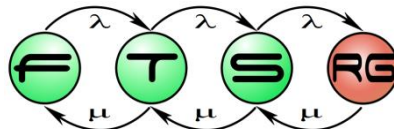


Szkriptelési feladat megoldása

Micskei Zoltán

<http://mit.bme.hu/~micskeiz>



Feladat

Készítsünk egy olyan szkriptet, ami

- paraméterként kap egy könyvtárnevet
- kiírja, hogy hány alkönyvtár van benne
- opcionálisan kiírja, hogy melyik kiterjesztésből van a legtöbb a könyvtárban lévő fájlknál

0. Interfész pontosítása

```
Collect-Content.ps1  
-Folder <String>  
[-Extensions]
```

- Folder paraméter kötelező
- Extensions switch típusú, opcionális

1. Interfész definíció (params)

```
param (  
    [string] $Folder,  
    [switch] $Extensions  
)
```

- Ellenőrzés:

```
Get-Help .\Collect-Content.ps1
```

- Kimenet:

```
Collect-Content.ps1 [[-Folder] <string>] [-Extensions]
```

- Eredmény:

- **Hiba:** Folder nem kötelező így

1. Interfész definíció (params, mandatory)

```
param (  
    [Parameter(Mandatory=$true)][string] $Folder,  
    [switch] $Extensions  
)
```

- Ellenőrzés:

```
Get-Help .\Collect-Content.ps1
```

- Kimenet:

```
Collect-Content.ps1 [-Folder] <string> [-Extensions]  
[<CommonParameters>]
```

- Eredmény:

- ~OK, Folder pozícionális is (maradhat?)

2. Paraméterek ellenőrzése

Milyen értékeket vehetnek fel a paraméterek?

■ Folder

- Nincs megadva
- Megadva



Kezelve (bekéri)

■ Extensions

- Nincs megadva
- Megadva

2. Paraméterek ellenőrzése (finomítás)

Milyen értékeket vehetnek fel a paraméterek?

■ Folder

- Nincs megadva → Kezelve (bekéri)
- Megadva
 - Nem érvényes könyvtár → Hibás eset, kezelni kell
 - Érvényes könyvtár

■ Extensions

- Nincs megadva
- Megadva

3. Paraméterek ellenőrzése (érték)

Folder értékének ellenőrzése

- Vizsgálat: `Test-Path`
- Hibakezelés:
 - Mit tegyünk hiba esetén?
 - Mit tesznek a beépített cmdletek?
 - Pl. `Remove-Item c:\notexists`
 - Eredmény: kivételt dob

```
if (! (Test-Path $Folder)){  
    throw "$Folder does not exist!"  
}
```


