Mildred Suriel

Intro to Linux

Week 12 Scripts with Looping

April 21st, 2019

Lab 12

#!/bin/bash

# Let the user know that you are taking in their phrase

echo "Give a phrase that you would like to be repeated"

# Use the read command to get the users phrase. The read command usually stops reading after a space is seen

# To get a phrase to work, IFS= is used to interpret all characters as input so space is not skipped

# We also need -r so that the input is read as raw values so that the carriage return added at the end of lines when enter

# is pressed does not cause issues

IFS= read -r inputPhrase

# Let the user know we are reading in the number of times to run

echo "How many times do you want us to repeat the phrase?"

# Read in the number from the user.

read inputRepeat

# We want to make sure that the input was a number, otherwise the for loop doesn't work right

# We compare the read input variable to make sure that all characters are a number

# If this isn't true, then we want the if statement to run, so we add the ! to the statement to reverse when it runs

if ! [[ "$inputRepeat" =~ ^[0-9]+$ ]] ; then

echo "ERROR: Input was not a number. Try again"

exit 1

fi

# If the input was a number, we will get to the looping portion now

# We still have the issue where the inputRepeat variable is being read as a string instead of a number

# To fix this, we need to use the eval command to convert the string to an integer

# We also use the echo command to print the value of $inputRepeat since we don't know it

for i in $(eval echo {1..$inputRepeat})

do

# Output the phrase and the number of the loop we are running so that we know how far along the run is

echo "Loop $i: $inputPhrase"

done

#!/bin/bash

# Let the user know we are reading in the number of times to run

echo "We are counting down to 0 from a number! What number do you want us to count down from?"

# Read in the number from the user.

read inputNumber

# We want to make sure that the input was a number, otherwise the for loop doesn't work right

# We compare the read input variable to make sure that all characters are a number

# If this isn't true, then we want the if statement to run, so we add the ! to the statement to reverse when it runs

if ! [[ "$inputNumber" =~ ^[0-9]+$ ]] ; then

echo "ERROR: Input was not a number. Try again"

exit 1

fi

# If the input was a number, we will get to the looping portion now

# We still have the issue where the inputNumber variable is being read as a string instead of a number

# To fix this, we need to use the eval command to convert the string to an integer

# We also use the echo command to print the value of $inputNumber since we don't know it

# We start at inputNumber and count down to 1

for i in $(eval echo {$inputNumber..1})

do

# Output the phrase and the number of the loop we are running so that we know how far along the run is

echo "Counting down from $inputNumber: $i!"

done

Script running:

home/msuriel6033 $ bash week12.sh

Give a phrase that you would like to be repeated

WOW

How many times do you want us to repeat the phrase?

5

Loop 1: WOW

Loop 2: WOW

Loop 3: WOW

Loop 4: WOW

Loop 5: WOW

We are counting down to 0 from a number! What number do you want us to count down from?

9

Counting down from 9: 9!

Counting down from 9: 8!

Counting down from 9: 7!

Counting down from 9: 6!

Counting down from 9: 5!

Counting down from 9: 4!

Counting down from 9: 3!

Counting down from 9: 2!

Counting down from 9: 1!