Advanced Scripting   
Tools

Last Updated: 9/16/2021 8:45 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

Install and explore tools to help you with PowerShell.

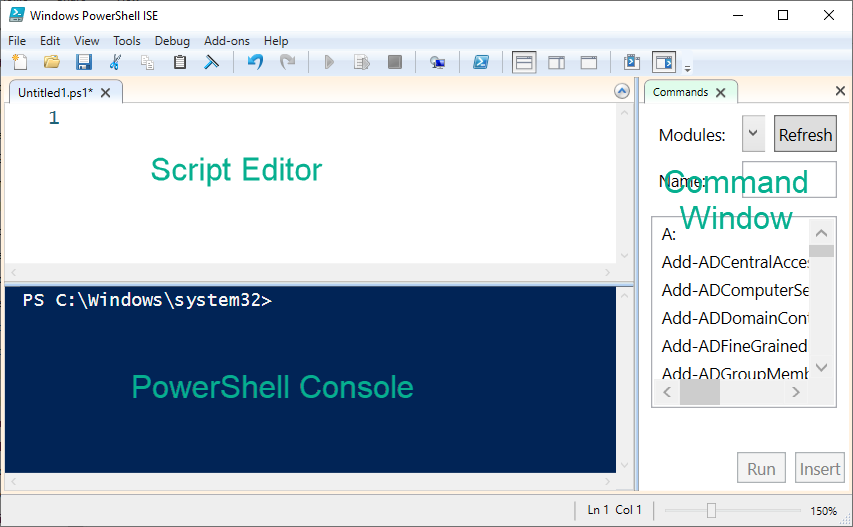
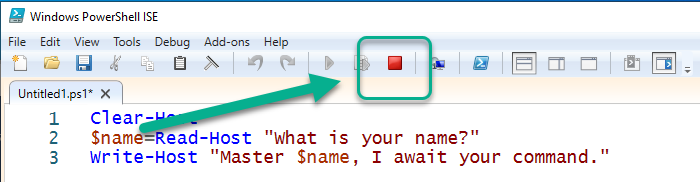
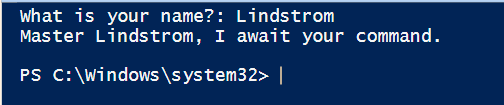
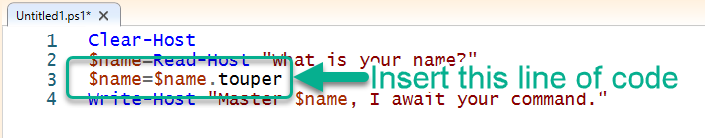
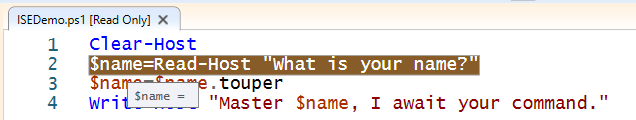
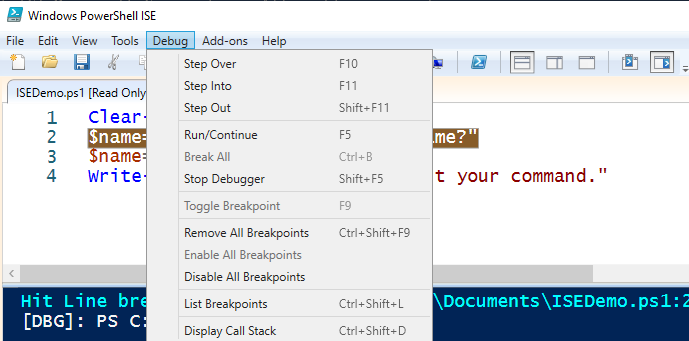
# Setup

Make sure you have access to Windows, either a VM or your main operating system. If you use Windows as your main OS you may still want to use a Windows VM for class. It is up to you.

