**Powershell, Powershell ISE, Windows Terminal külön telepíthető**

<http://www.powershellkonyv.hu>

dir, ls

dir /s

Get-Command

Get-Command -CommandType alias

Get-Alias

Get-Alias dir

cmd /c dir c:\

New-Alias -Name wh Value "write-host"

wh ggg

wh “ablak”

wh ‘ablak’

wh “ab`”lak`””

Remove-Item Alias:wh

gin

Get-ComputerInfo

New-Item alma.txt

Get-Item alma.txt

Get-Item alma.txt | Get-Member

(Get-Item alma.txt).BaseName

Add-Content alma.txt “Blabla…”

Get-Content alma.txt (type alma.txt —> Get-Alias type)

(Get-Item alma.txt).Delete()

1+1

2\*5

7-12

$a=5

$a

$b=3

$a\*$b

$a.GetType()

$c.GetType()

$a+$c

$c+$a

$c.ToInt32($null) + $a

Help

Get-Help

Help echo

2023.04.20 - eddig

ISE

valami.ps1

./valami.ps1

Get-Process

Get-Process total\*

(Get-Process total\*).Kill()

Get-Service

Get-Service | Get-Member

| - csővezeték - pipeline

=========================================================

**PSDrive, PSProvider**

PSmeghajtó lett így a fájlrendszeren kívül a registry, a tanúsítványtár, de a függvényeinket és a környezeti és saját változóinkat is ilyen **PSDrive** -okon keresztül is megnézhetjük.

Ezekhez a különböző **PSDrive**-okhoz tartozik egy-egy provider (ami egy .NET alapú program), ami biztosítja az egységes felületet,

Get-PSDrive

dir hkcu:

Get-PSProvider

[enum]::getValues(“System.Management.Automation.Provider.ProviderCapabilities")

New-PSDrive -Name MSReg -PSProvider Registry -Root HKEY\_CURRENT\_USER\Software\Microsoft

Remove-PSDrive MSReg

New-PSDrive -name munka -PSProvider FileSystem -Root 'C:\munka'

(Resolve-Path munka:\szöveg.txt).ProviderPath

notepad (Resolve-Path munka:\szöveg.txt).ProviderPath

New-PSDrive -Name P -PSProvider FileSystem -Scope Global -Root \\dc2016

\PowerShell -Persist

A meghajtó neve csak egybetűs lehet, ami nem ütközhet már meglevő meghajtó betűjelével

A **-Root** és a -Persist csak hálózati megosztás lehet

Csak a **Global** PSDrive-ok lesznek ténylegesen állandóra létrehozva

A felcsatolt meghajtó csak azt létrehozó felhasználó számára lesz látható

New-PSDrive -Name X -PSProvider FileSystem -Scope Global “C:\XXX”

(Hiába Global, nem globál :P, adminként sem)

notepad.exe (Resolve-Path .\alma.tyt).ProviderPath

notepad.exe (get-item .\alma.tyt).FullName

**Változók, konstansok**

**Értékadás**

$valtozoneve = “érték”

$valtozoneve = “újérték”

**Használat**

echo $valtozoneve

$valtozoneve

Write-Host $valtozoneve

Write-Output $valtozoneve

echo “A változó értéke: $valtozoneve”

echo ‘A változó értéke: $valtozoneve’

echo “A változó értéke: $valtozoneve”

echo 'A változó neve: $valtozoneve',"A változó értéke: $valtozoneve"

echo "A változó neve: `$valtozoneve / A változó értéke: $valtozoneve"

echo "`$valtozoneve=$valtozoneve"

$sz=@”

első sör, : ) “ + “

második, ‘

“@

$o=(get-item .\alma.txt)

$o.GetType() → FileInfo

**Érvényesség (Scope)**

Hol érhető el? Csak az adott shellben, addig, amíg be nem zárjuk!

**Egyéb változó kezelő parancsok**

Get-Variable valtozoneve

Get-Variable $valtozoneve — az érték nevű változó adatait adná!

(Get-Variable valtozoneve).Name

(Get-Variable valtozoneve).Value

(Get-Variable valtozoneve).Visibility

(Get-Variable valtozoneve).GetType()

New-Variable -Name x -Value 33

$x

New-Variable -Name xx -Value 33 -Visibility Private → nem elérhető a shellből sem

Set-Variable -Name x -Value 55

$x

$a = get-alias a\*

$a.GetType()

Clear-Variable a

$a

$a.GetType()

Remove-Variable x

$x

$x.GetType()

New-Variable -Name co -Option Constant -Value "Konstans"

$co

$co.Get-Type()

$co=”Új” —> hiba

$\_ (Output és láncolás , ha nincs külön kimenet, akkor alapesetben a teljes objektum átadódik a következő parancsnak, amire $\_ -al tudunk hivatkozni)

"szöveg" | ForEach-Object{$\_.Length}

**Változók jellemzői:**

* név
* érték
* típus (szám, szöveg, object, …)
* hatókör (public, privát)
* láthatóság (locale, global)
* írhatóság (variable,constant, read-only)
* leírás (description)

**Kifejezés / parancs üzemmód**

Kifejezés üzemmódba vált a PowerShell, ha a beírt szöveg számmal (vagy egy pont karaktert követő számmal), idézőjelek közé tett karakterlánccal, vagy $ jellel kezdődik.

