Mod 10 PowerShell Arrays & Advanced Flow Control Homework

SCRIPTING ESSENTIALS

DR. BURKMAN

Read through the entire homework problem before starting.

Cards!

We will pretty much follow the logic used in the BASH assignment for getting a card from a deck of cards. Use sftp to get “Mod 10 PowerShell Homework Starter.ps1” to get the pre-made suits.

- The file name has spaces so be sure to use quotes around it when getting it from the sftp server

Flow

Overall flow:

- Declare the play suits
- Function to make a new deck
- Function to remove a card
- Function to get a card
- Call the new deck function
- Menu
  - Simple menu that just calls functions. Just like the BASH assignment

New Deck Function

Use four foreach statements to build up the four decks. Use global variables here.

Also set global variables for four counters that will indicate if a suit is empty.

This function does not take any variables as input.

Remove Card Function

Make a temporary array, then copy all of the suit (minus the chosen card) to that temp array. Return it into the original array. This function takes the original array and the index position of the card to remove.

Do not use global variables in this function.

Function to Pick a Card

The flow is like the BASH solution.

Use global variable where needed in this function (but only where needed).

Ask the user for the number of cards to pick. Ensure the input is a valid integer (not alpha, not float) and that there are enough cards to meet the request. Check the demo for the proper responses.

Wrap the rest in a while loop based on the decrementing number of cards requested.

Check to see if all suits are empty. If so, inform the user and break out of the function

Get a random suit

Function to Pick a Card

For each suit:

- Get the number of cards in the suit. If that is zero then set that counter to 1 (or opposite whatever you initialized it in the new deck function) and continue.
- Else get a random number between zero and the length of the suit array.
- If the count of the suit is 1:
  - Output the card
  - Set that suit array to be an empty suit ($dog = @())
  - Decrement the cards requested amount by one
  - Continue
- If the count of the suit is > 1:
  - output the card
  - Call the array to remove the card
  - Decrement the cards requested amount by one

Function to Pick a Card

Why the check for one thing left in the suit? If you don't, then the function will likely try and parse out the letters of the last card rather than just pick the last card. If you don't have this problem then you are free to skip my flow that checks for one card remaining.

Tips

Write-Host can do multiline.

Remember –match and –nomatch for regex

Remember to $dog/1 when you want dog to do match things.

Watch the demo!