# Task 1—Exploring the ISE

Launch and explore the Windows PowerShell ISE

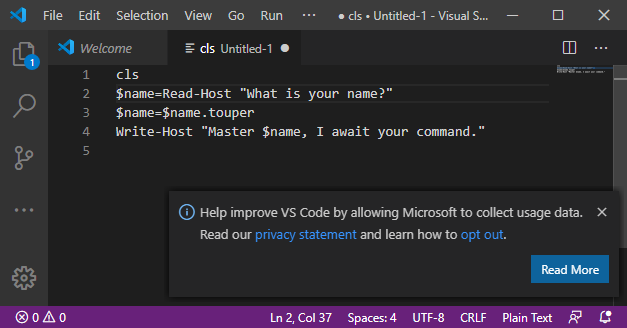
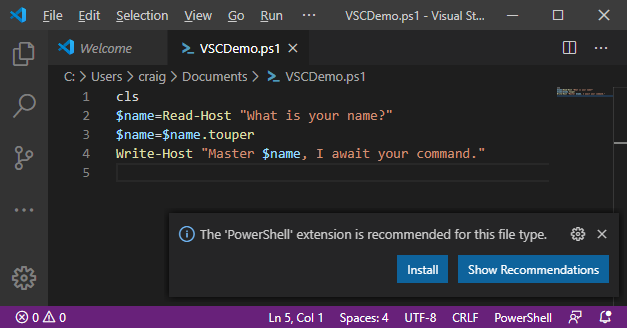
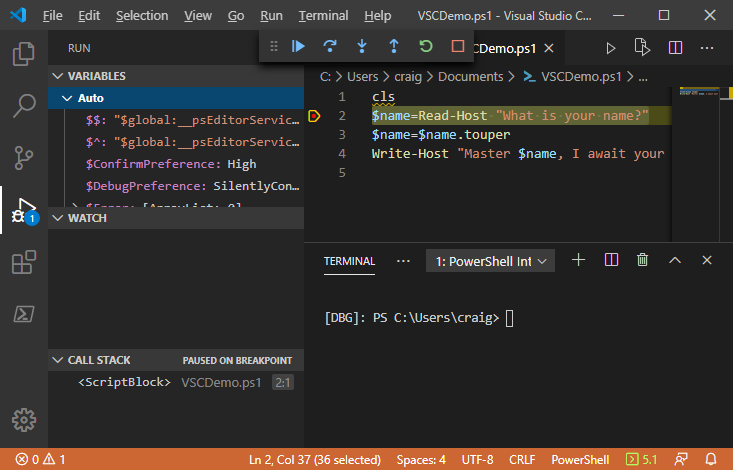
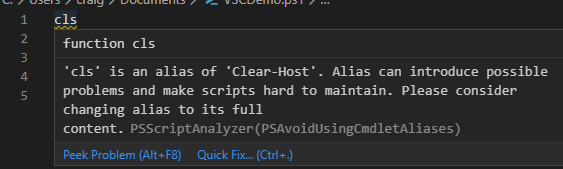
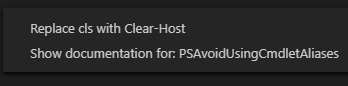
## Steps

1. Launch the PowerShell ISE (use whatever method you like).
2. You should see a new document. If not create one File🡪New  
   
3. Write a script
   1. Enter the following code in the Script Editor  
      Clear-Host  
      $name=Read-Host "What is your name?"  
      Write-Host "Master $name, I await your command."
4. Run your PowerShell script.
   1. Press the **Play button** in the toolbar or the **F5** key to run the program.
   2. You should see the following output in the console window.  
      
   3. Notice the stop button is now colored indicating that your script is currently running.  
      
   4. Enter your name then press enter.
   5. Your script will continue running and show results similar to this  
      
   6. Notice the stop button is no longer Red, this indicates that your script is done running. The Play button is now green, indicating it is ready to run again.
5. Debug your script. When writing code it is useful to be able to stop the program while running and interrogate the value of variables. Variables in PowerShell start with a $, in our script the only variable we use is **$name**.
   1. Let’s say we want to capitalize the name before we print the next message. Modify your script to read exactly as follows.  
      
      1. Now run your program. Did your name come out in all capital letters? It didn’t come out at all, it said everything except for my name.
      2. Copy the output from your script here: Master , I await your command.
   2. Something went wrong. Let’s see if we can figure it out. To start a debugging session you first need to save your file.
      1. Click on the save button  or File🡪Save, or Ctrl+S to save your file. Since the file has not been previously saved the ISE will prompt you for a filename. Name the file **ISEDemo** (the .ps1 extension will automatically be added)
   3. Use variable inspection to see what is happening to the contents of name.
      1. Place your cursor in Line 2 then press the F9 Key, this will toggle a break point on that line.
      2. Run your script again, notice the script stops on Line 2. Line 2 will highlight.
      3. Place your cursor over the variable $name (you don’t need to click there). A tooltip should pop up showing you the current contents of the variable $name  
         
      4. Here is a picture of the Debug menu  
         
      5. Notice the Debug commands along with their keyboard shortcuts. I’ll use the shortcuts in the instructions but if you forget them you can always use the menu. Press the **F11** key to step forward one line of code.
      6. Your console window should now show the prompt for your name. **Enter your name then press enter**.
      7. Notice the highlighted line of code moved down one line. Inspect the variable **$name**. What does it contain? zach
      8. Press F11 to run the next line of code. Inspect the variable **$name**. What does it contain?
      9. Press F11. What is the ouput on your console. Master , I await your command
      10. Your script is now done running. Explain what you think the problem is?  
          I don’t think the built in uppercase command is correct.
   4. Fix the problem. The problem has to do with the **touper** property. It really should be the ToUpper() method. Fix your script to read  
      $name=$name.ToUpper()
   5. Run your script again to see if it is working. If you don’t want the program to stop at your breakpoint either remove it or disable it.

# Task 2—Visual Studio Code

Install and configure VSCode and configure it for PowerShell development.