Parancs üzemmódba vált a PowerShell, ha a beírt szöveg bármilyen betűvel, a **&** karakterrel, egy pont utáni szóközzel, vagy pont utáni betűvel kezdődik.

A két üzemmód vegyes használatát zárójelezéssel érhetjük el, a zárójelek közötti szövegre minden esetben újra megtörténik az üzemmód meghatározása a fenti szempontok szerint.

2+2

Write-Host 2+2

Write-Host (2+2)

**Formázás**

Alapban is lehet formázva

Get-Service w\*

Format-Table = ft

Format-List = fl

Get-Service w\* | ft

Get-Service w\* | Format-Table Name, Status, CanStop

Get-Service w\* | Format-Table -GroupBy Status

HTML output

Get-Service w\* | ft | ConvertTo-Html -Property Name,Status -Head “Szolgáltatások” > services.html

“a={0}, b={1}, a+b={2}” -f $a, $b, $a+$b 1.4.16 a könyvben

“Összesen: {0, 5} Ft” -f 1200

“Összesen: {0, 5} Ft” -f 10

"Ez most az óra: {0:hh}" -f (get-date)

**Átirányítás: > ( < - linux)**

**Hozzáfűzés: >>**

**Beolvasás**

$a=Read-Host “Írj be egy számot”

**Adattípusok**

Eddig: szám, szöveg, objektum, objektum lista

$a=12 → szám

$b=”alma” → szöveg

$c=Get-Location

$d=Get-Alias a\*

[int] $f=34

$f=”asd” ->hiba

**Számtípusok**

* [int]
* [long]
* [double]
* [decimal]

0x1a -> 26

char - ‘m’

bool (0 v 1) [bool]$bl=0

$bl=10 >5 $bl=(10 >5) → kifejezések eredménye szokott lenni!

$bl=$true

$bl

**Dátum**, **Idő**

Get-Date

Set-Date

$d=[DateTime]::Parse("2009. november 18.")

$d

Get-Date -f yyMMdd\_HHmmss

**Tömbök** - adatsorozatok, az elemek indexelhetőek

$szamtomb=1,2,3,4

$szovegtomb=”egy”,”ketto”,”harom”,”negy”

$vegyestomb=1,”ketto”,3,”negy”

$szamtomb ->külön sorban az elemek

$szamtomb.GetType() -> Object[]

$egyelemu=,”elem”

$urestomb=@()

$szamtomb.Count = .Length

$szamtomb.Rank (dimenzió)

$szamtomb[0] -> első elem

$szamtomb[1..2] -> 2. és 3. elem

1..10 🙂

$szamtomb[-1] -> végétől indul

$b = 1..100

$b[2..4 + 56..58]

$szamtomb += 5 (új elem, de a háttérben új tömbbe másolódik az előző, az eredmény azonos név)

$szamtomb | get-member

gm -InputObject $szamtomb

count, add, clear, indexof, insert, remove, removeat,

**Többdimenziós tömbök**

$tábla = (1,2,3,4),("a","b","c","d") — 2 dim: sorok, oszlopok - mátrix

$tábla[0][0] -> 1

$tábla[1][0] -> a

$igazi = new-object -TypeName 'object[,]' -ArgumentList 3,2

$igazi[2,1]="kakukk"

**Típusos tömbök**

$t = New-Object int[] 20

$ttdt = New-Object 'int[,]' 5,2

$ttdt[0,0] = 12

**Gyűjtemények(collections), listák**

**ArrayList**

$intlist = New-Object System.Collections.ArrayList

$intlist.gettype() -> ArrayList

$intlist.Add(2)

$intlist |getmember

gm -InputObject $intlist

addrange, insertrange, removerange, reverse, sort, capacity

$scal = [system.collections.arraylist] (1,2,3,4,5)

**Sor (Queue)**

$sor = [System.Collections.Queue] @()

Enqueue, Dequeue, Peek, ToArray

**Verem (Stack)**

$verem = [System.Collections.Stack] @()

Push, Pop, Peek, ToArray

**Generic - paraméterezhető típusos tömb**

$genlist = New-Object collections.generic.list[string]

get-member -i $genlist

**Szótárak, hash tábla**

$szotar = @{ Név = "Soós Tibor"; Cím = "Budapest"; "e-mail"="[soostibor@citromail.hu](mailto:soostibor@citromail.hu)"}

$szotar

Összetartozó adatok = rekord, mezőkből áll, akár különböző típusok

Kulcs=Érték párokból áll

Az értékekre a kulcs nevével lehet hivatkozni!

