Advanced Scripting   
Operators-Pattern Matching and Containment

Last Updated: 9/30/2021 10:54 AM Version 1  
Document Prepared for: CIT361 Student

# Name Zach Lucas ID 895649438

# Instructions

Save a copy of this document. Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

# Requirements

PowerShell  
psfiles sample files. You should have these, if not get them here <http://cf.esage.com/psfiles.zip>

# Setup

# Task 1—-Like and -NotLike Basic Pattern Matching

-like and -notlike use simple pattern matching. The wildcard characters are:

|  |  |  |
| --- | --- | --- |
| **Wildcard** | **Matches** | **Example** |
| \* | Match any number of characters | \*.log |
| ? | Match a single character | 1225????.log |
| [<char>-<char>] | Match a range of characters | dorayme[1-3][a-b] |
| [<char><char>] | Matches single characters | [024][ade]xyz |

## Steps

1. Many cmdlets allow basic wildcard matching for parameters. Get-Command is a good example.
   1. To get a list of all the commands that start with New use the New-\* as the command name  
      get-command New-\*
   2. To get all the commands that contain the word process wrap the word process in \*s like this  
      get-command \*process\*
   3. Get all the commands that start with *get-* and the noun starts with the letters a-d  
      Get-Command get-[a-p]\*
2. The -like and -notlike operators can be used wherever you need a comparison. Given the following: predict then verify the expression with PowerShell

|  |  |  |
| --- | --- | --- |
| Expression | Prediction | Result |
| 'red' -like 'r\*' | True | True |
| 'red' -like 'R\*' | True | True |
| 'red' -clike 'R\*' | False | False |
| 8675309 -like '[5-9]??53\*' | True | True |
| 'red' -notlike 'red' | False | False |

# Task 2— -Match and -NotMatch

The match operator uses regular expression matching. Regular expressions are a more complex pattern matching than -like and -notlike. To learn about regular expressions there are numerous tutorials and references online.

## Steps

1. The simplest and probably the most used match is to see if a string contains a specific set of characters. Get familiar with it.

|  |  |  |
| --- | --- | --- |
| Expression | Prediction | Result |
| '99 red baloons' -match 'Red' | True | True |
| '99 red baloons' -notmatch 'Red' | False | False |
| '99 red baloons' -cmatch 'Red' | False | False |

1. You can use the regex or operator | to match one of a list words

|  |  |  |
| --- | --- | --- |
| Expression | Prediction | Result |
| '99 red baloons' -match '(red|blue|green)' | True | True |
| '99 orange baloons' -match '(red|blue|green)' | False | False |
| '99 blue baloons' -match '(red|blue|green)' | True | True |

1. [] can be used to match characters, similar to the -like statement

|  |  |  |
| --- | --- | --- |
| Expression | Prediction | Result |
| '99 orange baloons' -match '[0-9]' | False | True |
| '99 orange baloons' -match '[0-9][0-9]' | True | True |
| '99 orange baloons' -match '[0-9][0-9][0-9]' | False | False |

# Task 3—-Replace

The -replace operator is used to replace a series of characters in a string with a different set of characters. The basic syntax is <string> -replace <matchstring>, <replacestring>. The replace operator uses regular expression matching rather than simple wildcards.

## Steps

1. Try it out:  
   'I love cilantro' -replace 'love','hate'
   1. What is returned?I hate cilantro
2. Try this out:  
   'I love cilantro' -replace 'love'
   1. What is returned? I cilantro
3. And finally:  
   'I love, I mean really love, cilantro.' -replace 'love','hate'
   1. What was returned? I hate, I mean really hate cilantro.

# Task 4— -Split

The -split operator is used to break a string into an array of strings at the matched sequence of characters.

## Steps

1. Split on a space  
   'I love cilantro' -split
   1. What was returned? An error, you need to pass something into it
2. Split on an i  
   'I love cilantro' -split 'i'
   1. What was returned? Love c lantro
3. The unary -split is used to split on white space  
   -split 'I am really bad at spacing !'

# Task 5—-Join

The -join operator is used to combine an array of items into a single string with the characters provided concatenated between array elements.

## Steps

1. Join an array with a comma  
   1,2,3,4 -join ','
2. Join and array with a space  
   1,2,3,4 -join ' '
3. The unary form of -join joins with nothing  
   -join (1,2,3,4)

# Task 6— Containment operators -contains, -notcontains, -in, and -notin

The containment operators are used to test if a collection contains a specific item. The only difference between the -contains and -in operators is which side you put the collection on. Either can be used.

## Steps

1. Enter the following and record your results

|  |  |  |
| --- | --- | --- |
| Expression | Prediction | Result |
| 'red','green','blue' -contains 'Red' | True | True |
| 'red','green','blue' -ccontains 'Red' | False | False |
| 'red','green','blue' -contains 'yellow' | False | False |
| 5 -in 1,2,3,4 | False | False |
| 5 -notin 1,2,3,4 | True | True |

# Deliverable

Upload this document with completed answers to i-learn.