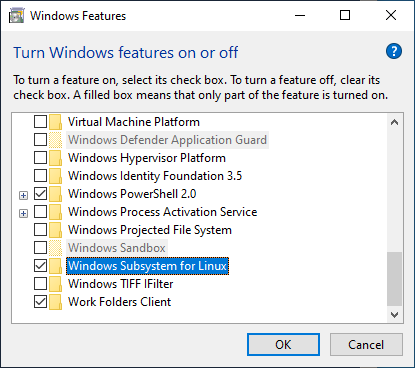
## Steps

1. Install VSCode in Windows. If you already have VSCode installed you can skip this step and continue to step 2.
   1. Download VSCode for Windows from <https://code.visualstudio.com/Download>. Use the 64bit System installer.
   2. Run the installer.
2. Launch VSCode if it is not running.
3. Create Script
   1. Open a new document File🡪New File
   2. Enter the following code.  
      cls  
      $name=Read-Host "What is your name?"  
      $name=$name.touper  
      Write-Host "Master $name, I await your command."
   3. You should end up seeing something like this  
      
   4. Notice there is no color coding and no PowerShell console. This is for two reasons. First since we have not saved the file, VSCode does not know what kind of file it is. Second, even if VSCode knows what kind of file it is, VSCode does not have a PowerShell Extension installed. Let’s fix the problem. First save the file
      1. Press Ctrl+S to save the file (or save it from the File🡪Save Menu) . Name the file:  
         VSCDemo.ps1 (Make sure you include the .ps1 extension)
      2. Shortly after your file saves you should see a message at the bottom of your screen  
         
      3. VSCode realized that this is a PowerShell file and suggested a recommended extension to work with the file. Click **Install** to install the extension. In a few moments the extension will be installed.
      4. Notice that you now have color coding and a powershell console.
   5. Run the script. press F5 to run the script
      1. You should see the prompt in the console window. Enter your name then press enter.
   6. Use F9 to toggle a break point on line 2 then press F5 to run the script. It should stop at line 2 just like in the PowerShell ISE.
      1. Can you hover over a variable to see its value? Yes, it is $null
   7. My screen looks like this:  
      
4. There is a lot going on here! Lets dig into it.
   1. The **Variables** window shows all of the variables PowerShell is using, both the ones defined by PowerShell and also the ones the user defined.
   2. The red dot by line 2 shows there is a breakpoint on that line
   3. The Highlighted line is where the code is currently executing
   4. The Debug toolbar is displayed. Hover over the buttons in the toolbar at the top of the screen and record what each icon means.
      1. What does  Do? Continue
      2. What does  Do? Step over
      3. What does  Do? Step into
      4. What does  Do? Step out
      5. What does  Do? Restart
      6. What does  Do? Stop
5. Fix the code so it works correctly.
6. Remove the break point by clicking on the red dot.
7. Clean up code
   1. Notice there is a yellow squiggly under the cls on line 1. This indicates there is a potential issue with the code. Hover your mouse over the squiggly line. You should see  
      
   2. As you can see the linter is telling you that cls in an alias and best practice it to use the full name. If you press in the QuickFix link the linter will offer suggested fixes.  
      
   3. Click on Replace cls with Clear-Host

# Task 3—Windows Subsystem for Linux (WSL)

You can run Linux programs directly on Windows using WSL. If you have not already installed this on your system add it in this exercise

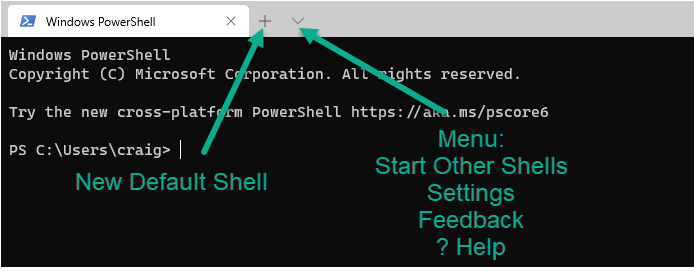
## Steps

1. Enable WSL
   1. From powershell enter the command   
      OptionalFeatures.exe
   2. This will open the Windows Features configuration utility.  
      
   3. **Check** the **Windows Subsystem for Linux** checkbox and click **OK**. After a bit the install will complete. You may need to restart.
   4. After Windows restarts open the **Microsoft Store**.
   5. Search for **Linux.**
   6. Select the flavor of Linux you want to install. I will use **Debian**.
   7. After it installs, **launch** it. It will take some time to finish installing.
   8. Enter a username and password when prompted.
   9. Yippee! Linux is now installed.
   10. Enter the command   
       cat /proc/version
   11. What version of Linux is reported? 4.4.0

# Task 4—Microsoft Terminal

Finally, a good terminal program will make life a lot easier.

## Steps

1. From the Microsoft Store search for “Windows Terminal” and install the Windows Terminal
2. Launch the terminal when installed.
3. You can have multiple terminal windows open at once.  
   
4. Click the Menu icon to view the menu options.
   1. What Shells can you launch? Powershell, command prompt, ubuntu, azure cloud shell

# Deliverable

Upload this document with completed answers to i-learn.