$szotar[“Név”]

$szotar.Keys

$szotar.Values

**Szótártömb**

$névjegyek = New-Object system.collections.arraylist

$névjegyek.Add(@{Név="Soós Tibor";"e‑mail"="soostibor@citromail.hu";Cím="Budapest"})

$névjegyek.Add(@{Név="Fájdalom Csilla";"e‑mail"="fcs@citromail.hu";Cím="Zamárdi"})

$névjegek

**Típuskonverzió**

$a=12

$a

$a.GetType()

$b=[string]$a

$b

$b.GetType()

$c=”46”

$c

$c.GetType()

$d=[int]$c

$d

$d.GetType()

$sz=”Alma”

$cc=[char[]]$sz

$cc

$cc[0]

[int]$cc[0]

**Operátorok**

+, -, \*, /, %, ++, -- ($a=10, $b=$a++, $b, $c=++$a, $c)

+=, -=, \*=, /=

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operátor** | **Leírás** |  | **Kis-nagybetű érzékeny operátor** | **Leírás** |
| **-eq** | egyenlő |  | **-ceq** | egyenlő |
| **-ne** | nem egyenlő |  | **-cne** | nem egyenlő |
| **-gt** | nagyobb |  | **-cgt** | nagyobb |
| **-ge** | nagyobb egyenlő |  | **-cge** | nagyobb egyenlő |
| **-lt** | kisebb |  | **-clt** | kisebb |
| **-le** | kisebb egyenlő |  | **-cle** | kisebb egyenlő |

1 -eq 1

1 -eq 2

“alma” -gt “korte”

“alma” -lt “korte”

Tartalmaz (-contains, -notcontains, -ccontains, -cnotcontains)

1,2,3,4,1,2,3,1,2 -contains 3

Van benne, nincs benne (-in, -notin, -cin, -cnotin)

Dzsóker :) - \*, ?, -like … -notlike …

"ablak" -like "[a-f]lak"

"ablak" -like "[a-f]?lak"

Logikai és bit (-not, -and, -or, -xor, -bnot, -band, -bor, -bxor, -shl, -shr) -not === !

(5 -eq 1) -and $true

(5 -gt 1) -and $true

0 -and 1

1 -and 1

12 -band 5 (1100 and 0101 —> 0100 = 4)

12 -and 5 0-tól eltérő az True

12 -shl 1

12 -shr 1

Típusvizsgálat (-is, -isnot)

“szam” -is [string] - “szam” -is [object] - “szam” -is [array]

Csoportosítás - (), $(), @(), {}, ${}

$a=12, ($a=22)

$($a ="ab"; $b="lak"; $a+$b).length

$tömb = "egy", @($tömb).length

$script = {$dátum = get-date; $dátum}, $script, $script.invoke()

${ilyen név nem lehetne} = 1

Tartomány ( .. )

1..5, 3..-2

Szöveges operátorok (-join, -split, -csplit)

"egy", "kettő", "három" -join "-"

-join ("egy", "kettő", "három")

"egy-kettő-három" -split "-"

'1.2.3.4.5.6.7.8' -split "\.", 3

Parancsok feltételes összefűzése (&& és ||)

Test-Path C:\Windows && Write-Host "Success"

remove-item C:\kakukk || Write-Host "Failure"

**Vezérlési szerkezetek**

**Feltételes végrehajtás (if/then/else, ?:)**

if($a -gt 10) {“Nagyobb, mint 10”}

if($a -gt 10) {“Nagyobb, mint 10”}else{“Kisebb egyenlő mint 10”}

if($a -gt 10) {“Nagyobb, mint 10”} elseif($a -eq 10) {“Pontosan 10”} else {“Kisebb mint 10”}

(Get-Command -CommandType alias | Where-Object {$\_.Name -eq "wh"}).Length

?:

$true ? "Igaz" : "Hamis"

**Esetszétválasztás (switch)**

switch($a){ 1 {"egy"} 2 {"kettő"} 3 {"három"} 10 {"tíz"} default {"egyéb"}}

**Ciklusok (for, foreach, while, do-while, do-until)**

**Menü: 1.7.5**

do

{

**Write-Host** "========================================================="

**Write-Host** "0 - Folyamatok listázása"

**Write-Host** "1 - Szolgáltatások listázása"

**Write-Host** "2 - Folyamat indítása"

**Write-Host** "3 - Folyamat leállítása"

**Write-Host** "9 - Kilépés"

$v = **Read-Host**

**Write-Host** "========================================================="

switch ($v)

{

"0" {**Get-Process**}

"1" {**Get-Service**}

"2" {&(**Read-Host** "Folyamat neve")}

"3" {**Stop-Process** *-name* (&Read-Host "Folyamat neve")}

"9" {break}

default {**Write-Host** "Érvénytelen billentyű!";break }

}

} while ($v -ne "9")

**Véletlenszám gen**

Get-Random

Get-Random 100

Get-Random -Minimum 50 -Maximum 100

5..15 | Get-Random -Count 3

"Aladár", "Bence", "Csaba", "Dénes", "Ferenc", "Hugó", "István" | Get-Random -Count 3

