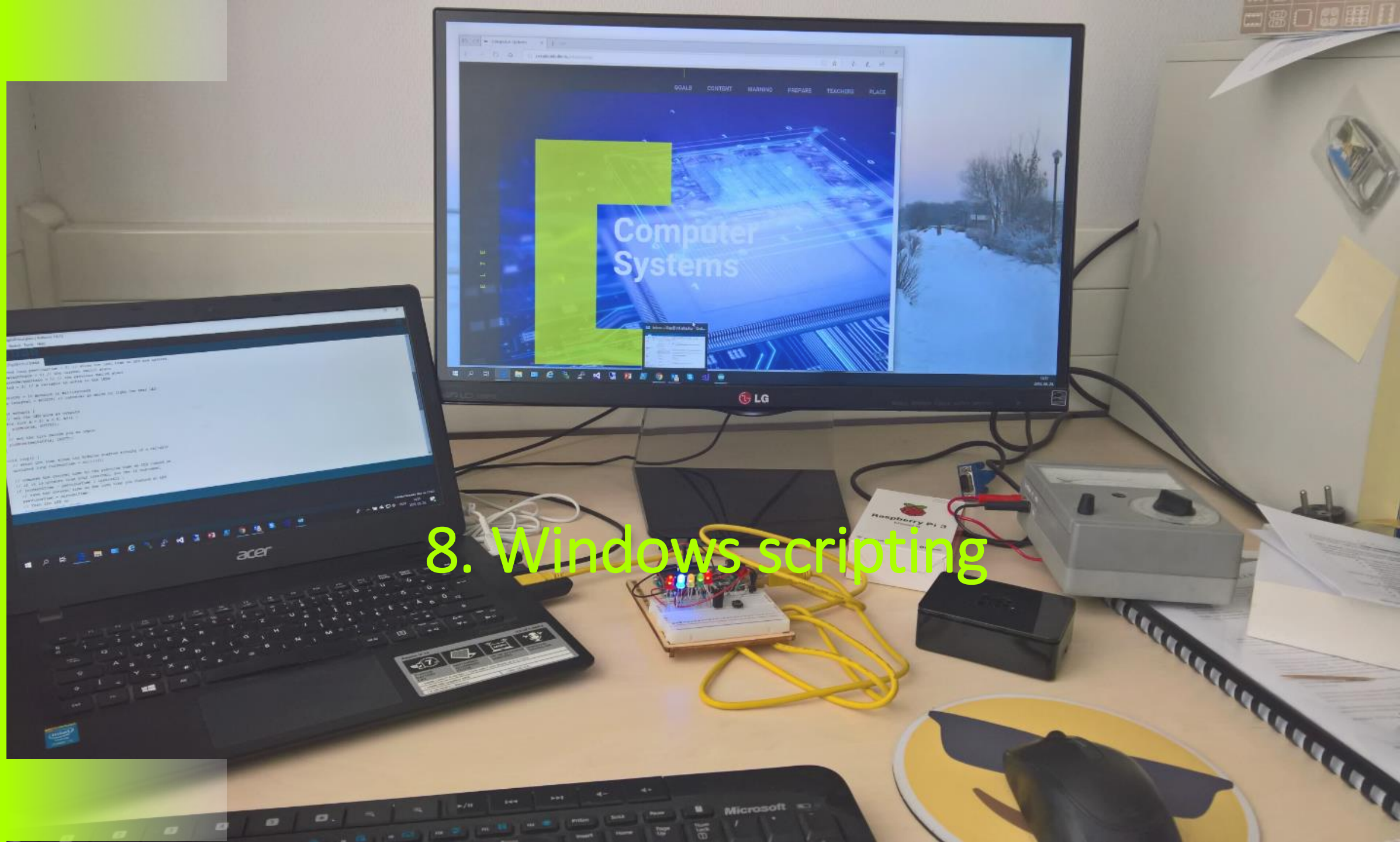


## 8. Windows scripting



# Visszatekintés

- Számítógépek, számábrázolás, kódolás, felépítés, fájlrendszerek
- Alapvető parancsok, folyamatok előtérben, háttérben
- I/O átirányítás, szűrők, reguláris kifejezések
- Változó, parancs behelyettesítés, aritmetikai, logikai kifejezések
- Script vezérlési szerkezetek, Sed, AWK
- Hálózati szolgáltatások

# Mi jön ma?

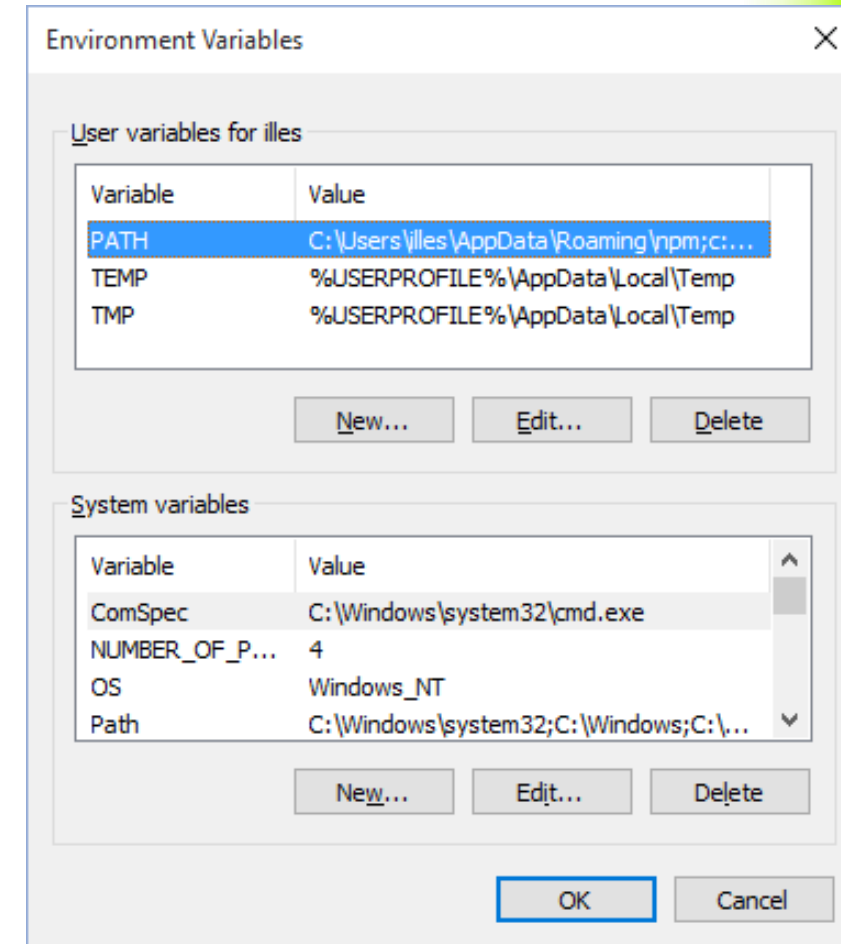
- Windows kiszolgáló
  - Szolgáltatások
- Műveletek automatizálása
  - Batch parancsok
  - Batch hiányosságok
- Windows Script Host
- PowerShell

