LINUX

Bash Scripting-Part 2

TEJ4M

Looping

```
#!/bin/bash
for i in {1..5}
do
   echo "Welcome $i times"
done
```

```
#!/bin/bash
# set counter 'c' to 1 and condition
# c is less than or equal to 5
for (( c=1; c<=5; c++ ))
do
   echo "Welcome $c times"
done
```

We can implement the classic ‘for’ loop as follows (you may have to change increments as c=$c+1):

For Loop Through Strings

```
#!/bin/bash

for X in cyan magenta yellow
do
    echo $X
done
```

A string with space separated words can be looped through as follows:

While Loop

A while loop iterates until a condition becomes false. Again, note the spacings inside the condition:

```
#!/bin/bash
x=1
while [ $x -le 5 ]
do
  echo "Welcome $x times"
  x=$(( $x + 1 ))
done
```

While Loop to Read Files

A while loop can be used to iterate through the lines of a file as follows:

```
#!/bin/bash

LINE=1

while read -r CURRENT_LINE
    do
        echo "$LINE: $CURRENT_LINE"
    ((LINE++))
done < "sample_file.txt"
```

Example

A file, data.dat contains each letter of the alphabet. We can parse and display each line as follows:

```
~$ cat data.dat
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
```

```
#! /bin/bash
while read -r line
do
        if [ -z $line ]
        then
                echo "Line is null"
        else
                echo "LINE: $line"
        fi
done < data.dat
```

Passing Arguments to Scripts

You can pass values to a script when you run it. The values are stored in the variables $1,$2,$3 etc.

You pass the values as follows:
./script.sh value1 value2 value3 etc.

main.sh ×

```
1  #!/bin/bash
2
3  for x in $@
4  do
5      echo "Entered arg is $x"
6  done
```

Console Shell

```
~/myfirstscript$ ./main.sh pink grey yellow
Entered arg is pink
Entered arg is grey
Entered arg is yellow
~/myfirstscript$
```

Another Sample

```
#!/bin/bash
echo "Script Name: $0"
echo "First Argument: $1"
echo "Second Argument: $2"
```

Note that $0 holds the script name. The variable $# holds the number of arguments.

```
LinuxConfig.org:~$ ./example1.sh Linux Config
Script Name: ./example1.sh
First Argument: Linux
Second Argument: Config
LinuxConfig.org:~$
```

QUESTIONS AND EXERCISES

Create a bash script that does the following:

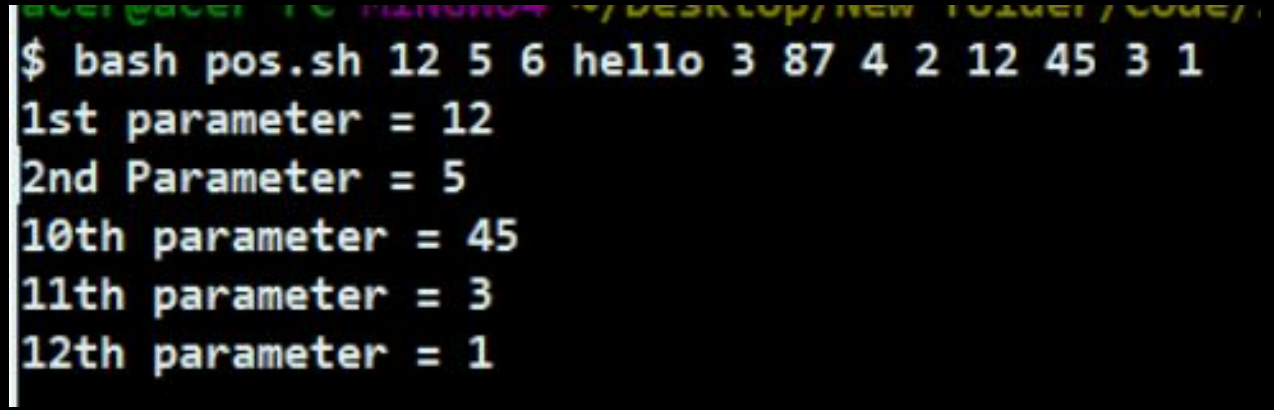
```
$ sh userReg-positional-parameter.sh john 25 'John Smith'
Username : john
Age: 25
Full Name: John Smith
```

QUESTIONS AND EXERCISES

Create a bash script that takes in three 3 numbers as arguments to the script and then displays the sum of those numbers.

