Scripting PEs

Course	Scripting
Projects	Scripting PEs CBTC
Created time	@March 18, 2025 8:58 AM
Author	Brandon Walker
Est. Minutes	120

All Practical Exercises for Scripting

Scripting

Bash Fundamentals

Bash Scripting

PowerShell Fundamentals

POSH Basic - 00 0

Before Starting these practical exercises:

1. Run Powershell as an administrator

```
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
```

```
Update-Help -Verbose -Force -ErrorAction SilentlyContinue
```

POSH Basic - 02 2

The PowerShell format for cmdlets

Verb-Noun

POSH Basic - 03 2

Powershell is an open-source proprietary programming language developed by the linux foundation

- True
- False

False

POSH Basic - 04 2

All output and inputs from PowerShell are

.NET Objects

POSH Basic - 05 2

PowerShell methods are

- Source of information collection

Actions that can be taken by an object

Information about objects defined

A property or value of a characteristic

Actions that can be taken by an object

POSH Basic - 06 2

Which of the following cmdlet will provide PowerShell help functionality.

man

?

Get-help

ls

Get-help

POSH Basic - 07 2

I want to see examples of the cmdlet get-process what would I type in?

```
Get-Help Get-Process -Examples
```

POSH Basic - 08 2

____ is used to run many commands sequentially.

Redirectors

Pipelines

Operators

Hierarchy

Pipelines

POSH Basic - 09 2

____ are alternate names that can be used to run commands.

Commands

Nicknames

Aliases

Redirectors

Aliases

POSH Basic - 10 2

A PowerShell object is made up of three types of data: the object type, its _____, and its _____.

input as = answer1, answer2

Properties, Methods

POSH Basic - 11 2

The core cmdlets in PowerShell are built in _____

Fairy dust

Python

Wmlc

.NET Core

.NET Core

POSH Basic - 12 2

Cmdlet is often interchangeably called a

Query

Command

Action

Execution

Command

POSH Basic - 13 2

Which of the following is a wildcard character for any string in Powershell

- <
- |
- \
- -
- *
- @

*****

POSH Basic - 14 2

Observing the minimal help documentation for `get-help get-process` which group contains a notice to see more examples?

- Name
- Synopsis
- Syntax
- Description
- Related Links
- Remarks

Remarks

POSH Basic - 15 2

Observing the -full help documentation of get-command, which group contains information about piping into the cmdlet?

NOTE: Ensure your version of powershell is updated to 5.1.19041.4170

- Parameter
- Synopsis
- Syntax
- Inputs
- Related Links
- Remarks

Inputs

POSH Basic - 16 2

You've been entering gci in PowerShell forever since you copied and pasted it from online what is it really running as?

Get-ChildItem

POSH Basic - 17 2

Create an alias every time np is entered will open notepad.exe

What is the absolute path for notepad.exe?

```
Set-Alias -Name np -Value notepad.exe
```

POSH Basic - 18 2

An input mechanism for users to select options or provide input

- add-on
- pipe
- kicker
- parameter

parameter

POSH Basic - 19 2

Variables that you create at the PowerShell command line exist only while the PowerShell window is open.

- Automatic Variable
- Explicit Variable
- User-Created Variable
- Preference Variable

User-Created Variable

POSH Basic - 20 2

Variables that store state information for PowerShell. These variables are created and maintained by Windows PowerShell

- Automatic Variable
- Explicit Variable
- User-Created Variable
- Preference Variable

Automatic Variable

POSH Basic - 21 2

Customized variables that affect the PowerShell operating environment and all commands that run in the environment.

- Automatic Variable
- Explicit Variable
- User-Created Variable
- Preference Variable

Preference Variable

POSH Basic - 22 2

Based on the following what would be the outcome?

```
$apple = "pears","watarmellon","heap","waterfall",15,10,"squirrel"

$apple[3]
```

- waterfall
- pears
- ERROR
- heap
- squirrel
- 15
- 10
- watarmellon
- nothing

waterfall

POSH Basic - 23 2

Based on the following what would be the outcome?

```
$apple = "pears","watarmellon","heap","waterfall",15,10,"squirrel"

$apple[-4]
```

- waterfall
- pears
- ERROR
- heap

squirrel

15

10

watarmellon

nothing

waterfall

PowerShell Objects

POSH Obj - 01 2

PowerShell is an object-oriented language and shell

True

False

True

POSH Obj - 02 2

A pipeline is a series of commands connected by "|"

True

False

True

POSH Obj - 03 2

Every cmdlet can be piped to any other command?