# Windows operációs rendszer

- Múlt: IBM PC DOS – Disk Operating System
  - 86-DOS – egy CP/M klón, Seattle Computer Products
  - A Microsoft ezt megvette 1981-ben.
  - Átnevezéssel: MS-DOS , 1981
- 1984 – Microsoft Windows mint az MS-DOS grafikus kiterjesztése.
- 1993 – 32 bites rendszer, Windows NT
- 1995 – Windows 95, Windows NT 4.0
- Jelen: 2015-2017 – Windows 10

# Windows kezelőfelület

- Grafikus felület, parancsok indítása – egy ikonra kattintás
  - Start menüpont, task bar, desktop
- Parancssor ablak (cmd) létezik
  - Minden parancs, installált program parancssorból is indítható!
- Path környezeti változó 2 részből áll!
  - User path + system path



# Van script készítési lehetőség Windows rendszerben?

- Igen!
- Elsősorban adminisztráció megkönnyítésére
  - Hasonlóan a UNIX, Linux alatti lehetőséghez!
- Többféle script létezik MS Windows!
  - Batch program (alapok)
  - Windows Script(ing) Host (VB Script vagy JScript alapú)
  - PowerShell

# Batch program alapok

- Az MS DOS rendszer részeként jelent meg!
- Fő feladata: Parancsok összegyűjtése és indítása egy parancs a ,batch' parancs segítségével.
- Ma is használt, bár ritkán!
- Szöveges parancsok
- Fájlnev kiterjesztés: .bat
- Megjegyzés: rem
- echo utasítás
- call masik.bat

# Batch változók, paraméterek

- Kis-nagybetű azonos!
- Változó definiálás: `set a=5`
- Összes változó kiírása: `set`
- Változó értéke: `%név%`, Pl: `%a%`
- `path` parancs, `%path%` változó
- `prompt` utasítás
- `%1`, `%9` a batch paraméterei
- `%0` a batch program neve
- `shift` parancs, balra tolja a paramétereket



# Batch vezérlési szerkezetek

- Címke definiálás: :cimke1
- Ugrás egy címkére: goto cimke1
- Elágazás: if [not] feltétel utasítás
  - if errorlevel 5 goto ot
    - Igaz, ha az előző parancs visszatérési értéke nagyobb vagy egyenlő mint 5!!
  - if %a%=="5" goto 5
  - if exist fájlnev goto hat
- Ciklus: for %%változó in (lista) do (utasítás)

