

Shell Scripting

Shell Script:

1. (output to terminal) Write a script to print:
 - a. "Welcome to Intelligrape"

```
#!/bin/bash  
echo "Welcome to Intelligrape"
```

```
jay@Jay-Patel:Documents (master)$ vim jay1.sh  
jay@Jay-Patel:Documents (master)$ chmod jay1  
jay123.txt jay1.sh  
jay@Jay-Patel:Documents (master)$ chmod u+x jay1.sh  
jay@Jay-Patel:Documents (master)$ ./jay1.sh  
Welcome to Intelligrape  
jay@Jay-Patel:Documents (master)$
```

- b. <username>@<hostname>:<your present working directory>

```
#!/bin/bash  
  
#echo "Welcome to Intelligrape"  
  
echo "$(whoami)@$(hostname):$(pwd)"  
~
```

```
jay@Jay-Patel:Documents (master)$ vim jay1.sh  
jay@Jay-Patel:Documents (master)$ ./jay1.sh  
jay@Jay-Patel:/home/jay/Documents  
jay@Jay-Patel:Documents (master)$
```

- 2 (arguments) Write a script

- a. which takes in two arguments and print those arguments.

```
read -p "Enter Two Arguments : " a b  
echo $a $b  
~
```

```
jay@Jay-Patel:Documents (master)$ vim jay1.sh  
jay@Jay-Patel:Documents (master)$ ./jay1.sh  
Enter Two Arguments : 1 2  
1 2  
jay@Jay-Patel:Documents (master)$
```

- b. which checks the number of arguments passed and if the number is greater than two print ERROR message along with printing the number of arguments.

```
if [ $# -gt 2 ]  
then  
    echo "Error, number of arg >= 2. You have pass $# arg"  
fi
```

```
jay@Jay-Patel:Documents (master)$ vim jay1.sh  
jay@Jay-Patel:Documents (master)$ ./jay1.sh 1 2 3  
Error, number of arg >= 2. You have pass 3 arg  
jay@Jay-Patel:Documents (master)$ ./jay1.sh 1 2
```

3. Continue with the above script

- a. check the two arguments are only integer values and if these are not integers print the proper error on terminal and also log it into a file.

```

if [ $# -gt 2 ]
then
    echo "Error, number of arg >= 2. You have pass $# arg"
fi

if [[ $1 =~ ^[+-]?[0-9]+$ && $2 =~ ^[+-]?[0-9]+$ ]]
then
    echo "Inputs are integer."
else
    echo "Enter valid argument.[Integer type]"
fi

jay@Jay-Patel:Documents (master)$ ./jay1.sh 1 2 | tee log.txt
Inputs are integer.
jay@Jay-Patel:Documents (master)$ cat log.txt
Inputs are integer.
jay@Jay-Patel:Documents (master)$ ./jay1.sh 1 jay | tee log.txt
Enter valid argument.[Integer type]
jay@Jay-Patel:Documents (master)$ cat log.txt
Enter valid argument.[Integer type]
jay@Jay-Patel:Documents (master)$ █

```

- b. perform addition on the two arguments and print result on screen. Use function for this.

```

sum()
{
    echo $(( $1 + $2 ))
}

sum $1 $2
jay@Jay-Patel:Documents (master)$ ./jay1.sh 4 5
9
jay@Jay-Patel:Documents (master)$ █

```

4. Create a calculator using the above script which would perform addition, subtraction, division and multiplication.

- the script should ask user which operation the user wants to perform: +, -, *, /
- if user enters other than "+, -, *, /", print proper message on terminal and keeps on asking for correct input (use while loop to accomplish this).
- Use case statement instead of if.

```
#!/bin/bash

read -p "Enter the first number : " var1
read -p "Enter the second number : " var2
#echo "$var1 $var2"
echo -e "choose the operation\n 1.additon\n 2.subtract\n 3.multiply\n 4.divide\n"
read -p "Enter your option : " var3
case $var3 in
    1)      echo "ans=$((var1 + var2))"
            ;;
    2)      echo "ans=$((var1 - var2))"
            ;;
    3)      echo "ans=$((var1 * var2))"
            ;;
    4)      echo "ans=$((var1 / var2))"
            ;;
esac
```

```
jay@Jay-Patel:~ $ ./script2.sh
Enter the first number : 6
Enter the second number : 7
choose the operation
 1.additon
 2.subtract
 3.multiply
 4.divide

Enter your option : 3
ans=42
jay@Jay-Patel:~ $ ./script2.sh
Enter the first number : 4
Enter the second number : 2
choose the operation
 1.additon
 2.subtract
 3.multiply
 4.divide

Enter your option : 4
ans=2
jay@Jay-Patel:~ $ ./script2.sh
Enter the first number : 77
Enter the second number : 34
choose the operation
 1.additon
 2.subtract
 3.multiply
 4.divide

Enter your option : 1
ans=111
jay@Jay-Patel:~ $
```