True

False

False

POSH Obj - 04 2

Im running powershell as my user with administrative permissions; Based on the following

If the following text file "test.txt" exists in $home with the words "hello world" written inside of it, what will be the outcome if i run the following command?

```
Get-content $Home\test.txt | Remove-Item
```

The content will be erased

The File will be deleted

The filename will be blank

An error will occur looking for "C:\Users\student\hello world"

An error will occur looking for "C:\windows\hello"

An error will occur looking for "C:\Users\student\hello world"

POSH Obj - 05 2

Based on the following what is Standard-In for Where-Object?

```
Get-ChildItem C:\Windows\System32 | Where-Object {$_.Name -like "bad*"}
```

The Standard input of Get-childitem

The Standard out of Get-childitem

|

Both file locations defined as parameters

User will be prompted

The Standard out of Get-childitem

POSH Obj - 06 2

Based on the following what is Standard-out for Where-Object?

```
Get-ChildItem C:\Windows\System32 | Where-Object {$_.Name -like "bad*"}
```

The Standard input of Get-childitem

The standard out of Get-childitem

The output of the pipelined series anything in system32 starting with bad

The output of only Where-Object before parameters are defined.

User will be prompted

The output of the pipelined series anything in system32 starting with bad

POSH Obj - 07 2

Based on the following "-eq Automatic " is what?

```
Get-Service | Where-Object StartType -eq Automatic | Select-Object -Property Name
```

Command

positional

Cmdlet

Argument

Argument

POSH Obj - 08 2

From the following what is "fl"

```
Get-Process -id 1360 | fl * -Force
```

Format-List

POSH Obj - 09 2

Property members are _____ that describe an object

Attributes

POSH Obj - 10 2

Based on the following "Select-Object" is what?

```
Get-Service | Where-Object StartType -eq Automatic | Select-Object -Property Name
```

cmd

statement

cmdlet

option

cmdlet

POSH Obj - 11 2

Which of the following is not a method available in get-process

(In Powershell version 5)

- close
- CreateObjRef
- Dispose
- Refresh
- Kill
- GetLifetimeService
- Strings
- BeginOutputReadLine

Strings

POSH Obj - 12 2

Which of the following is not a method available in get-help

ToChar

Remove

Length

Replace

Endswith

GetHashCode

GetEnumerator

PadLeft

Length

POSH Obj - 13 2

Which of the following is not a method available in get-typedata

ToString

GetType

Members

Copy

Members

POSH Obj - 14 2

What is the GeoID member type in Get-WinHomeLocation

Property

POSH Obj - 15 2

if given the following code, what would you write in the if condition to evaluate the statement as true and output "You win!"

$teehee="xD"

if (_____)

{

write-output "You win!"

}

$teehee -eq "xD”

PowerShell Functions

POSH Func - 01 2

In PowerShell grouping code together and giving it a name, so that we can call it by that name later, is called

Function

POSH Func - 02 2

How do you run a function?

Help

CMD Line

enter name of function

function followed by the name

enter name of function

POSH Func - 03 2

Based on the following, what is the outcome once you run the function?

```
$arg="Talking Cat"
function hello { "Hello there $arg, how are you?" }
$arg="Schrodinger"
```

Hello there Schrodinger, how are you?

Hello there Talking Cat, how are you?

Nothing

Talking Cat

Hello there Schrodinger, how are you?

POSH Func - 04 2

Developers, when making a script or function, use this to enable script users to provide input at runtime.

Parameter

POSH Func - 05 2

Based on the following what is the function name?

```
$arg="Talking Cat"
function hello { "Hello there $arg, how are you?" }
$arg="Schrodinger"
```

Hello there Schrodinger, how are you?

Hello

Nothing

Talking Cat

Schrodinger

Hello

POSH Func - 07 2

Functions declared will remain if the terminal exits and reopens

- True
- False

False

POSH Func - 08 2

Functions you want to remain persistant should be defined in/at/with

- Using the Set Command
- PowerShell Profile
- Environment Setup
- Bootup

PowerShell Profile

POSH Func - 09 2

Strings in powershell are always objects

- True
- False

True

POSH Func - 10 2

The following are two strings when running the command will _____ the strings

```
$string = "Wubba Lubba"; $string + " Dub Dub"
```

Concatenate

POSH Func - 11 2

What takes a property argument passed and converts it to a string for string manipulation

- Format-List
- Converto.String
- Expand Property

```
$_.ToSting
```

Expand Property

POSH Func - 12 2

This operator will take a string occurrence and put a new one in its place.

-replace

POSH Func - 13 2

What will be the output of the following.

```
$YearofCovid = "Sometimes I think if all this is happening because I didnt forward that email to 10 people"
$YearofCovid -replace "think","wonder" -replace "email","meme"
```

Sometimes I thlnk if all this is happening because i didnt forward that meme to 10 people

Sometimes I wonder if all this is happening because i didnt forward that meme to 10 people

Nothing will output

Sometimes I think if all this is happening because I didnt forward that email to 10 people

Sometimes I wonder if all this is happening because i didnt forward that meme to 10 people

POSH Func - 14 2

What will be the output of the following.

```
"Be8azgoodhpersonxbutzdont8wasteztimextrying8tohprove it." -split {$_ -eq "8" -or $_ -eq "h" -or $_ -eq 'x' -or $_ -eq "z"}
```

Be a good person but dont waste time trying to prove it.