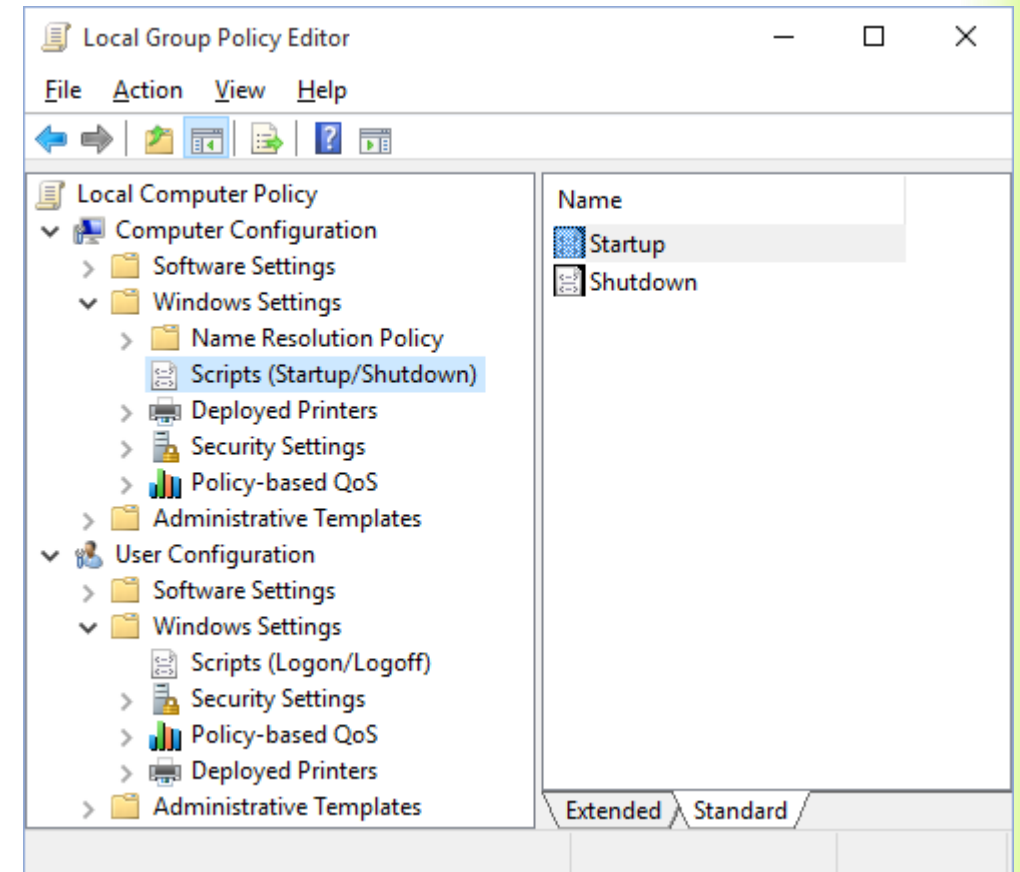
# Batch ciklus példa

- Hasonlít a shell script for ciklusához
- XP-ben /f kapcsoló

```
rem for ciklus példa
rem ne írja ki a parancsokat
@echo off
rem %%név formát kell használni!!!
for %%i in (alma korte) do (
    echo %%i
)
rem egy file sorainak első szavait vegyük
for /f %%j in (alma.txt) do (
    echo %%j
)
```

# autoexec.bat vs. GPEDIT.MSC

- Feladata: belépéskor automatikusan lefutó parancsok (batch) gyűjteménye!
- Mára helyét átvette a csoportos házirend készítés lehetősége!
  - Windows startup/shutdown
  - User startup/shutdown

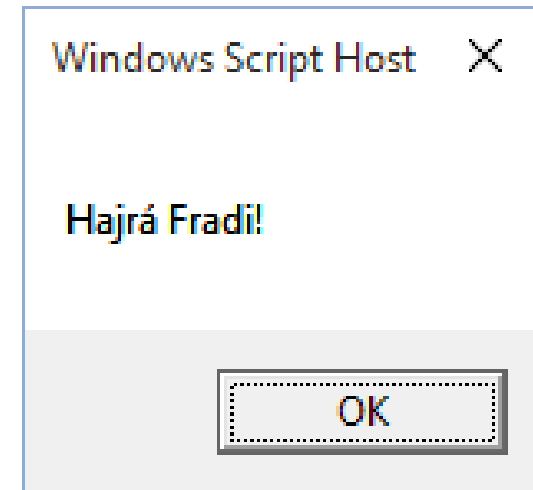
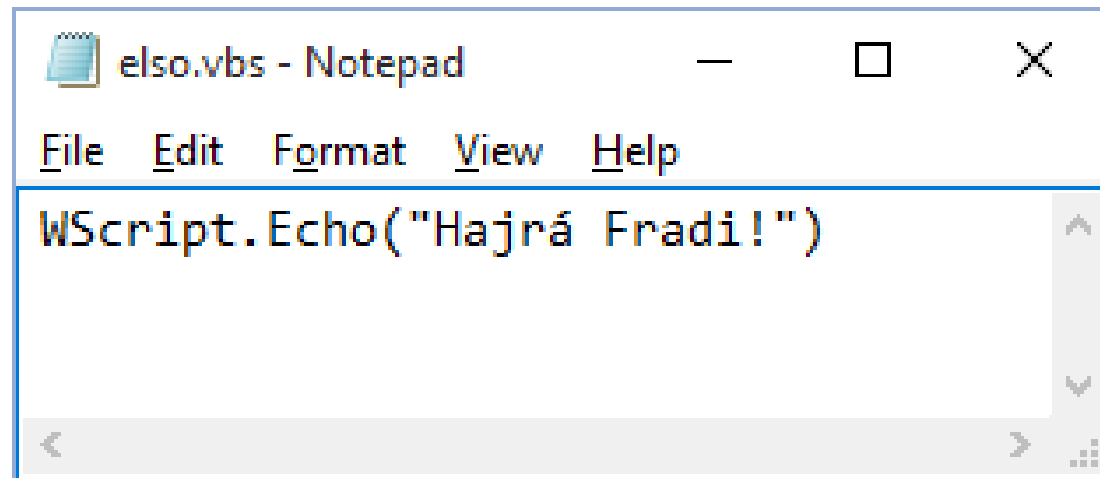


# Windows Script Host - VBS

- Windows 2000 szerverben jelenik meg– új script nyelv
  - Ami nem új, hiszen az MS Office 97-ben jelent meg mint új makró nyelv!
  - Általános script környezet (Visual Basic Script, JScript bár ez ritkán használt)
  - A mai Office környezet makró nyelve ugyanez!
- .vbs kiterjesztés (.js is lehet, Jscript)
- Leírás: MSDN dokumentáció
  - [https://msdn.microsoft.com/en-us/library/9bbdkx3k\(v=vs.84\).aspx](https://msdn.microsoft.com/en-us/library/9bbdkx3k(v=vs.84).aspx)

# WSH példa

- Készíteni kell egy .vbs vagy .js kiterjesztésű parancsállományt!
- A WSH objektumelvű, Wscript a főobjektum!
  - Echo metódus
- Futtatás: Parancssorból vagy egér kattintás



# Mi van a WSH után?

- Miközben a WSH ma is használt script nyelv, a .NET Framework megjelenése után új script eszköz jelenik meg!
- .NET Framework 2.0, 2005
- PowerShell

# PowerShell

- A Microsoft új generációs script nyelve
  - Batch, VBS, WSH utód
- Ingyenes, utólag kellett installálni XP, Vista alá, a WS 2008, 2012, Win7, Win8, Win10 része
- Letölthető: <http://www.microsoft.com/> letöltések oldaláról. Ingyenes!
- Jelenlegi verzió: 3.0 (Windows 8), 4.0 (Win 8.1), 5.0 (Win10)
  - Windows 7 alatt a Powershell 2.0 érhető el, de frissíthető!
  - Windows Management Framework 4 is elérhető Win7 alá!
  - Get-PSSnapin – Megadja a PS verziót!

