

Christmas Cheer Laser, part 5

Where-Object vs. Select-Object

For me, this was one of the most confusing parts of PowerShell, and in my humble opinion (IMHO) it is due to poor naming. The `Where-Object` cmdlet tests objects as they travel down the pipeline and drops any object that doesn't pass the test. Ok, then what does `Select-Object` do? `Select-Object` also examines the objects going down the pipeline; usually it allows all objects to pass, but it only allows certain properties of each object to pass. I think `Select-Property` would be a much better name. For example, `Get-ChildItem` has a lot of properties; if we were only interested in outputting path for each object, we would say this.

```
Get-ChildItem | Select-Object -Property Path, or, dir | select path
```

Answer to the Previous Question

- 6) Do a search of `/home/elf` to a "shallow" depth. Compute the MD5 hash of all the files you find and output the name and path of the file that has a hash of

25520151A320B5B0D21561F92C8F6224

First, the hint tells us to search `/home/elf` to a "shallow" depth. I will guess "shallow" means 2 layers deep.

```
Get-ChildItem /home/elf -Depth 2
```

This will generate a lot of output, so I won't show it. Now compute the MD5 hash. The cmdlet is `Get-FileHash` and we need to use `-Algorithm MD5`

```
Get-ChildItem /home/elf -Depth 2 | Get-FileHash -Algorithm MD5
```

```
PS /home/elf> Get-ChildItem /home/elf -Depth 2 | Get-FileHash -Algorithm MD5
Get-FileHash : Access to the path '/home/elf/depts' is denied.
At line:1 char:36
+ Get-ChildItem /home/elf -Depth 2 | Get-FileHash -Algorithm MD5
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [Get-FileHash], UnauthorizedAccessException
+ FullyQualifiedErrorId : System.UnauthorizedAccessException,Microsoft.PowerShell.Commands.GetFile
```

Oops. **Access to the path '/home/elf/depts' is denied.** You can't take the hash of a directory; we only want files. If you search on "get-childitem files only" you will find there are a lot of ways to do this; the `-File` option is one of them.

There's still a lot of output. We need to narrow that down to one file. Let's filter on the hash, but, what do we filter on? Let's put the output of our last statement into `Get-Member` and see what it holds.

```
Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 | Get-Member
```

```
PS /home/elf> Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 | Get-Member

    TypeName: Microsoft.PowerShell.Commands.FileHashInfo
Name      MemberType Definition
-----
Equals     Method      bool Equals(System.Object obj)
GetHashCode Method      int GetHashCode()
GetType    Method      type GetType()
ToString   Method      string ToString()
Algorithm  Property    string Algorithm {get;set;}
Hash       Property    string Hash {get;set;}
Path       Property    string Path {get;set;}
```

So, the output is an object with properties Algorithm, Hash, and Path. Note that Hash is a string.

```
Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 |
Where-Object {$_.Hash -eq '25520151A320B5B0D21561F92C8F6224'} | fl
```

```
PS /home/elf> Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 |
>> Where-Object {$_.Hash -eq '25520151A320B5B0D21561F92C8F6224'}

Algorithm      Hash                                     Path
-----
MD5            25520151A320B5B0D21561F92C8F6224      /home/elf/...

PS /home/elf> Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 |
>> Where-Object {$_.Hash -eq '25520151A320B5B0D21561F92C8F6224'} | fl

Algorithm : MD5
Hash      : 25520151A320B5B0D21561F92C8F6224
Path      : /home/elf/depths/produce/thhy5h11.txt

PS /home/elf>
```

Rather than typing that long path, or copying and pasting, we can just pipe our answer into Get-Content.

```
Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 |
Where-Object {$_.Hash -eq '25520151A320B5B0D21561F92C8F6224'} | gc
```

```
PS /home/elf> Get-ChildItem /home/elf -Depth 2 -File | Get-FileHash -Algorithm MD5 |
>> Where-Object {$_.Hash -eq '25520151A320B5B0D21561F92C8F6224'} | gc
temperature?val=-33.5

I am one of many thousand similar txt's contained within the deepest of /home/elf/depths. Finding me will give you
the most strength but doing so will require Piping all the FullName's to Sort Length.
PS /home/elf>
```

Great! At the top of the riddle, it says **temperature?val=-33.5**. Two parameters down, two to go.

This search will be like the last two you've done. This time you want to recursively search the entire /home/elf/depths tree. Then sort the entries by the length of the FullName parameter and take the file with the longest.

Question

7) What is the riddle at the end of this clue?