POSH Func - 15 2

Based on the following, two words will be on the same line which is one of them

```
"Be8azgoodhpersonxbutzdont8wasteztimextrying8tohprove it." -split {$_ -eq "8" -or $_ -eq "h" -or $_ -eq 'x' -or $_ -eq "z"}
```

person

time

work

prove

prove

POSH Func - 16 2

A sequence of characters that specifies a search pattern.

REGEX

POSH Func - 17 2

REGEX is short for

Regular Expression

POSH Func - 18 2

The following will result in True

```
'big' -match 'b[iou]g'
```

True

False

True

POSH Func - 19 2

The following will result in True

```
'bog' -match 'b[iou]g'
```

True

False

True

POSH Func - 20 2

The following will result in True

```
'boig' -match 'b[iou]g'
```

True

False

False

POSH Func - 21 2

The result of the following would be?

```
write-host 100
100 -match '[0-9][0-9]'
```

True

False

True

POSH Func - 22 2

Which of the following regex character ranges will match any number?

\w

\d

\k

\u

\d

POSH Func - 23 2

Which of the following regex character ranges will match any word?

\w

\d

\k

\u

\w

POSH Func - 24 2

The period "." will match any character except a _____

New Line

POSH Func - 25 2

Which quantifier is for zero or more times?

^E

?

+

*

*****

POSH Func - 26 2

Which quantifier is for at least one or more times?

^E

?

+

!

+

POSH Func - 27 2

Which quantifier is for zero or one time?

^E

?

+

!

?

POSH Func - 28 2

In Powershell a variable $message contains text. apply a regex to validate true if it begins with the word "error".

`$message -match '^error'`

POSH REGEX - 29 2

"Lesson: B03"

"Regular Expressions"

Task:

Within the HR Employees file, count the total # of times the "**: S**" tag appears.

HR_Employee_list.txt

Filebin: https://filebin.net/bbju9b242jvxr1mt

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read the file content
$fileContent = Get-Content -Path $filePath -Raw

# Define the regex pattern to match ': S'
$pattern = ': S'

# Find all matches for ': S'
$matches = [System.Text.RegularExpressions.Regex]::Matches($fileContent, $pattern)

# Count the total number of times ': S' appears
$matchesCount = $matches.Count

# Output the result
Write-Host "Total number of times ': S' appears: $matchesCount"
```

548

POSH REGEX - 30 2

"Lesson: B03"

"Regular Expressions"

Task:

Within the HR Employees file, count the total # of times "**Antarctica**" appears anywhere in the file.

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read the file content
$fileContent = Get-Content -Path $filePath -Raw

# Define the regex pattern to match 'Antarctica' as a whole word
```

```
$pattern = '\bAntarctica\b'

# Find all matches for the word 'Antarctica'
$matches = [System.Text.RegularExpressions.Regex]::Matches($fileContent, $pattern)

# Count the total number of times 'Antarctica' appears
$matchesCount = $matches.Count

# Output the result
Write-Host "Total number of times 'Antarctica' appears: $matchesCount"
```

179

POSH REGEX - 31 2

"Lesson: B03"

"Regular Expressions"

Task:

`Within the HR Employees file, count the total # of times the "Na" appears at the beginning of the line.`

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Initialize a counter for the occurrences
$count = 0

# Read the file line by line
Get-Content -Path $filePath | ForEach-Object {
    # Define the regex pattern to match 'Na' at the beginning of the line
    if ($_ -match '^Na') {
        $count++
    }
}

# Output the result
Write-Host "Total number of times 'Na' appears at the beginning of a line: $count"
```

1500

POSH REGEX - 32 2

"Lesson: B03"

"Regular Expressions"

Task:

`Within the HR employees file, count the total # of times a "." appears in the file.`

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read the content of the file
$content = Get-Content $filePath -Raw

# Use regex to find all periods and count them
$dotCount = ([regex]::Matches($content, '\.')).Count

# Output the count
Write-Output "Total periods found: $dotCount"
```

7874

POSH REGEX - 33 2

"Lesson: B03"

"Regular Expressions"

Task:

Count the total # of times "**84**" appears in the middle of a Social Security Number.