# Powershell elérhetősége - Install

- .NET Framework 2.0 szükséges
- Start- Programok- Windows Powershell programcsoport
- Windows XP alá kell külön installálni!
  - PowerShell 1.0 Documentation Pack
- Dokumentáció: online help
- A parancs egy cmd (command.com) –hoz hasonló karakteres ablakként jelenik meg, vagy grafikus felületen (ISE).

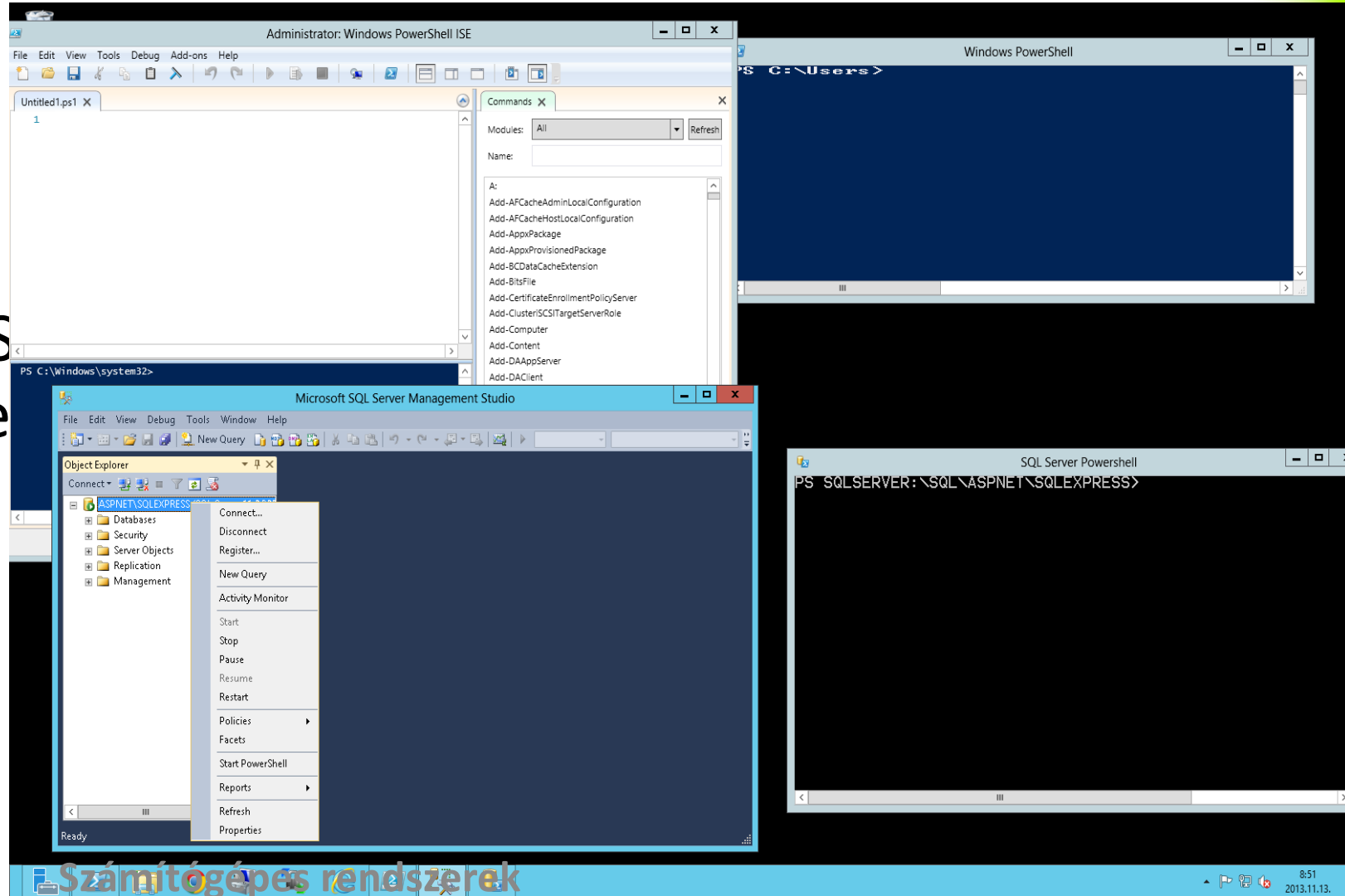


# Mire jó a PowerShell?

- Mint a shell script!
- Elsősorban menedzsment célra
  - Első hivatalos MS tanfolyam: Course 6434A (2008)
  - Automating Windows Server® 2008 Administration with Windows PowerShell
  - Általános script programozási környezet Windows operációs rendszer alatt.