3. Paraméterek ellenőrzése (érték)

```
Windows PowerShell
PS C:\temp\_b\steps> Remove-Item c:\notexists
Remove-Item : Cannot find path 'C:\notexists' because it does not exist.
At line:1 char:1
+ Remove-Item c:\notexists
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\notexists:String) [Remove-Item],
ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.RemoveItem
Command

PS C:\temp\_b\steps> .\03-Collect-Content.ps1 c:\notexists
c:\notexists does not exist!
At C:\temp\_b\steps\03-Collect-Content.ps1:8 char:2
+   throw "$Folder does not exist!"
+   ~~~~~
+ CategoryInfo          : OperationStopped: (c:\notexists does not exist!:Strin
g) [], RuntimeException
+ FullyQualifiedErrorId : c:\notexists does not exist!
```

- [Még lehetne finomítani, de egyelőre jó így]

3. Paraméterek ellenőrzése (érték)

- Most már van funkcionalitás + ellenőrzés hozzá
 - Van egy tesztesetünk 😊

- Jegyezzük fel:
 - Bemenet:
 - `Collect-Content.ps1 c:\notexists`
 - Elvárt eredmény:
 - Kivétel

- (Erre lehetne teszt szkriptet írni)

3. Paraméterek ellenőrzése (összesítés)

Milyen értékeket vehetnek fel a paraméterek?

■ Folder

○ Nincs megadva

→ Kezelve (bekéri)

○ Megadva

• Nem érvényes könyvtár

→ Kezelve (kivétel)

• Érvényes könyvtár

→ Finomítani kell még (funkció)

■ Extensions

→

Nem kell ellenőrizni

○ Nincs megadva

○ Megadva

4. Paraméterek értékei (funkció)

Milyen értékeket vehetnek fel a paraméterek?

■ Folder

- Nincs megadva
- Megadva
 - Nem érvényes könyvtár
 - Érvényes könyvtár
 - Nincs alkönyvtára
 - 1 alkönyvtára van
 - több alkönyvtára van
 - alkönyvtárainak is van alkönyvtára

→ Finomítani kell még (funkció)

→ Ilyenkor mit tegyünk?

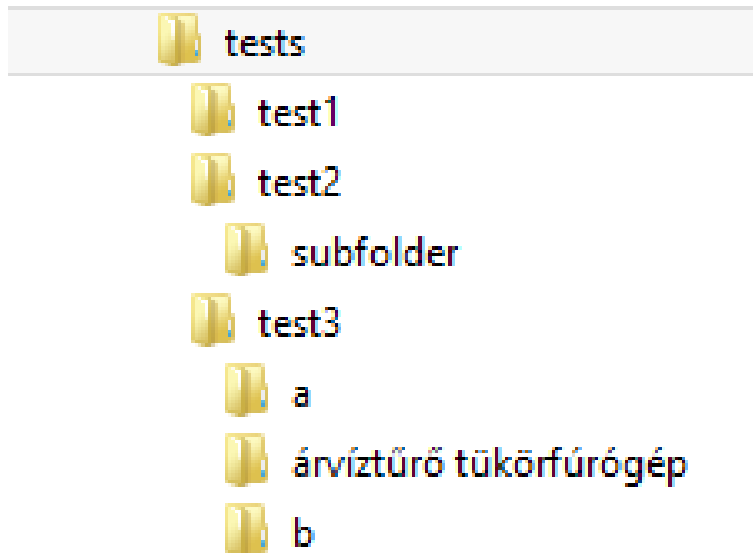
■ Extensions

- Nincs megadva
- Megadva

4. Tesztek a főbb esetekhez

- Mik a tesztek ehhez a programhoz?
 - Igazából adott könyvtárak (fájlokkal és könyvtárakkal)
 - Hozzuk létre ilyen példa könyvtárakat!
- Válasszuk ki a kód megírása előtt (lásd TDD)
 - Segít végiggondolni a funkcionalitást
 - Kis, ellenőrzött kódrészleteink lesznek
- [Lehetne izolálni (mock, stub...), hogy ne függjünk a fájlrendszerrel, de most jó ez így]

4. Tesztek a főbb esetekhez



- test1: nincs alkönyvtára
- test2: 1 alkönyvtára
- test3: több alkönyvtár (+szóköz, ékezetes betűk)

4. Tesztek a főbb esetekhez

Tesztesetek:

- 1.
 - Bemenet:
 - Elvárt eredmény:
- 2.
 - Bemenet:
 - Elvárt eredmény:
- 3.
 - Bemenet:
 - Elvárt eredmény:

Abszolút vagy relatív elérés?

Collect-Content.ps1 test1
<Mi legyen a kimenet formája?>

4. Tesztek a főbb esetekhez

Tesztesetek:

- 1.
 - Bemenet: `Collect-Content.ps1 test1`
 - Elvárt eredmény: `Subfolders in <Folder> : 0`
- 2.
 - Bemenet: `Collect-Content.ps1 test2`
 - Elvárt eredmény: `Subfolders in <Folder> : 1`
- 3.
 - Bemenet: `Collect-Content.ps1 test3`
 - Elvárt eredmény: `Subfolders in <Folder> : 3`

4. Alkönyvtárak megszámlálás

