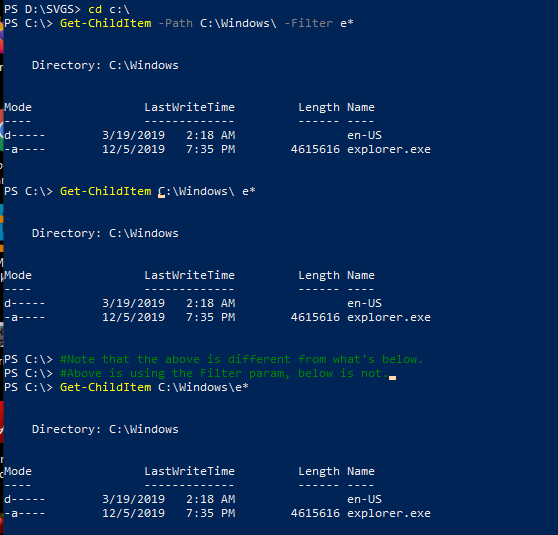
# PowerShell Lab 2 KEY

## More with help and positional parameters

## Exercise with positional parameters

Look at the help for Get-ChildItem, also known by its aliases, gci, dir and ls. Change directory (or Set-Location) to the root of the C:\ drive. Use Get-ChildItem (or an alias) to find all the files and directories in C:\Windows sub-directory that start with the letter “e”, using the parameters “Path” and “Filter”. Execute the command once with parameter names and once without. Note: The version of Get-ChildItem that uses both Path and Filter is the second one listed by Get-Help. Note: you can do the same thing without “Filter” if you put the wildcard in the path.   


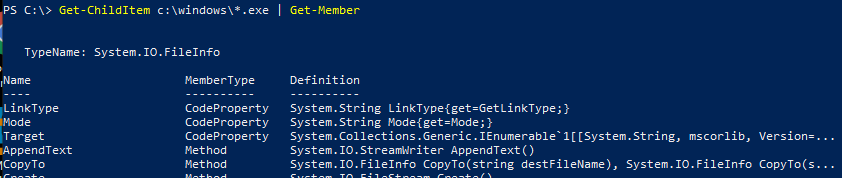
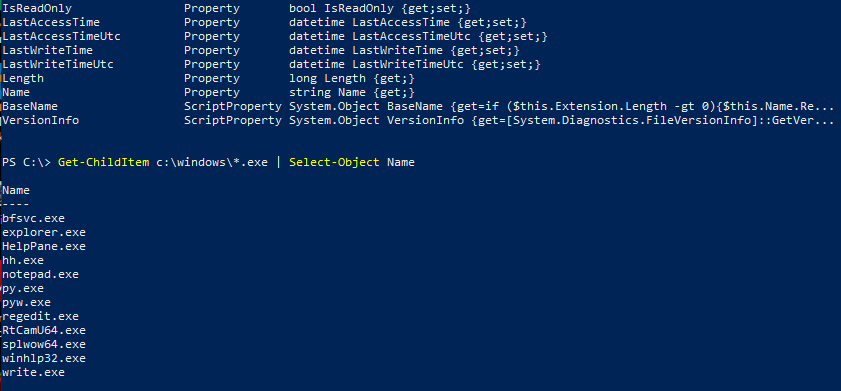
## Exercise with Pipes

Analyze the following command (it is just one line, too long to fit on the page), and say what the command does before you run it. Note: parameters without values, like CaseSensitive, are called “switches.” They cause something to happen or not, depending on whether or not they are present.

Get-ChildItem c:\windows\\*.exe | Select-Object Name | Select-String 'regedit'

Get-ChildItem finds all files in c:\windows with .exe extensions. Then Select-Object filters most of the other stuff except Name. Finally Select-String just picks the ones that include ‘codec’.



This shows that it is still a FileInfo object and still has some methods after going through Select-Object Name . You’ll see a big difference between   
Get-ChildItem c:\windows\ \*.exe | Get-Member and  
Get-ChildItem c:\windows\ \*.exe | Select-Object Name | Get-Member  
  
<snip>  


## Exercise with regular expressions

Like most modern shells and languages, PowerShell can use regular expressions (regex). We covered regular expressions in BASH Lab 5, Parsing and Searching, if you need a review.

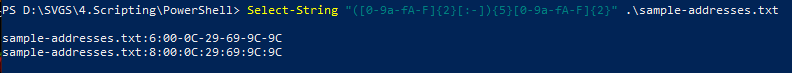
Examine the regex "\w+@[a-zA-Z\_]+\.[a-zA-Z]{2,6}" to see if you can determine what it matches. Then test your guess by running it against the file sample-addresses.txt (from Canvas) with Select-String and the regex as the value for the parameter –Pattern. Note: Select-String is not a direct replacement for Linux grep, but it has similar capabilities.

The first one grabs email addresses.  

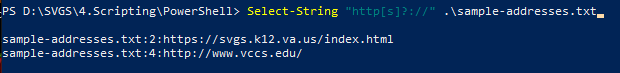

Now, do the same with these patterns:

(\d{1,3}\.){3}\d{1,3} ip addresses  


([0-9a-fA-F]{2}:){5}[0-9a-fA-F]{2} MAC address with :  


([0-9a-fA-F]{2}[:-]){5}[0-9a-fA-F]{2} MAC address with : or -  


Make a regex that will grab an http or https URL from sample-addresses.txt. There are lots of ways to do it, you don’t need to get fancy.

You could get really fancy with this one. I chose a filter that just includes http:// and https://. The [s]? says that there can be zero or more ‘s’ characters. You could simplify it by removing the square brackets. Quantifiers like ‘?’ only work on the character just before, but the brackets make that more obvious.  
Select-String "http[s]?://" .\sample-addresses.txt  
  
Select-String "https?://" .\sample-addresses.txt  