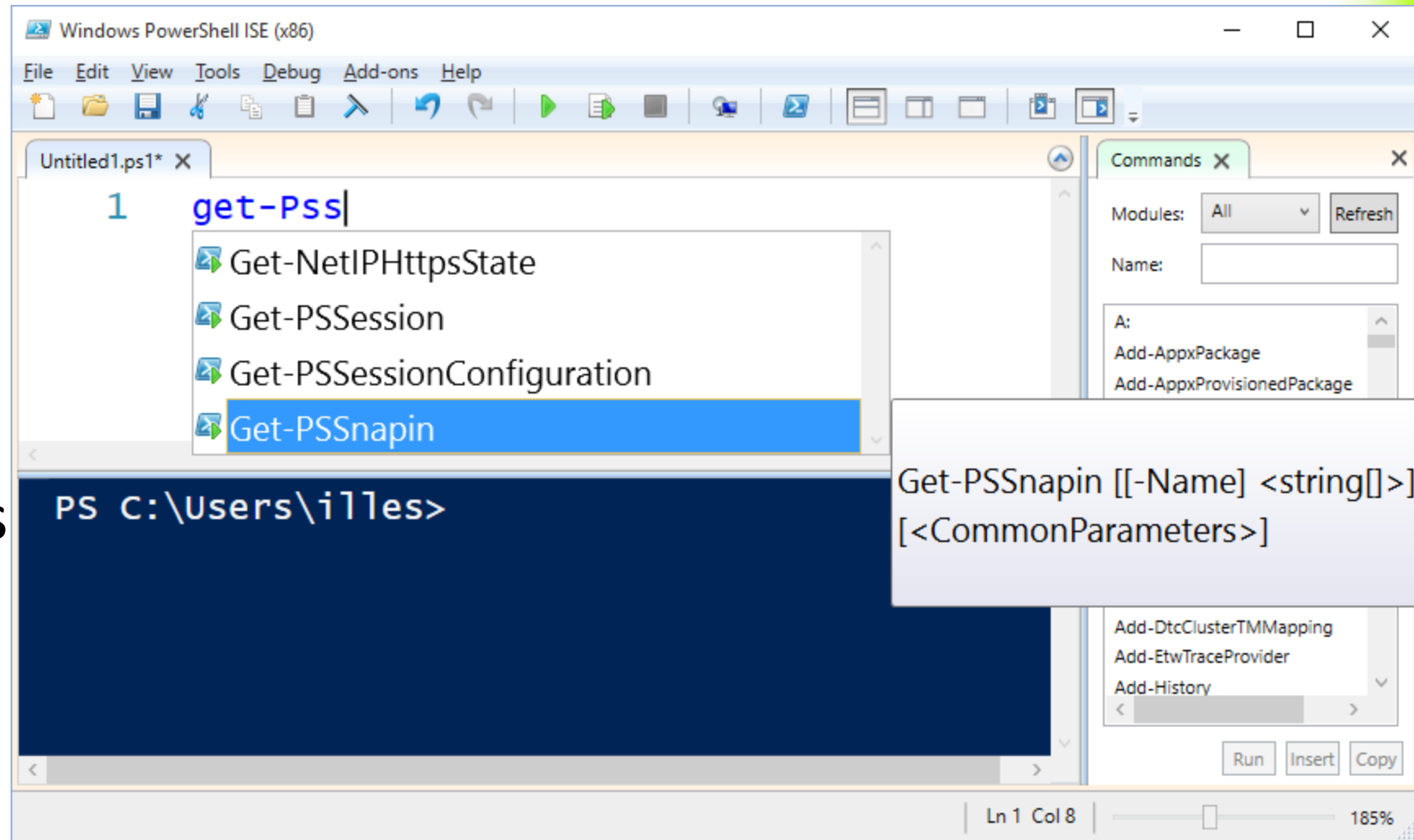
# Powershell „változatok”

- Alap (Core), minden operációs rendszer része
- A szolgáltatások (MS SQL, Web, Exchange) saját kiegészítést (modult) adnak az alap rendszerhez!



# PowerShell (core) ma

- Szokásos grafikus lehetőségek (File, Edit, View)
- Tools – munka ablak font, szín beállítások.
- Debug – Szokásos nyomkövetési lehetőség!



# PowerShell Debug

- F9- Toggle Breakpoint
  - Szokásos step over, stb.
  - Csak ISE-n belül(F5)!!!
    - Parancsablakban indítva nincs a töréspontnak hatása
- Kiíráthatók a script változók!
- Speciális változók:
  - A debugger belső pipeline változói!
    - \$Args
    - \$MyInvocation
    - \$Input
    - \$\_
    - \$PSBoundParameters

The screenshot shows the Windows PowerShell ISE (x86) interface. The script editor displays a PowerShell script with the following content:

```
1 "hajra"
2 for($i=1;$i -lt 10;$i++)
3 {
4     $i.ToString()+"Fut a powershell"
5 }
6 $max=0
7 $nev="almás"
8 dir|ForEach-Object -Process { if ($_.Length -gt $max)
9 { $max=$_.Length
10  $nev=$_ .FullName
```

A breakpoint is set at line 8. The debug console at the bottom shows the following output:

