Advanced Scripting   
Operators Misc.

Last Updated: 10/14/2021 7:28 PM Version 1  
Document Prepared for: CIT361 Student

# Name Ryan Skidmore ID 615907798

# 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.

# Requirements

PowerShell

# Task 1—Property and Method Operators

## Steps

1. The . operator is used to access instance properties and methods on objects. Methods require a () after the method name.
   1. Get an object  
      $gem='Diamond'
   2. Access a property  
      $gem.length
      1. How many characters are in $gem? 7
   3. Call a method with no parameters  
      $gem.toupper()
      1. What was returned? DIAMOND
   4. Call a method with a parameter  
      $gem.Substring(3)
      1. What was returned? mond
   5. When calling a method with multiple parameters use a , (comma) to separate parameter values. Call a method with multiple parameters  
      $gem.Substring(3,2)
      1. What was returned? mo
2. The :: operator is used to access static properties and methods. Static members are used with an instance of an object.
   1. Call a static method of the string type  
      [string]::IsNullOrEmpty($gem)
      1. What was returned? false
   2. Access a static property  
      [datetime]::now
      1. What was returned? Sunday, January 30, 2022 4:59:56 PM

# Task 2—Grouping and Subexpressions

## Steps

1. Parenthesis are used to group expressions or force order of operations.
2. Group expressions  
   (1 \* 2.5).gettype()
   1. What is returned? IsPublic IsSerial Name BaseType  
      -------- -------- ---- --------  
      True True Double System.ValueType
3. Group the results of a cmdlet  
   (Get-Process).count
   1. What is returned? 250
4. $() groups statements, may contain loops, statements are separated by ; (semi-colon)   
   "Hi $($n=read-host 'what is your name?';$n.toupper())"
   1. What is returned? what is your name?: ryan  
      Hi RYAN
5. @() is the same as $() except it always returns an array.  
   $a= $($n=read-host 'what is your name?';$n.toupper())  
   $b=@($n=read-host 'what is your name?';$n.toupper())
   1. What type is $a? RYAN
   2. What type is $b? RYAN

# Task 3—Redirection operators

The redirection operator “redirects” the output of a command to a file. The > operator creates a new file whereas the >> operator appends to the file. If the file does not exist, it creates the file.

## Steps

1. Make your psfiles folder your current directory.
2. Redirect standard output to a file
   1. Save you process list to a file  
      get-process > profile.txt
   2. View the contents of the file  
      Get-Content profile.txt
   3. Add a list of aliases to the end of profile.txt  
      get-alias >> profile.txt
   4. Confirm the data was added to the file.
3. Redirect error output. PowerShell has several output streams. Stream #2 is the error stream.
   1. Create some files to work with  
      1,2,3,4|%{New-Item -Type Directory -Name "out$\_"}
   2. Delete the files with some errors  
      0,1,2,3,4,5|%{Remove-Item "out$\_"}
   3. This should have deleted the files you created and tried to delete two files that do not exist, causing a couple of errors.
   4. Recreate the files  
      1,2,3,4|%{New-Item -Type Directory -Name "out$\_"}
   5. Now delete the file redirecting the errors to a file named errors.txt  
      0,1,2,3,4,5|%{Remove-Item "out$\_"} 2> errors.txt
      1. Describe what is in errors.txt? the errors from the deletion of directories, errors came from trying to delete no existent items.

# Deliverable

Upload this document with completed answers to i-learn.