```
# count subfolders
```

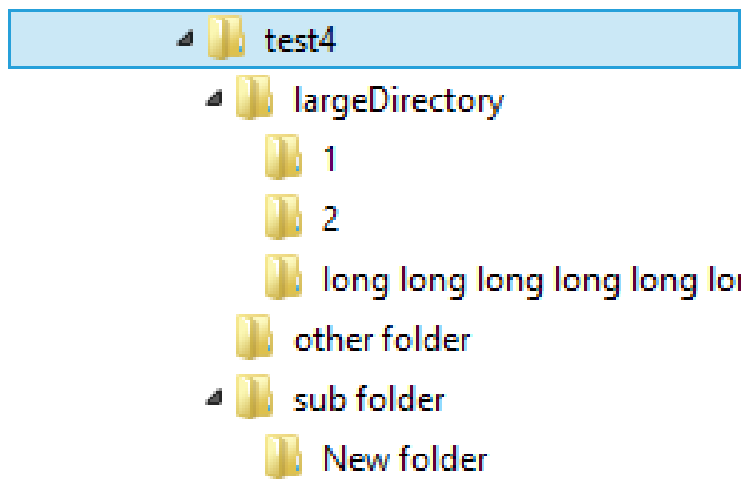
```
$subfolders = (Get-ChildItem -Path $Folder -Directory).Count
```

```
Write-Output "Subfolders in $Folder : $subfolders"
```

- Futtassuk a tesztjeinket!
 - Most már jól jönne legalább egy szkript a bemenetekkel
- Eredmény: **OK**

5. Alkönyvtárak megszámlolása (rekurzív)

- Bővítsük a tesztek a rekurzív számolással



```
Collect-Content.ps1 .\tests\test4
```

```
Subfolders in .\tests\test4 : 3
```

- **Eredmény: HIBA**

5. Alkönyvtárak megszámlolása (rekurzív)

- Bővítsük a kódot is

```
# count subfolders
```

```
$subfolders = (Get-ChildItem -Path $Folder -Directory -Recurse).Count
```

```
Write-Output "Subfolders in $Folder : $subfolders"
```

- Nézzük meg, hogy jó-e az új tesztre
- Nézzük meg, hogy nem rontotta-e el a régieket (!)

6. Kiterjesztések megszámlálása

Mik az esetek a kiterjesztések számolásakor?

- Nincs egy fájl sem
- Csak 1 fájl van, van kiterjesztése
- Több fájl, többféle kiterjesztéssel
- Több fájl, többféle kiterjesztéssel, több legtöbb kiterjesztés
- Van olyan fájl, aminek nincs kiterjesztése

6. Kiterjesztések megszámlálása

Tesztesetek:

- Egybe az előző könyvtárakkal (előny/hátrány?)

test1 (üres)

test2
file.txt

test3
New Text Document.txt
other.txt
szöveg.txt
command.cmd
lista.cmd

test4

New Text Document.txt
other.txt
command.cmd
lista.cmd
program.exe

6. Kiterjesztések megszámlolása

Első próbálkozás:

```
$maxExtension = Get-ChildItem -Path $Folder -Recurse -File |  
Group-Object -Property extension |  
Sort-Object -Property Count -Descending |  
Select-Object -First 1
```

```
Write-Output "Most frequent extension: $($maxExtension.Name)  
($($maxExtension.Count))"
```

Eredmény:

- **HIBA**, ha nincs fájl: Most frequent extension: (0)
- **HIBA**, ha több kiterjesztés a leggyakoribb

7. Kiterjesztések megszámlolása (javítás)

```
# count extensions
if ($Extensions){
    $extensionsGroup = Get-ChildItem -Path $Folder -Recurse -File |
        Group-Object -Property extension
    $maximalCount = ($extensionsGroup |
        Measure-Object -Property Count -Maximum).Maximum