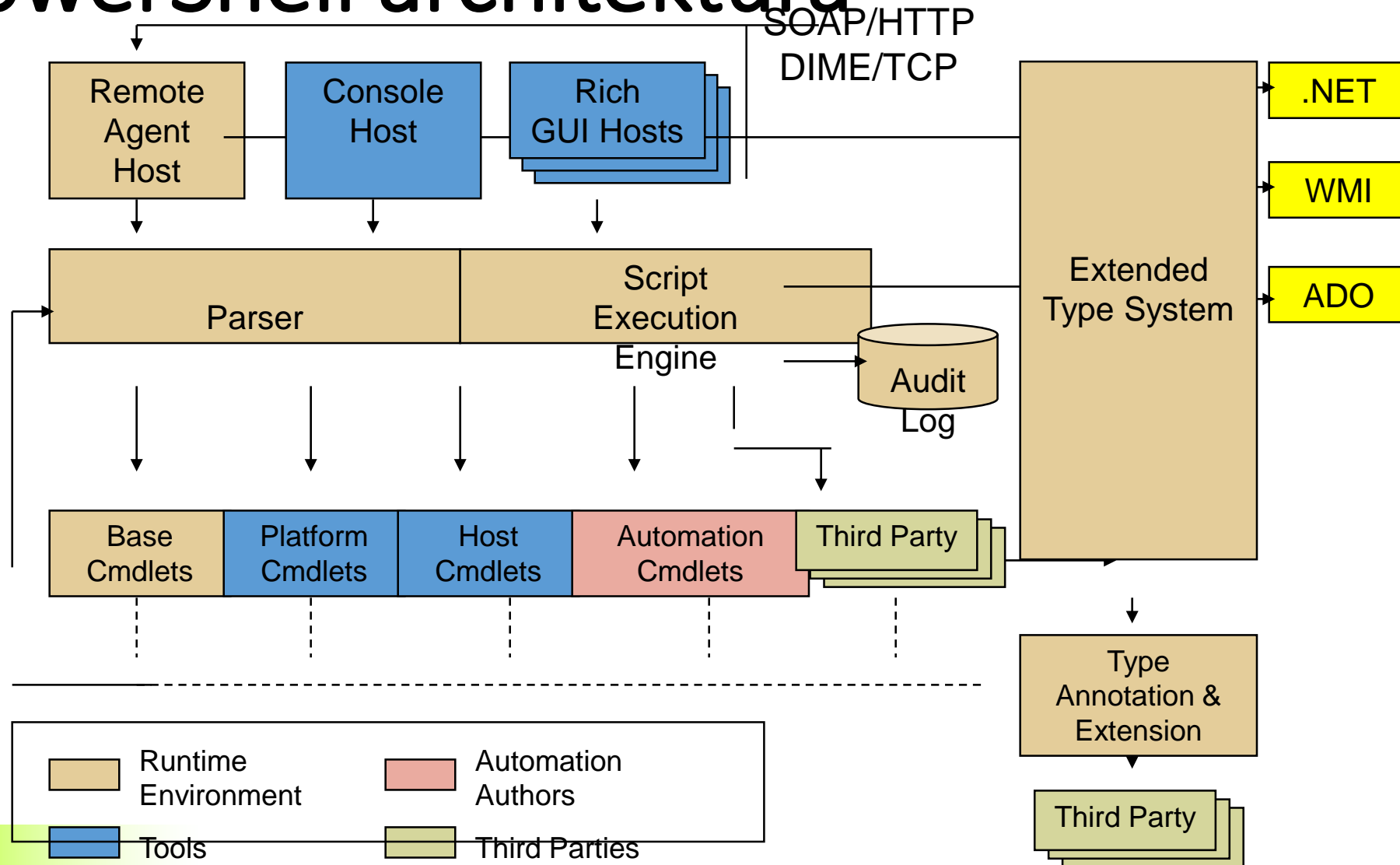
```
[DBG]: PS C:\Users\illes>> $max
0

[DBG]: PS C:\Users\illes>> $i
10

[DBG]: PS C:\Users\illes>>
```

The status bar at the bottom indicates "Completed", "Ln 45 Col 28", and "185%".

# PowerShell architektúra



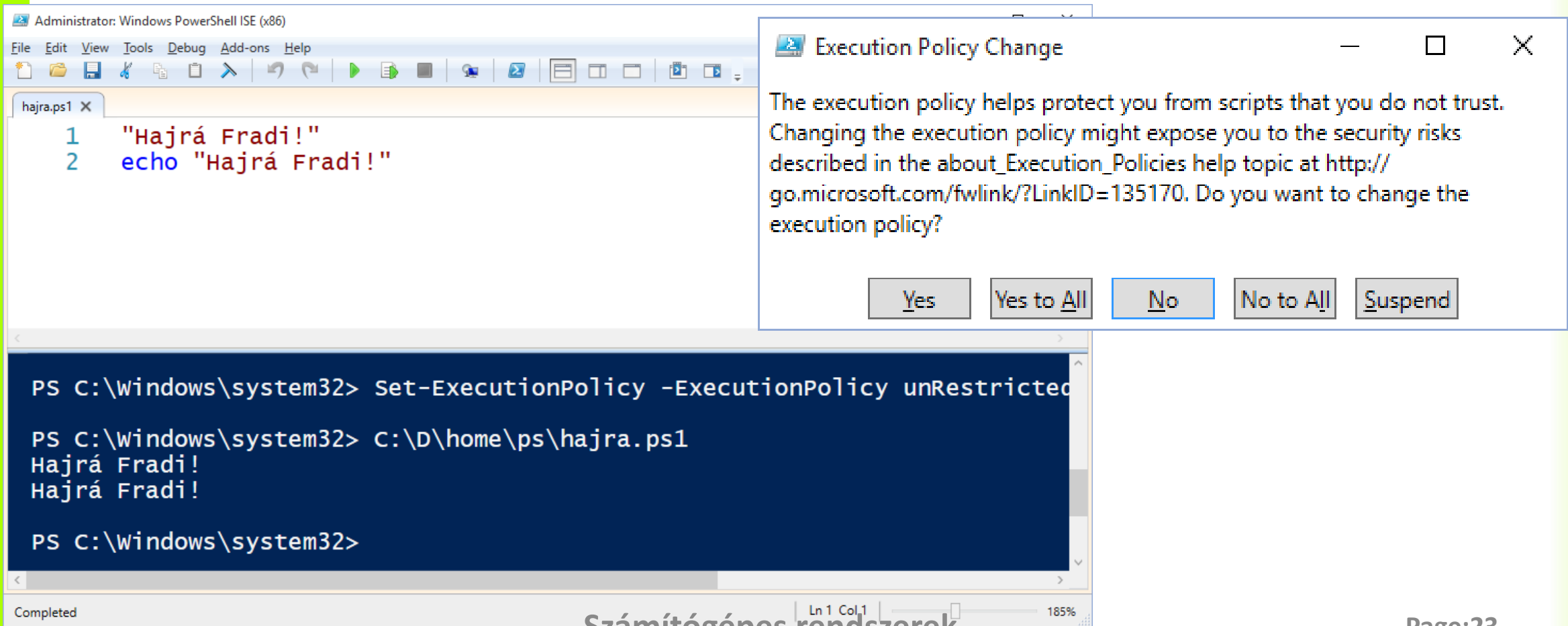
Modified:2018. 11. 03.

