#! Start of the script

$\* list of all set values

$# number of parameters

Shift moves to the right (shift # moves #of spaces)

-f checks for the file

-eq equals

-ne not equals

-gt greater then

-ge greater or equal

-lt less then

-le less or equal

#!/bin/bash The start of the script

# COURSE: ULI101 Any # after #! Doesn’t effect the script (commenting the script

# USERNAME: brian.gray

# SAMPLE SCRIPT #1

echo My username is brian.gray

printf "enter your full name: "

read name Read asks the user for input from keyboard

echo "Hello, my name is $name." Variable substitution

course="ULI 101"

echo "I am currently enrolled in $course."

Example 2

#!/bin/bash

# COURSE: ULI101

# USERNAME: brian.gray

# SAMPLE SCRIPT #2

# Using and shifting positional parameters

echo "\$0 is: $0"

echo "\$1 is: $1"

echo "\$2 is: $2"

echo "\$3 is: $3"

echo "\$\* is a list of all positional values (not including \$0): $\*"

echo "\$# is the number of positional values (not including \$0): $#"

echo

echo "the shift command can be used to move positional values to the left"

echo "shifting 1:"

shift

echo

echo "this is what the positional values are now: $\*"

echo "\$1 is now: $1"

echo "shifting 2"

shift 2

echo

echo "this is what the positional values are now: $\*"

echo "\$1 is now: $1"

echo

example 3

#!/bin/bash

# COURSE: ULI101

# USERNAME: brian.gray

# SAMPLE SCRIPT #3

# Using set to assign positional parameters

echo "The positional parameters are: $\*"

echo "Saving original parameters to user defined variables"

arg1=$1

arg2=$2

arg3=$3

echo

echo "Using set to assign new values"

set bring on summer

echo "the new positional parameters are: $\*"

echo

echo "luckily I saved my original values first: $arg1 $arg2 $arg3"

echo

#6

#!/bin/bash

# COURSE: ULI101

# USERNAME: brian.gray

# SAMPLE SCRIPT #6

# Using logic -- if... then... fi

set 58 745 ~/file1 Sean Shawn sets the variables ($1,$2,3$,4$,5$)

if [ $2 -gt $1 ]

then

echo "$2 is greater than $1"

elif [ $1 -gt $2 ]

then

echo "$1 is greater than $2"

else

echo "They are equal"

fi

if [ ! -f $3 ]

then

echo "$3 is not present"

fi

if [ ! $4 = $5 ]

then

echo "$4 and $5 are not the same"

else

echo "$4 and $5 are the same"

fi

#7

#!/bin/bash

# COURSE: ULI101

# USERNAME: brian.gray

# SAMPLE SCRIPT #7

# Using Logic -- for loops

for item in $\*

do

echo "$item" >> shopping.list

done

echo "Don't forget to pick up:"

for item in $(cat shopping.list)

do

echo $item

done

#!/bin/bash

# COURSE: ULI101

# USERNAME: brian.gray

# Walk Through Example

# Example Script used for Walk Through

validcount=0

invalidcount=0

if [ $# -eq 0 ]

then

echo "Please enter at least one valid postal code"

exit 1

fi

for item in $\*

do

echo $item | grep -i "^[a-z][0-9][a-z][0-9][a-z][0-9]$" > /dev/null

if [ $? = 0 ]

then

validcount=$(( $validcount+1 ))

item=$(echo $item| tr 'a-z' 'A-Z')

echo "$item is valid"

else

invalidcount=$(( $invalidcount+1 ))

item=$(echo $item| tr 'A-Z' 'a-z')

echo "$item is invalid"

fi

done

if [ $invalidcount -ge 2 ]

then

echo "Too many invalid Postal Codes"

echo "Quiting"

exit 2

fi

What would be the output of the following commands be:  
 $1 $2 $3 $4  
walkthru.bash  l0m1J0  PIXIE  TTS5tt  M5C3w8

$\*

$# 4

$invalidcount0

$vailedcount0

$iteam l0m1J0

Answer   
  
Then:

echo  $?  
  
  
I will take it up before the quiz.  
  
Brian

Quiz1 script

Script 12