    # get that / those extensions that are the most frequent
    $frequentExtension = $extensionsGroup | ? {$_.Count -eq $maximalCount}

    if ($extensionsGroup -eq $null){
        Write-Output "No files in the folder"
    } elseif ($frequentExtension.Length -eq 1) {
        Write-Output "Most frequent extension: $($frequentExtension.Name)
($($frequentExtension.Count))"
    } else {
        Write-Output "Most frequent extensions:"
        $frequentExtension | % {Write-Output " $($_.Name) ($($_.Count))"}
    }
}
```

7. Kiterjesztések megszámlolása (javítás)

Tesztek eredménye:

```
Collect-Content.ps1 -Extensions -Folder .\tests\test1  
Subfolders in .\tests\test1 : 0  
No files in the folder
```

```
Collect-Content.ps1 -Extensions -Folder .\tests\test2  
Subfolders in .\tests\test2 : 1  
Most frequent extension: .txt (1)
```

```
Collect-Content.ps1 -Extensions -Folder .\tests\test3  
Subfolders in .\tests\test3 : 3  
Most frequent extension: .txt (3)
```

```
Collect-Content.ps1 -Extensions -Folder .\tests\test4  
Subfolders in .\tests\test4 : 7  
Most frequent extensions:  
  .cmd (2)  
  .txt (2)
```

```
Collect-Content.ps1 -Extensions -Folder .\tests\test5  
Subfolders in .\tests\test5 : 0  
Most frequent extension: (2)
```

Nincs kiterjesztése
a fájlnak, mi az
elvárt?

8. Fejkomment hozzáadása

<#

.SYNOPSIS

Counts the subfolders and most frequent extensions in a folder.

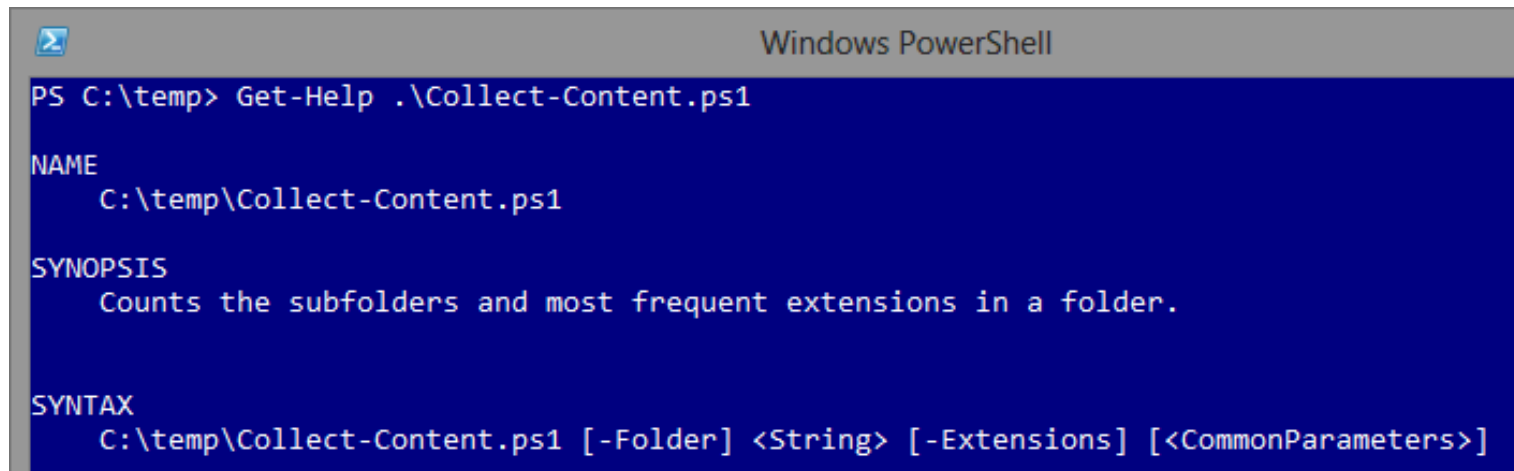
.PARAMETER Folder

The folder to process.

.PARAMETER Extensions

If present, the most frequent extension(s) is listed also.

#>



