Advanced Scripting   
Help Documentation

Last Updated: 10/28/2021 8:58 AM Version 1  
Document Prepared for: CIT361 Student

# Name Zach Lucas ID 895649438

# Instructions

Complete the steps then upload both this document and your completed PowerShell script.

# Overview

In this exercise your will add inline documentation to provide information about the function.

# Requirements

PowerShell

# Setup

Create a copy of the script you wrote for Exercise 7.2

# Task 1—Adding Help Documentation.

Help documentation is created by adding a multiline comment to the body of the function just after the function declaration. Start a help topic by using a . then the keyword on a single line. Everything you add below that will be in that help section.

## Steps

1. Using the information in the ‘about\_Comment\_Based\_Help’ help topic, create a help comment block in the Get-Soup function you created last exercise.
2. Include help for at least the following keywords
   1. SYNOPSIS
   2. DESCRIPTION
   3. EXAMPLE (at least two examples)
3. Test your help with the help command  
   get-help get-soup

# Wrap-up

Copy your script code here

function Get-Soup {  
 <#  
 .SYNOPSIS  
 Pick your soup, size and quantity for later output, make sure to be polite!  
 .DESCRIPTION  
 This will grab your soup, size, and quantity and tell you if you truly deserve it if you say please.  
 .EXAMPLE  
 get-soup -Please -soup tomato -size cup -Quantity 2  
 get-soup -soup potato -size bowl -Quantity 4  
 #>  
 [Alias('Soup')]  
 param(  
 [Parameter(Mandatory=$true)]  
 $soup,  
 [Parameter(Mandatory=$true)]  
 [ValidateSet('cup','bowl','pot')]  
 $size,  
 [Parameter(Mandatory=$true)]  
 [validateRange(1,5)]  
 $Quantity = 1,  
 [switch]$Please  
 )  
 if ($please) {  
 "$Quantity $size$(if($Quantity -gt 1){'s'}) of $soup soup for you"  
 }  
 else {  
 "No $soup soup for you!"  
 }  
}

# Deliverable

Upload this document with completed answers **and** your script to i-learn.