# Biztonságos script végrehajtás

- Get-ExecutionPolicy
- Set-ExecutionPolicy –ExecutionPolicy UnRestricted
  - Alapértelmezett: Restricted – nem engedélyezett a futtatás!
  - Lehetséges policy értékek: Allsigned, Remotesigned, Bypass
    - Unrestricted esetén a letöltött, nem aláírt script esetén rákérdez, Bypassnál ezt sem!
  - -scope process vagy currentuser vagy localmachine
  - Remotesigned: Internetről letöltött állományok esetén csak akkor futtatja, ha megbízható partner írta alá.
- Fontos! PowerShell indítása: Run As Administrator!

# PS script futtatás

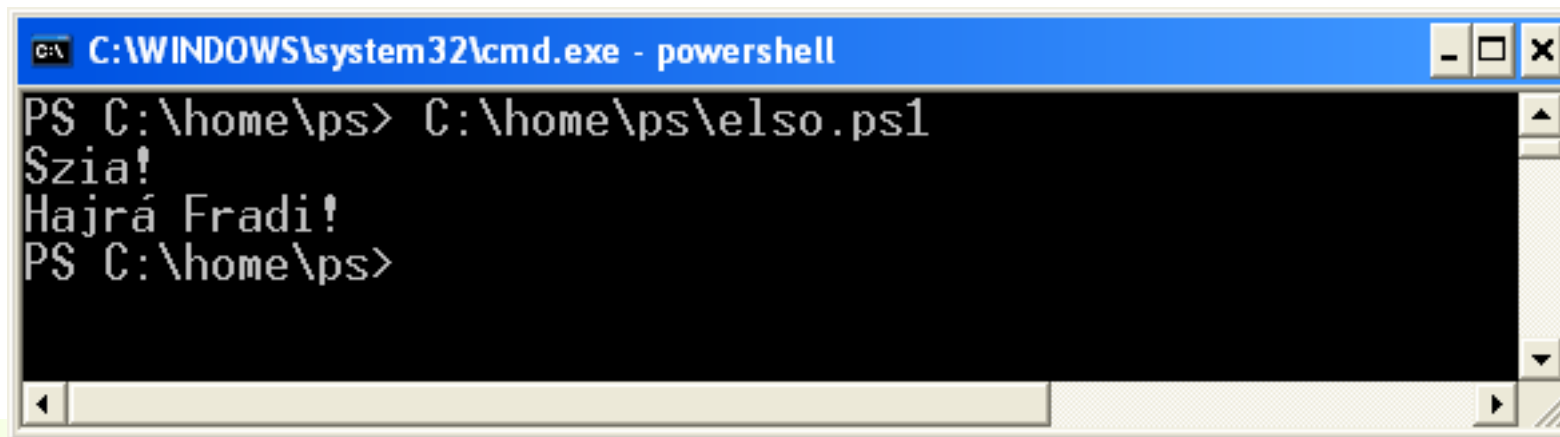
- Fájlnev kiterjesztés: ps1



# PS script futtatása

- Parancs futtatás, teljes útvonal beírásával
- Ha helyköz van egy könyvtárban, akkor az & jelet írjuk a parancs elé, és "" között legyen a parancs.
  - &"c:\alma fa\jonatán.ps1"

```
# Megjegyzés  
echo Szia!  
#  
Write-Host "Hajrá Fradi!"
```

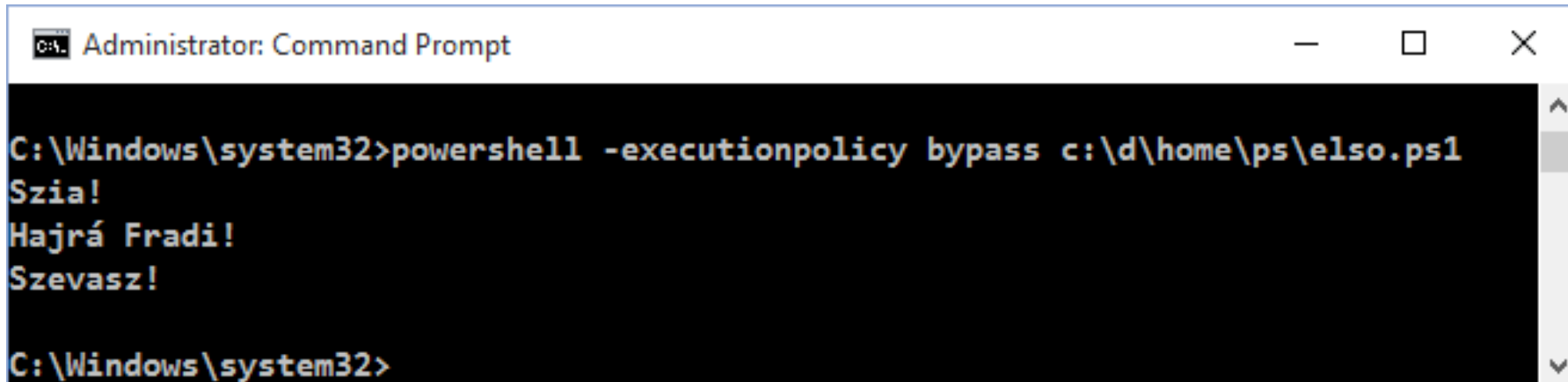


```
C:\WINDOWS\system32\cmd.exe - powershell  
PS C:\home\ps> C:\home\ps\elso.ps1  
Szia!  
Hajrá Fradi!  
PS C:\home\ps>
```



# PS script futtatás parancssorból

- Meg kell adni a scriptet mint paramétert!
- Meg kell adni az executionpolicy paramétert! (ha nincs beállítva!)