**Függvények**

function függvény1 {"első"}

függvény1

—-------------------------------------------

function körterület ($sugár)

{

$sugár\*$sugár\*[math]::Pi

}

körterület 12

—-------------------------------------------

function üdvözlet ($név = "Tibi") {“üdv $nev” }

üdvözlet

üdvözlet(“Pisti”)

—-------------------------------------------

function körterület ([double] $sugár)

{

$sugár\*$sugár\*[math]::Pi

}

—-------------------------------------------

$a=1

"a=$a"

function fv1($par=2)

{

"fv1 - a=$a"

"fv1 - par=$par"

$b=$par+1

"fv1 - b=$b"

}

fv1

"b=$b"

"par=$par"

fv1(4)

"b=$b"

"par=$par"

—-------------------------------------------

function fv1($p1, $p2=3)

{

"p1=$p1"

"p2=$p2"

}

fv1

fv1(4)

fv1 5 6

—-------------------------------------------

function fv2 ($par1, [ref]$par2)

{

$par1++

$par2.Value++

"par1=$par1"

Write-Host("par2={0}" -f $par2.Value)

}

$a=10

$b=20

fv2 $a ([ref]$b)

"a=$a"

"b=$b"

—-------------------------------------------

function fv5 ($par1, $par2)

{

return ($par1+$par2)

}

$c=fv5 11 22

$c

$c.gettype()

**Kiírás fájlba**

10..20 > adatok.txt

30..50 >> adatok.txt

$file="adatok.txt"

$sorok="Első sor", "Második sor"

Set-Content -Path $file -Value $sorok

Add-Content -Path $file -Value "Harmadik sor"

Get-ChildItem | Out-File c:\dir.txt #1.8.13

**Beolvasás fájlból**

$file="adatok.txt"

$sorok=Get-Content $file

$sorok

$sorok.GetType()

$sorok[1]

$sorok[1].GetType()

**Táblázatok kezelése (import, export, convertfrom, convertto)**

Excel 2016 export CSV - Adatok.csv

Get-Content Adatok.csv

Import-Csv .\Adatok.csv -Encoding Default

Get-Content .\Adatok.csv | ConvertFrom-Csv -Delimiter ";"

Import-Csv .\Adatok.csv -Encoding Default -Delimiter ";"

Get-Process -Name A\* > proclist.txt

Get-Process -Name A\* | ConvertTo-Csv -Delimiter “;” > proclist.csv

Get-Process -Name A\* | Export-Csv -Delimiter ";" proclist2.csv

**Változók láthatósága 1.6.4**

get-variable a -scope 0 #0 -local, 1-szülő, 2-nagyszülő

$a=”valami”

function fv3{$a="másvalami";$a; (get-variable a -scope 1).value; set-variable a -scope 1 -value "változás"}

$c=1

function kie{$c=2; "Én c-m:$local:c";"Te c-d:$global:c"}

kie

**Hibakezelés 2.4**

terminating, nonterminating error

"Eleje"; 15/0; "Vége"

"Eleje"; throw "kakukk"; "Vége"

"Eleje"; get-content blabla.txt -ErrorAction SilentlyContinue; "Vége"

$error[0]

function deletefile ($path){

try {remove-item -Path $path -ErrorAction Stop}

catch {

"Nincs ilyen fájl!"

return -1

}

finally {"Itt a vég!"}

"Ez a normális vég"

}

Remove-Item nemlétezőfájl.txt 2> $null

filter bulk-remove

{

if (-not (Test-Path $\_))

{ Write-Warning "Nincs ilyen file: $\_!" }

else

{ Remove-Item $\_ }

}

"nincs.txt", "file.txt" | bulk-remove

filter bulk-remove

{

**remove-item** $\_ *-ErrorAction* silentlycontinue

if (!$?)

{ $Error[0].categoryinfo.reason}

}

**Futtatás háttérben és távol 1.11**

Get-Command -noun job

Get-Job, Start-Job, Stop-Job, Wait-Job, Receive-Job, Remove-Job, Suspend-Job

start-job -ScriptBlock { for($i=0; $i -lt 60; $i++){$i; Start-Sleep -Seconds 1}}

Enable-PSRemoting

Enable-PSRemoting -force

New-PSSession -ComputerName dc, member

New-PSSession -ComputerName server -Credential VZ\Administrator

New-PSSession -ComputerName 192.168.56.200 -Credential VZ\Administrator

Invoke-Command -Session (Get-PSSession) -ScriptBlock {get-service g\*}

Enter-PSSession 1

Exit-PSSession

Disconnect-PSSession 2

Remove-PSSession 2

**Környezeti változók (env: PSDrive) 2.1.5**

CMD-ben: echo %USERNAME%

cd env:

dir

$env:USERNAME

(Get-Item USERNAME).Value

**Szűrők 1.8**

**ForEach-Object 1.5.5**

1,2,3 | ForEach-Object {$\_ \* 3}

1,2,3 | ForEach-Object {“-eleje-”} {$\_ \* 3} { “-vége-”}

##get-date; 10..1|ForEach-Object -Parallel {Start-Sleep $\_; Write-Host $\_}; Get-Date

