

10. gyakorlat PowerShell gyakorlat I.

PowerShell alapok megismerése, scriptek írása, a már megismert UNIX shelllel való összehasonlítása

Mondjuk el, hogy a PowerShell parancsok objektumokon dolgoznak! Nem kisbetű-nagybetű érzékeny! UTF-8 kódolású, magyar ékezetes betűk is használhatóak! A .NET-ben megvalósított lehetőségeket, osztályokat is felhasználhatjuk. Mondjuk el, hogy a parancsoknak van aliasa, a shellben megszokott formalizmus is használható! Csővezeték, IO átirányítás a megszokott módon történik! Beszéljünk a script futtatási jogosultságról (executionpolicy), amit be kell otthon állítaniuk. Grafikus, karakteres változat.

Feladatok

1. Mutassuk meg, hogy hol található a Windows-ban! Win8 és Win10 alatt:
%SystemRoot%\system32\WindowsPowerShell\v1.0\powershell.exe. Egyébként keresés PowerShell ISE.
2. Felfele nyíl, tabulátor, intellisense
3. Fontos ismerni a help-et, itt get-help, get-command. Mutassuk meg a PS utasítások szerkezetét: parancs, paraméterek és argumentumok.
4. Próbáljuk ki a get-childitem, copy-item, rename-item, remove-item, new-item file/directory, get-content, set-location, push-location, pop-location parancsokat. Figyeljünk a write-host és write-output különbségre! Mutassuk meg, hogy még a dátum is objektum! (get-date).year...
5. Csővezeték használata ugyanolyan, mint a UNIX-ban, de az eredmény objektum. Ismerjük meg a következő szűrőket: format-list, format-table, sort-object (sort), where-object, select-object, select-string (grep), measure-object (wc), out-File, get-member!
6. Változók – get-variable (pshome, home) - utaljunk a shell env, set parancsaira! Hozzunk létre saját változókat! Mutassuk meg a unixtől eltérő lehetőségeket! Objektum is lehet egy változó értéke! Adjuk értéket egy változónak egy fájl tartalom beolvasásával, amit szöveges tömbként használhatunk! (\$X=get-content file; \$x[2]; \$x.length; \$x+="új elem") Lássunk egy asszociatív tömböt is! \$y=@{„a”=„A”;„b”=„B”}; \$y[„b”]; \$y.keys ; \$y.values
7. Készítsünk egy egyszerű scriptet, helló világ! Kiterjesztése ps1, (set-executionpolicy remotesigned). Mutassuk meg a PS ISE használatát!
8. Készítsünk scriptet, amelyik az aktuális könyvtár bejegyzései közül a fájlokat kiírja!
9. Készítsünk scriptet, amelyik meghatározza egy fájl sorainak a számát!
(cat fájl | measure-object).Count; UNIX-ban cat fájl | wc -l)
10. Készítsünk scriptet, amelyik kiírja a paramétereinek számát! (\$args.Length; Unix-ban \$#)
11. Készítsünk scriptet, amelyik az első és második paramétert összeadja illetve kiírja a szorzatukat is! (UNIX-ban kell az expr és a `` is)
12. Készítsünk scriptet, amelyik egy fájl soraiban lévő szavak közül kiírja a másodikat! A szavak között például legyen szóköz. (\$sor.split(„ „”)[1], UNIX-ban cut)
13. Készítsünk scriptet, amelyik egy fájl sorainak 2-5 karaktereit írja ki!
((ls).substring[2,3]; UNIX-ban cut -c2-5)