QUESTIONS AND EXERCISES

Create a bash script that accepts arguments and does the following:



QUESTIONS AND EXERCISES

Create and test the following bash script with the numbers 12 and 5 as arguments:

```
#!/bin/bash

a=$1
b=$2
p=$(($a*$b))
echo "The product of $a and $b = $p"
```

TEJ4M

QUESTIONS AND EXERCISES

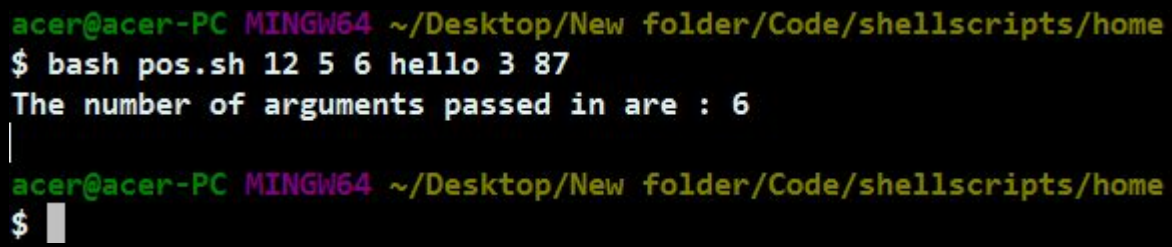
Create and test the following bash script. Explain what's stored in $@.:

```
#!/bin/bash

echo "The arguments passed in are :
$@"
```

QUESTIONS AND EXERCISES

Create a bash script to do the following:



QUESTIONS AND EXERCISES

Create a bash script to print the numbers from 0 to 10 using a for loop.

QUESTIONS AND EXERCISES

```
# Initializing the cc
ctr=10

# Using a while loop
while [ $ctr -gt 0 ]
do
    echo $ctr
    ((ctr--))
done

# Printing "Bash Scri
echo "Bash Script!"
```

Write, test and explain this bash script:

QUESTIONS AND EXERCISES

Write a bash script to ask a user for a number and then create those many files i.e if they enter 3 then create files file1.dat, file2.dat and file3..dat.

QUESTIONS AND EXERCISES

Write a bash script to ask a user for a password and only if they get it right will you print out the contents of the env command.

QUESTIONS AND EXERCISES

Create a bash script that uses a for loop to go through all the arguments passed to it and prints them out with a number listing their order. It will also print out the name of the bash script as well as the list of arguments. The following is an example of the input and out:

```
~$ bash s.sh x y z
s.sh
x y z
1 x
2 y
3 z
```

```bash
#!/bin/bash

# Question 1: Create a bash script that takes 3 numbers as arguments and sums them
sum=$(($1 + $2 + $3))
echo "The sum is: $sum"

# Question 2: Accept arguments and perform some operations
# Example: ./script.sh 10 20 30
echo "You passed $# arguments: $@"
echo "Script name: $0"

# Question 3: Script that takes 2 arguments (12 and 5) and shows operations
a=$1
b=$2
echo "Sum: $(($a + $b))"
echo "Difference: $(($a - $b))"
echo "Product: $(($a * $b))"
echo "Quotient: $(($a / $b))"
echo "Remainder: $(($a % $b))"

# Question 4: Explain $@
# $@ holds all the arguments passed to the script as a list (individually quoted)
echo "All arguments: $@"
echo "Total arguments: $#"

# Question 5: Simple for loop from 0 to 10
for i in {0..10}; do
  echo $i
done

# Question 6: Script that loops through string words
for word in Linux Scripting Is Fun; do
  echo "$word"
done

# Question 7: Create N files based on user input
read -p "Enter how many files to create: " count
for ((i = 1; i <= count; i++)); do
  touch "file${i}.dat"
  echo "Created file${i}.dat"
done

# Question 8: Password check to reveal env variables
read -sp "Enter password: " pwd
echo ""
if [ "$pwd" == "secret123" ]; then
  env
else
  echo "Incorrect password."
fi

# Question 9: For loop to display argument order and values
echo "Script name: $0"
echo "Arguments: $@"
i=1
for arg in "$@"; do
  echo "Argument $i: $arg"
  ((i++))
done
```