#7.0-tól van csak -Paralell

**Where-Object 1.5.6**

$tömb = 1,2,"szöveg", 3

$tömb | Where-Object {$\_ -is [int]} | ForEach-Object {$\_\*2}

Get-Process | Where-Object -Property Ws -GE 130mb

Get-Process | Where WS -ge 130mb

**Sort-Object 1.8.3**

3,32,1,6,24 | Sort-Object

3,32,1,6,24 | Sort-Object -Descending

get-process | sort-object -property workingset -Descending

Get-ChildItem C:\Windows -File | Sort-Object -Property extension, name

Get-ChildItem . | Sort-Object -Property {$\_.name.length}

**Select-Object 1.8.5**

**Format-Table, Format-Wide 1.8.6**

**Out-String 1.8.16**

**Group-Object 1.8.4**

**Compare-Object 1.8.7**

**Measure-Object 1.8.8**

**Tee-Object 1.8.9**

**Server Core-n nincs ISE, nem is telepíthető!!!**

**De van lehetőség Remote ISE-re!**

<http://mctexpert.blogspot.com/2015/08/can-you-add-powershell-ise-to-server.html>

De tűzfal nem engedi távolról, csak helyi hálón!

“ Alapértelmezés szerint a Rendszerfelügyeleti webszolgáltatások tűzfalkivétele a

nyilvános profilok esetében ugyanazon helyi alhálózaton található távoli számítógépekre korlátozza a hozzáférést.”

<https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_remote_troubleshooting?view=powershell-7.3>

<https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7.3>

AWS-en engedni az 5985/TCP portot

Enable-PSRemoting

Enable-PSRemoting -Force

Set-NetFirewallRule -Name "WINRM-HTTP-In-TCP-PUBLIC" -RemoteAddress Any

2019-en intellisence is megy, a 2022-n is megy! 🙁 Várni kell, lassan éled! Illetve kapcsolgatni kell! Picit lassúúúúúúúúúúúúúúú 🙁

**Rendszerkezelési lehetőségek**

**Számítógép kezelés 2.10**

Get-Command \*-computer

Restart-Computer / Restart-Computer -Force

Stop-Computer / Stop-Computer -ComputerName member (Távoli gép leállítása)

Rename-Computer -NewName azújnév -Force

Add-Computer -WorkGroupName WG (Munkacsoport beáll.)

$cred = new-object -typename System.Management.Automation.PSCredential -argumentlist r2\administrator", ( Convertto-SecureString 'Password123' -force -asplaintext) (Tartományba léptetés)

Test-Connection member / Test-Connection member -Quiet (Kapcsolat ell.)

Test-ComputerSecureChannel -Server member (Tartományi komm. ell.)

**Processzek kezelése 2.15**

Get-Command \*-process

Start-Process notepad.exe

Start-Process notepad.exe -ArgumentList C:\fájl.txt -PassThru

**A -PassThru paraméterrel visszakapjuk az elindított processz adatait.**

stop-process -name "notepad"

Invoke-Command -ComputerName member -ScriptBlock {start-process notepad} (Távoli gépen háttérben)

(Get-WmiObject -Class Win32\_process -Filter "Name = 'notepad.exe'").getowner().user (Ki indította)

(Get-Process notepad).CloseMainWindow() (Szelíd leállás)

(Get-Process notepad).Kill() (Get-Process notepad).Close() (Erőszakos leállítás)

whoami /priv

Start-Process -FilePath notepad -Verb runas (Futtatás emelt jogokkal - -Verb)

**Szolgáltatások kezelése 2.15.3**

Get-Command \*-service

get-service | where-object {$\_.CanStop}

get-service | where-object {$\_.CanStop} | Stop-Service

get-service audiosrv | get-member -MemberType properties

$service=[WMI] "Win32\_Service.Name=`"Alerter`"" - hiba

$service.StartMode

get-help Set-Service

Set-Service -Name "SERVICE-NAME" -Status stopped -StartupType disabled

New-Service -Name "YourServiceName" -BinaryPathName <yourproject>.exe

Szerveren

Install-WindowsFeature …

Remove-WindowsFeature …

**Registry 2.8**

New-PSDrive -Name Uninstall -PSProvider Registry -Root HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

cd Uninstall:

Get-ItemProperty 7-Zip

New-ItemProperty . -Name DumpsterAlwaysOn -Value 1 -type DWORD

Set-ItemProperty . -Name DumpsterAlwaysOn -Value 0

New-Item Valami

**Fájlkezelés 2.6**

Get-Location

(Get-Location).GetType() — PathInfo

(Get-Location).ToString()

pwd

Get-Alias pwd -> Get-Location

Get-Item .

Get-Item ..