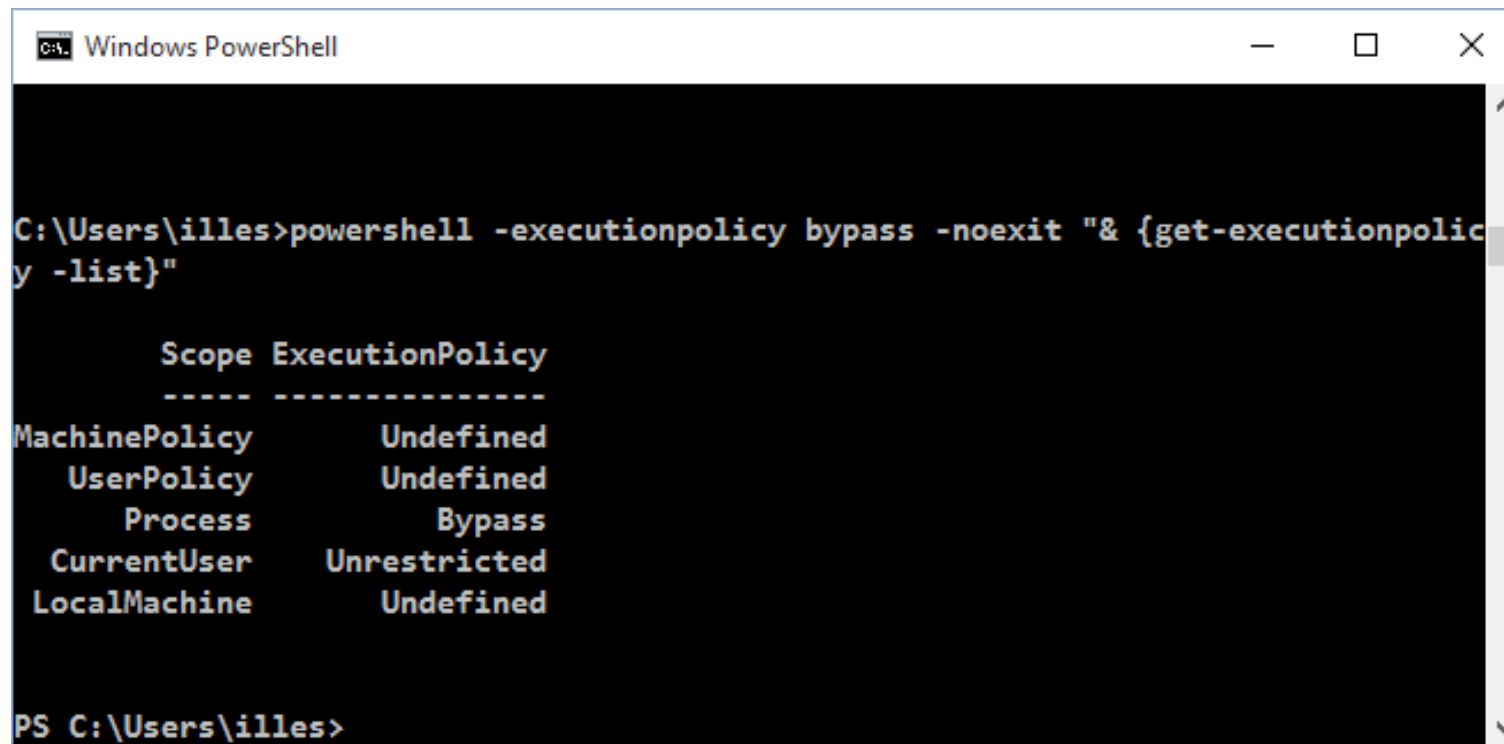
```
Administrator: Command Prompt

C:\Windows\system32>powershell -executionpolicy bypass c:\d\home\ps\elso.ps1
Szia!
Hajrá Fradi!
Szevasz!

C:\Windows\system32>
```

# PS parancs futtatása

- PS paraméter: "& {parancs}"
- -noexit nem lép ki a PS-ből



```
CA: Windows PowerShell

C:\Users\illes>powershell -executionpolicy bypass -noexit "& {get-executionpolicy -list}"

Scope ExecutionPolicy
-----
MachinePolicy Undefined
UserPolicy Undefined
Process Bypass
CurrentUser Unrestricted
LocalMachine Undefined

PS C:\Users\illes>
```

# PowerShell parancssor

- Ha konstanst írunk, azt az értelmező próbálja egy típushoz illeszteni.
- Ha mást nem mondunk, az alap művelet: write-host
  - Azaz egy konstans leírása automatikusan annak képernyőre írását jelenti!

```
PS C:\Users\illes> fradi
The term 'fradi' is not recognized as a cmd in. At line:1
char:5 + fradi <<<<
PS C:\Users\illes> "fradi"
fradi
PS C:\Users\illes> 5
5
PS C:\Users\illes> 'F'
F
PS C:\Users\illes> F
The term 'F' is not recognized as a cmdlet, At line:1char:1
+ F <<<<
```

# PS parancsok formája

- Két nagy család (kezdetben, ma több😊)
  - Get-parancs család, Set-parancs család
- PS parancs : commandlets- cmdlets
  - Beépített parancsok + kiegészítések.
  - „Hagyományos parancsok” (echo) is használhatók!
- A Tab billentyű kiegészíti a parancsot
- Megjegyzés: #
  - Többsoros megjegyzés: <# ....#>
  - Script elején ha szerepel benne .Description, .Syntax,.Synopsis, .Examples blokk valamelyike, akkor a get-help script ezt veszi alapul!
- Kis-nagybetűre nem érzékeny

# PowerShell Help

- A segítség parancsa: Get-Help
- Alapból csak a parancsok szintakszisa érhető el!
- Update-Help parancs a lokális gépre installálja a dokumentációt!
  - Hasznosabb az online help!
- <https://technet.microsoft.com/en-us/library/hh847741.aspx>
  - PS 5.0
- Magyar nyelvű help (PS 2.0 pdf)

# Köszönöm a figyelmet!

