PowerShell I.

Summary: In this lesson we will learn about PowerShell, the most modern script language (not case-sensitive) of Windows operating system. You will see that you can use the well-known unix command syntax as well, but here you have far more possibilities due to the usage of objects. The usage of output redirection and pipelines are the same. You will get to know the usage of PowerShell ISE, which helps you with a professional intellisense and learn how to write small scripts.

Execution permission: - *set-ExecutionPolicy -ExecutionPolicy RemoteSigned (to be able to execute your scripts, run it as administrator)*
Commands*: get-hep, get-command (man), get-childitem (ls), set-location (cd), get-location (pwd), push-location, pop-location (to remember the location), copy-item, remove-item, rename-itam (cp, rm, mv), new-item (mkdir or new file), write-host, write-output (echo, difference!!!), get-content (cat)*
Filters: *format-table, format-list (formatting of output), select- (like grep), select-object string –First/-Last/-Property (like head, tail, or let through a given part of objects), where-object ($_ means the actual object), sort-object (sort), measure-object (wc), tee-object (tee), get-member (gives back the properties and methods of objects)*
Variables: *Get-ChildItem Env (env), $name=value (to set the value), $name (to get the value), $array=@() (empty),* $name=@{key1=value;key2....}(for *associative array)*
Script: *$args (array of parameters), $args.Length (the number of parameters), param ($name) (named parameters)*

Base commands

a) Create a directory with your Neptun code! (new-item)
b) Change to this directory! (set-location)
c) Copy some txt files from the system to this directory! (copy-item)
d) Rename one of it! (rename-item)
e) List the content of the directory! (get-content)
f) Write out hello on the screen! (write-host)
g) Try to write it in colors! Use get-help to find the appropriate properties!
h) Redirect a hello message into the hello.txt file! (write-output, >)
i) See the content of it! (get-content)
j) Remove it! (remove-item)
k) Remember the location – change to somewhere else and return to the original place! (push-location, set-location, pop-location))

Filters

l) List the content of the actual directory in a nicer format! Try format-list and format-table!
m) Sort the content of the directory by file-size! (get-childitem, sort-object)
n) Filter the content of the actual directory to txt files! (get-childitem or get-childitem, where-object)
o) Count the files inside the actual directory! (get-childitem, measure-object)
p) Find that lines in a text file which contain the word apple! (get-content, select-string)
q) Write out the methods and properties of a date object! (get-date, get-member)

Financed from the financial support ELTE won from the Higher Education Restructuring Fund of the Hungarian Government

r) Filter the object's properties of a directory list to filename and size - going through the pipeline! get-childitem, select-object
s) Write out the filenames and sizes of the files (actual directory) – in this order! (get-childitem, select-object)

Variables

t) List out the variables! (get-variable) What was the command for the same goal in UNIX?
u) Create custom variables – first in command line!
    a. Create a $d variable with the actual date! Write out the content of it! Write out only the year! (get-date, .Year)
    b. Create a $f variable, the starting value should be the content of a text file! Write out the first line of it! (get-content, use as an array)
    c. Create a small English-Hungarian dictionary! (apple = alma, child=gyerek, cat=cica, dog=kutya) Use associative arrays for the implementation! Write out the hungarian word for dog!

First scripts

v) Implement a PowerShell script which writes out „Hello world!"
w) Write a PowerShell script which gives back the number of the lines of a file! (measure-object or use an array and use its length property)
x) Write a script which writes out how many parameters we used in command line! ($args)
y) Write a Powershell script which adds its first and second parameter! (In UNIX we needed expr and `` in PS you can simply give the mathematical expression and it will be evaulated!)
z) Write a PS script which writes out each second word from the lines of a given file! (string.split) (In UNIX you can use cut or sed or awk)
aa) Write a PS script which gets a file and writes out only the characters between the 2nd and the 5th position of the lines. (string.substring())

Financed from the financial support ELTE won from the Higher Education Restructuring Fund of the Hungarian Government