(Get-Item .).GetType() — DirectoryInfo

(Get-Item .).GetType() | gm

(Get-Item .).Name

(Get-Item .).FullName

Get-Alias dir -> Get-ChildItem

Get-ChildItem

Get-ChildItem -Force (Rejtetteket is)

Get-ChildItem -Directory

Get-ChildItem -File

Get-ChildItem -Attributes “h”

Get-ChildItem -Filter \*.txt

Get-ChildItem C:\PSKönyv -Exclude \*.txt -Recurse -Attributes !directory

New-Item -Path . -Name szöveg.txt -type file -Value "Ez egy szöveg"

New-Item -Path . -Name "Alkönyvtár" -type Directory

New-Item uresfile

New-Item -ItemType Directory ujmappa

Test-Path ujmappa

Test-Path uresfile

Test-Path C:\scripts\ -OlderThan 2016.01.01

(Get-Item C:\scripts\alice.txt).attributes = "Archive, hidden"

[enum]::GetValues("System.IO.FileAttributes")

Copy-Item C:\PowerShell\20150923\\* -Destination C:\PowerShell\másolat\

Get-Command \*-Item

Move-Item

Rename-Item

Remove-Item

**Jogosultságok kezelése - ACL 2.6.11**

Get-Command \*-acl

Get-Acl

Get-Acl | fl

Set-Acl —> FELEJTŐS!!!! Megoldás icacls!

icacls . #nem powershell!

**Aktuális jogok:**

icacls mappa

**Öröklés tiltása:**

icacls mappa /inheritancelevel:d

**Users csoport törlése:**

icaclsmappa/remove:g:d Users /t

**Subfolder and files only Full:**

icacls mappa /grant "csoport:(OI)(CI)(IO)(F)"

**This folder RX,W:**

icacls mappa /grant "csoport:(RX,W)"

**Csak olvasási jog**

icacls mappa /grant "masikcsoport:(OI)(CI)(RX)

**Írási jog**

icacls mappa /grant "harmadikcsoport:(OI)(CI)(RX,W)

**Lokális felhasználók és csoportok 2.22**

Get-Command \*-Local\*

Get-LocalUser

Get-LocalGroup

New-LocalUser -Name "felhasználóneve"-Password (ConvertTo-SecureString -AsPlainText `  
"Password123" -Force) `

-PasswordNeverExpires `

-AccountNeverExpires `

-UserMayNotChangePassword

New-LocalGroup “Újcsoport”

Add-LocalGroupMember -Group "Újcsoport" -Member "felhaszn1", “felhaszn2”

Get-LocalGroupMember -Group "Újcsoport"

**Megosztások**

Get-ChildItem \\asus\powershell

Get-Command \*share\* Get-Command \*-Smb\*

Get-SmbShare

New-SmbShare -Name "megosztásneve" -Path "mappa" -FullAccess "Everyone"

New-SmbMapping -LocalPath 'S:' -RemotePath '\\ServerCoreGyak\12A'  
New-SmbMapping -LocalPath 'K:' -RemotePath '\\ServerCoreGyak\Kozos' -Persistent

Get-SmbMapping

net #nem powershell!

net share #nem powershell!

net

**Hálózati beállítások**

Get-NetAdapter

Get-NetIPInterface

Get-NetIPAddress

Get-NetIPInterface -AddressFamily IPv4

**Set-NetIPInterface -InterfaceIndex 6 -Dhcp Disabled**

New-NetIPAddress -InterfaceIndex 6 -AddressFamily IPv4 -IPAddress "192.168.0.110" -PrefixLength 24 -DefaultGateway "192.168.0.254"

Set-DnsClientServerAddress -InterfaceIndex 6 -ServerAddresses "8.8.8.8" -PassThru

Set-DnsClient -ConnectionSpecificSuffix vz.lan -InterfaceIndex 14 -RegisterThisConnectionsAddress:$true -UseSuffixWhenRegistering:$true -Verbose

Remove-NetIPAddress -Confirm -DefaultGateway 192.168.111.254 -IPAddress 192.168.111.200 -PassThru

Get-NetAdapterBinding | Where-Object ComponentID -EQ 'ms\_tcpip6'

Disable-NetAdapterBinding -Name 'Ethernet' -ComponentID 'ms\_tcpip6' #IPv6 tiltás

ipconfig /all #nem powershell!

**Tűzfal**

Get-Command \*-NetFirewall\*

Get-NetFirewallRule

Get-NetFirewallRule -DisplayName "\*ICMPv4\*"

Get-NetFirewallRule -Name "FPS-ICMP4-ERQ-In"

Enable-NetFirewallRule -Name "FPS-ICMP4-ERQ-In" #ping engedélyezés

New-NetFirewallRule -Name 'ICMPv4' -DisplayName 'ICMPv4' -Description 'Allow ICMPv4' -Profile Any -Direction Inbound -Action Allow -Protocol ICMPv4 -Program Any -LocalAddress Any -RemoteAddress Any

Enable-NetFirewallRule -Name 'ICMPv4'

**DHCP**

Install-WindowsFeature DHCP –IncludeManagementTools  
Restart-Computer –Force   
Get-Command \*-DhcpServer\*