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read the entire file as a single string
$content = Get-Content $filePath -Raw

# Define the REGEX pattern and count the matches
$pattern = '\d{3}-84-\d{4}'
$count = ([regex]::Matches($content, $pattern)).Count

# Output the result
Write-Output "Total times '84' appears in the middle of an SSN: $count"
```

25

POSH REGEX - 34 2

"Lesson: B03"

"Regular Expressions"

Task:

Within the HR Employees file, count the total # of times two digits appears at the end of the line, using any combination of the numbers "**1,2,or 3**".

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read the file line by line
$content = Get-Content $filePath

# Define the REGEX pattern
$pattern = '[1-3]{2}$'

# Count how many lines match the pattern
$count = ($content | Select-String -Pattern $pattern).Count

# Output the result
Write-Output "Total times two digits (1, 2, or 3) appear at the end of a line: $count"
```

553

POSH REGEX - 35 2

"Lesson: B03"

"Regular Expressions"

Task:

Within the HR Employees file, count the total # of times "**72**" appears at the end of a line.

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read the file line by line
$content = Get-Content $filePath

# Define the REGEX pattern
$pattern = '72$'

# Count how many lines end with "72"
$count = ($content | Select-String -Pattern $pattern).Count

# Output the result
Write-Output "Total times '72' appears at the end of a line: $count"
```

60

POSH REGEX - 36 2

"Lesson: B03"

"Regular Expressions"

Task:

`Within the HR Employees file, count the total # of unique area codes anywhere in the file.`

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read File Content
$fileContent = Get-Content -Path $filePath -Raw

# REGEX pattern to match area codes (numbers in the format (XXX) where X is a digit)
$pattern = '\((\d{3})\)'

# Find all matches for area codes
$areaCodes = [System.Text.RegularExpressions.Regex]::Matches($fileContent, $pattern) | ForEach-Object { $_.Groups[1].Value }

# Get unique area codes
$uniqueAreaCodes = $areaCodes | Sort-Object -Unique

# Count the unique area codes
$uniqueAreaCodesCount = $uniqueAreaCodes.Count

# Output the result
Write-Host "Total unique area codes: $uniqueAreaCodesCount"
```

716

POSH REGEX - 37 2

"Lesson: B03"

"Regular Expressions"

Task:

`Within the HR Employees file, count the total # of IPs in the 150.0.0.0/8 network anywhere in the file.`

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read File Content
$fileContent = Get-Content -Path $filePath -Raw

# REGEX pattern for IPs in the 150.0.0.0/8 network
$pattern = '\b150\.(?:[0-9]{1,3})\.(?:[0-9]{1,3})\.(?:[0-9]{1,3})\b'

# Find all matches for IPs in the 150.0.0.0/8 network
$ips = [System.Text.RegularExpressions.Regex]::Matches($fileContent, $pattern) | ForEach-Object { $_.Value }

# Count the unique IPs
$ipsCount = $ips.Count

# Output the result
Write-Host "Total number of IPs in the 150.0.0.0/8 network: $ipsCount"
```

11

POSH REGEX - 38 2

"Lesson: B03"

"Regular Expressions"

Task:

`Within the HR Employees file, count the total # of` **`".mil"`** `emails in the file.`

```
# Path to file
$filePath = "C:\Users\cvte1\Downloads\HR_Employee_list.txt"

# Read File Content
$fileContent = Get-Content -Path $filePath -Raw

# REGEX pattern for emails ending in .mil
$pattern = '\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.mil\b'

# Find all matches for .mil email addresses
$milEmails = [System.Text.RegularExpressions.Regex]::Matches($fileContent, $pattern) | ForEach-Object { $_.Value }

# Count number of .mil email addresses
$milEmailsCount = $milEmails.Count

# Output the result
Write-Host "Total number of .mil email addresses: $milEmailsCount"
```

215

POSH REGEX - 39 2

"Lesson: B03"

"Regular Expressions"

Task:

`Create a string that matches the following regex pattern`**`[A-Za-z]{1,3}z[0-9a-z].\:`**`"`

```
# Define the regex pattern
$pattern = '[A-Za-z]{1,3}z[0-9a-z]\.:'

# Create a string that matches the pattern
$testString = 'Abz1.:'

# Check if the string matches the pattern
if ($testString -match $pattern) {
    Write-Host "The string matches the pattern."
} else {
    Write-Host "The string does not match the pattern."
}
```

Abz1.:

POSH REGEX - 40 2

"Lesson: B03"

"Regular Expressions"

Task:

`Create a string that matches with the following pattern "`**`((.?.?a[0-9]){2}){2}`**`"`

```
# Define the regex pattern
$pattern = '((.?.?a[0-9]){2}){2}'

# Create a string that matches the pattern
$testString = 'xa1a2ba3a4'

# Check if the string matches the pattern
if ($testString -match $pattern) {
    Write-Host "The string matches the pattern."
} else {
    Write-Host "The string does not match the pattern."
}
```

xa1a2ba3a4