```
Windows PowerShell
PS C:\temp> Get-Help .\Collect-Content.ps1

NAME
    C:\temp\Collect-Content.ps1

SYNOPSIS
    Counts the subfolders and most frequent extensions in a folder.

SYNTAX
    C:\temp\Collect-Content.ps1 [-Folder] <String> [-Extensions] [<CommonParameters>]
```

HF ellenőrző lista (1)

Menjünk végig a HF ellenőrző listán (lásd honlap)

- **SZ1** Szintaktikai hibás megoldás
 - Futtatjuk még egyszer a tesztekkel: **OK**
- **SZ2** Pontosan specifikált interfész
 - Megegyezik betűről-betűre az elvárt interfésszel:

```
SYNTAX
```

```
C:\temp\Collect-Content.ps1 [-Folder] <String> [-Extensions] [<CommonParameters>]
```

- **SZ3** Ne csak a fejlesztői környezetből fusson
 - Nem az ISE-ből indulva is fut a kód: **OK**
- **SZ4** Használjunk nevesített paramétereket
 - param kulcsszó megoldja: **OK**

HF ellenőrző lista (2)

- **SZ5** Ne tegyen olyat, amit nem kértünk
 - Csak olvas, nem ír/módosít: **OK**
- **SZ6** Ne használjunk bedrótzott neveket
 - Paraméterként kapja a könyvtárat: **OK**
- **SZ7** Ellenőrizzük a bemenetet
 - Könyvtár létezését ellenőrizzük: **OK**
- **SZ8** Adatforgalom, teljesítmény
 - Nincs távoli lekérdezés, de
 - ~11ezer alkönyvtár esetén 20 sec (**elfogadható?**)
- **SZ9** Távoli fél hibájára felkészülés N/A

HF ellenőrző lista (3)

- **SZ10** Legyenek részletes tesztesetek
 - Végén 10 teszteset (bemenet+elvárt kimenet): **OK**

SHOULD

- **SZ20** Hibakezelés
 - Nem létező könyvtár esetén kivétel: **OK**
- **SZ21** Beszédes, differenciált hibaüzenetek
 - (Most csak egy hibaüzenetünk van)
- **SZ22** Kerüljük a kód duplikálását: **OK**
- **SZ23** Legyen fejkomment: **OK**
- **SZ24** Legyen kommentezve: **OK**

HF ellenőrző lista (4)

- **SZ25** Angol változónevek: **OK**
- **SZ26** Tagoljuk a kódot: **OK**
 - Főbb részek üres sorral elkülönítve, if esetén behúzás
- **SZ27** Adott környezet kihasználása
 - Rekurzív bejárást nem implementáltuk újra: **OK**
- **SZ28** Beszédes változónevek
 - \$maximalCount, \$subfolders és nem \$m, \$s: **OK**

Mit kell leadni ebből?

Egy **ZIP** fájl, benne:

```
dist
src
  Collect-Content.ps1
test
  test1
  ...
  ...
  test5
  README.txt / run.ps1
```

Tesztelés leírása:

- tesztek célja
- bemenet
- elvárt kimenet
- kapott eredmény (értékelés)

Összefoglalás

- Paraméterek végiggondolása
 - Szintaxis, kötelezőség
 - Értékek lehetséges osztályai
- Tesztesetek végiggondolása
 - Paraméterosztályok alapján
 - Egyszerű tesztek előre elkészíteni
 - Lásd: <https://inf.mit.bme.hu/content/tesztelesi-alapok>
- HF ellenőrző lista