Add-DhcpServerv4Scope -Name "Internal Network" `  
-StartRange 192.168.20.71 -EndRange 192.168.20.90 `  
-SubnetMask 255.255.255.0 `  
-LeaseDuration 0.08:00:00 `  
-State Active `

-PassThru

Set-DhcpServerv4OptionValue -ScopeId 192.168.20.0 `  
-DnsServer 192.168.20.1,192.168.20.2 `  
-Router 192.168.20.254 `  
-DnsDomain 'test.com'

vagy

Set-DhcpServerv4OptionValue -ScopeId 10.11.12.0 -DnsServer 10.11.12.201 -DnsDomain vz.lan

Set-DhcpServerv4Reservation -IPAddress 192.168.111.10 -ClientId 08-00-27-F1-73-59 -Name W10

vagy

Add-DhcpServerv4Reservation -IPAddress 192.168.111.10 -ClientId 08-00-27-F1-73-59 -Name W10

Remove-DhcpServerv4Reservation ...

Restart-Service DHCPServer

Get-DhcpServerv4Scope

Get-DhcpServerv4Reservation 192.168.111.0 //a 192.168.111.0 a scope id

Get-DhcpServerv4Binding

Get-DhcpServerv4DnsSetting

Get-DhcpServerv4Lease 192.168.111.0

**DNS**

Install-WindowsFeature DNS –IncludeManagementTools  
Restart-Computer –Force

Get-Command \*-DnsServer\*

Forward zóna:

Add-DnsServerPrimaryZone -Name "tartománynév" -ZoneFile "tartománynév.dns" -DynamicUpdate None –PassThru

Get-DnsServerZone

Reverse zóna:

Add-DnsServerPrimaryZone -NetworkID 192.168.111.0/24 -ZoneFile "111.168.192.in-addr.arpa.dns" -DynamicUpdate None -PassThru

Remove-DnsServerZone "tartománynév.nev" –PassThru

Get-DnsServerZone

Add-DnsServerResourceRecordA -Name "rx-8" -ZoneName "srv.world" -IPv4Address "10.0.0.110" -TimeToLive 01:00:00 -CreatePtr -PassThru

Add-DnsServerResourceRecordCName -Name "fd3s" -HostNameAlias "rx-7.srv.world" -ZoneName "srv.world" -PassThru

Get-DnsServerResourceRecord -ZoneName "srv.world" | Format-Table -AutoSize -Wrap

Add-DnsServerForwarder -IPAddress 10.0.0.10 -PassThru

**Web**

Install-WindowsFeature Web-Server -IncludeManagementTools

Get-Command \*-WebSite\*

Stop-Website 'Default Web site'

New-WebSite -Name "NewSite" -Port 20080 -HostHeader "RX-8.srv.world" -PhysicalPath "C:\inetpub\newsite"

inkább

New-WebSite -Name "NewSite" -Port 20080 -PhysicalPath "C:\inetpub\newsite"

**NAT**

Rename-NetAdapter -Name "Ethernet" -NewName "Belso"

Install-WindowsFeature Routing -IncludeManagementTools  
Install-RemoteAccess -VpnType RoutingOnly

netsh routing ip nat install  
netsh routing ip nat add interface InterfaceNeve (Ez itt a belső)  
netsh routing ip nat add interface ‘InterfaceNeve’ mode=full

**AD 2.19**

Install-WindowsFeature -name AD-Domain-Services –IncludeManagementTools

Restart-Computer –Force

Install-ADDSForest -DomainName "domain.neved " -ForestMode WinThreshold -DomainMode WinThreshold -DomainNetbiosName FD3S01 -SafeModeAdministratorPassword (ConvertTo-SecureString -AsPlainText "Password123" -Force) -InstallDNS

Get-ADDomain -Server SzerverNeve

Get-ADDomainController -Server SzerverNeve

Get-ADForest -Server SzerverNeve

Get-ADUser -Filter \*

Get-ADGroup -Filter \*

Get-ADOrganizationalUnit -Filter \*

New-ADOrganizationalUnit -Name valami

New-ADOrganizationalUnit -Name alvalami -Path OU=VALAMI,DC=XY,DC=LOCAL"

New-ADGroup -GroupScope Global -Name csoport -Path "OU=ALVALAMI,OU=VALAMI,DC=XY,DC=LOCAL"

New-ADUser -Name valaki -AccountPassword (ConvertTo-SecureString -AsPlainText "Password123" -Force) -Path "OU=ALVALAMI,OU=VALAMI,DC=XY,DC=LOCAL"

Add-ADGroupMember -Identity csoport -Members valaki

Felhasználók .csv fájlból: 2.19.3

**GPO 2.23**

<https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/ee461027(v=technet.10)?redirectedfrom=MSDN>

**Windows Update**

Install-Module PSWindowsUpdate

Get-WindowsUpdate

Install-WindowsUpdate

Get-WindowsUpdate -AcceptAll -Install -AutoReboot

**Azure**

Tulajdonos weboldalon mindent csinálhat, de powershellben nem.

