```
}
```

8. #W nowym oknie PowerShell

załaduj moduł do pamięci

```
Import-Module -Name 'c:\temp\utilities.psm1' -Verbose
```

Komenda powyższa powinna dostarczyć bardziej szczegółowe informacje o tym, co mogło pójść nie tak podczas próby zaimportowania modułu.

uruchom funkcje np

```
Get-FilesOlderThan -OlderThan '2024-09-16' -BaseFolder 'c:\temp'
```

Komenda powyższa uruchamia konkretną funkcję, z zaimportowanego modułu.