Weben új felhasználó felvétele, szerepkörét tekintve Globális rendszergazda -> weben dolgozhat.

Azure előfizetésnél (Subscription) a felvett felhasználót classic administrátorhoz is hozzá kell tenni, ekkor tud erőforrásokkal dolgozni és ekkor tudja a get-azsubscription-al lekérdezni az előfizetési infókat!

Get-AzSubscription

Set-AzContext -Subscription "Azure-előfizetés 1"

Nem kizárt, hogy ezt csak akkor kell elvégezni, ha az ingyenes hozzáférésről váltottunk fizetősre!

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

Install-Module -Name Az -Scope CurrentUser -Repository PSGallery -Force (Online és Adminként)

Get-Module

Uninstall-Module -Name Az

Connect-AzAccount (Online - böngésző)

Disconnect-AzAccount (egymás után kétszer kell futtatni - csak másodikra megy)

Get-AzContext

Get-AzTenant

$tenantId = (Get-AzContext).Tenant.Id

Get-AzADUser

<https://learn.microsoft.com/en-us/powershell/module/az.resources/get-azaduser?view=azps-9.3.0>

Get-AzADUser -DisplayName "Walaki" |fl

Get-AzADGroup

New-AzADUser -DisplayName "Új felhasználó" -Password (ConvertTo-SecureString -AsPlainText -Force "Password123") -AccountEnabled $true -MailNickname "ujfelh" -UserPrincipalName "ujfelh@gwalakigmail.onmicrosoft.com"

Update-AzADUser -UPNOrObjectId $upn -City $city

New-AzADGroup -DisplayName "Új csoport" -MailNickname "ujcsop"

Update-AzADGroup -ObjectId $objectid -Description $description

Add-AzADGroupMember -TargetGroupObjectId (Get-AzADGroup -DisplayName "Új csoport").Id -MemberObjectId (Get-AzADUser -DisplayName "Új felhasználó").Id

Get-AzADGroupMember -GroupDisplayName "Új csoport"

Remove-AzADGroupMember -GroupDisplayName "Új csoport" -MemberObjectId (Get-AzADUser -DisplayName "Új felhasználó").Id

Remove-AzADGroup -DisplayName "Új csoport"

Get-AzResourceGroup

<https://learn.microsoft.com/en-us/powershell/module/az.resources/get-azresourcegroup?view=azps-9.3.0>

Get-AzLocation

New-AzResourceGroup -Name "ujfelh-rg" -Location "North Europe"

Remove-AzResourceGroup -Name "ujfelh-rg" -Force

<https://learn.microsoft.com/en-us/azure/storage/common/storage-account-create?toc=%2Fpowershell%2Fazure%2Ftoc.json&tabs=azure-powershell&view=azps-9.3.0>

New-AzStorageAccount -ResourceGroupName "ujfelh-rg" -Name "ujfelhsa" -Location "North Europe" -SkuName Standard\_RAGRS -Kind StorageV2

Remove-AzStorageAccount -Name "ujfelhsa" -ResourceGroupName "ujfelh-rg" -Force

[Get-AzStorageAccount](https://learn.microsoft.com/en-us/powershell/module/az.storage/get-azstorageaccount?view=azps-9.3.0)

<https://learn.microsoft.com/en-us/azure/storage/blobs/blob-containers-powershell>

$ctx = New-AzStorageContext -StorageAccountName <storage account name> -UseConnectedAccount

New-AzStorageContainer -Name $containerName -Context $ctx

Get-AzStorageContainer -Name $containerName

…

Remove-AzStorageContainer -Name $containerName -Context $ctx

<https://learn.microsoft.com/en-us/azure/storage/blobs/blob-powershell>

$path = "C:\temp\"

$containerName = "myContainer"

$filename = "demo-file.txt"

$imageFiles = $path + "\*.png"

$file = $path + $filename

Set-AzStorageBlobContent -File $file -Container $containerName -Context $ctx

Get-AzStorageBlob -Container $namedContainer -Context $ctx

Get-AzStorageBlobContent -Container $containerName -Blob $blobName -Destination $path -Context $ctx

Copy-AzStorageBlob -SrcContainer "photos" `

-SrcBlob $blobname -DestContainer "archive" `

-DestBlob $("photos/$blobname") -Context $ctx

Create VM

<https://learn.microsoft.com/en-us/powershell/azure/azureps-vm-tutorial?view=azps-9.3.0>

$vmParams = @{

ResourceGroupName = 'TutorialResources'

Name = 'TutorialVM1'

Location = 'eastus'

ImageName = 'Win2016Datacenter'

PublicIpAddressName = 'tutorialPublicIp'

Credential = $cred

OpenPorts = 3389

Size = 'Standard\_D2s\_v3'

}

$newVM1 = New-AzVM @vmParams

Get-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"

Stop-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"

Start-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"

Restart-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"

Remove-AzVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"