5. Write proper help documentation and print it with -h for above script.

```
Description

    This is Manual Calculator
It performs basic arithmetic calculator
Additon -- +
Subtraction -- -
Multiplication -- *
Division -- /

Step 1: Enter the two number.
Step 2: Enter the operator.
~
~
~
~
```

```
jay@Jay-Patel:~ $ ./script2.sh -h
jay@Jay-Patel:~ $ ./script2.sh -h
jay@Jay-Patel:~ $ ./script2.sh -h
```

```
Description

    This is Manual Calculator
It performs basic arithmetic calculator
Additon -- +
Subtraction -- -
Multiplication -- *
Division -- /

Step 1: Enter the two number.
Step 2: Enter the operator.
manual-calc (END)
```

6. Create a script which takes input of "/etc/passwd" file and find out and print the sum of uids and gids. The script should tell which sum of greater.

```
#!/bin/bash
awk -F: '{a+= $3; b+= $4}END {print "Sum of UIDs : "a" " \nSum of GUIDs : "b" "; if(a>b){print a} else {print b}}' /etc/passwd
```

```
jay@Jay-Patel:~ $ ./jay123.sh
Sum of UIDs : 70964
Sum of GUIDs : 463499
463499
jay@Jay-Patel:~ $
```

7. A directory contains files and sub-directories. Move files to destination1 and directories to destination2

```
for i in `ls`
do
    if [[ "$i" != "files" && "$i" != "dir" && "$i" != "jay123.sh" ]]
    then
        if [ -f $i ]
        then
            mv $i files/$i
        fi
        if [ -d $i ]
        then
            mv $i dir/$i
        fi
    fi
done
```

```

jay@Jay-Patel:test $ ll
total 40
drwxrwxr-x  9 jay jay 4096 Apr 14 15:09 ./
drwxr-xr-x 54 jay jay 4096 Apr 14 15:09 ../
drwxrwxr-x  2 jay jay 4096 Apr 14 15:08 dir/
drwxrwxr-x  2 jay jay 4096 Apr 14 15:08 files/
-rw-rw-r--  1 jay jay    0 Apr 14 15:09 isfjg
-rw-rw-r--  1 jay jay    0 Apr 14 15:09 isfjgg
-rw-rw-r--  1 jay jay    0 Apr 14 15:09 isfjggfes
-rw-rw-r--  1 jay jay    0 Apr 14 15:09 isfjggfessg
-rwxrwxr-x  1 jay jay  328 Apr 14 15:08 jay123.sh*
drwxrwxr-x  2 jay jay 4096 Apr 14 15:09 jjf/
drwxrwxr-x  2 jay jay 4096 Apr 14 15:09 jjffd/
drwxrwxr-x  2 jay jay 4096 Apr 14 15:09 jjffdsf/
drwxrwxr-x  2 jay jay 4096 Apr 14 15:09 jjffs/
drwxrwxr-x  2 jay jay 4096 Apr 14 15:09 jjfsdf/
jay@Jay-Patel:test $ ./jay123.sh
jay@Jay-Patel:test $ ll
total 20
drwxrwxr-x  4 jay jay 4096 Apr 14 15:09 ./
drwxr-xr-x 54 jay jay 4096 Apr 14 15:09 ../
drwxrwxr-x  7 jay jay 4096 Apr 14 15:09 dir/
drwxrwxr-x  2 jay jay 4096 Apr 14 15:09 files/
-rwxrwxr-x  1 jay jay  328 Apr 14 15:08 jay123.sh*
jay@Jay-Patel:test $ █

```

```

jay@Jay-Patel:test $ cd files
jay@Jay-Patel:files $ ll
total 8
drwxrwxr-x 2 jay jay 4096 Apr 14 15:09 ./
drwxrwxr-x 4 jay jay 4096 Apr 14 15:09 ../
-rw-rw-r-- 1 jay jay    0 Apr 14 15:09 isfjg
-rw-rw-r-- 1 jay jay    0 Apr 14 15:09 isfjgg
-rw-rw-r-- 1 jay jay    0 Apr 14 15:09 isfjggfes
-rw-rw-r-- 1 jay jay    0 Apr 14 15:09 isfjggfessg
jay@Jay-Patel:files $ cd ../dir
jay@Jay-Patel:dir $ ll
total 28
drwxrwxr-x 7 jay jay 4096 Apr 14 15:09 ./
drwxrwxr-x 4 jay jay 4096 Apr 14 15:09 ../
drwxrwxr-x 2 jay jay 4096 Apr 14 15:09 jjf/
drwxrwxr-x 2 jay jay 4096 Apr 14 15:09 jjffd/
drwxrwxr-x 2 jay jay 4096 Apr 14 15:09 jjffdsf/
drwxrwxr-x 2 jay jay 4096 Apr 14 15:09 jjffs/
drwxrwxr-x 2 jay jay 4096 Apr 14 15:09 jjfsdf/
jay@Jay-Patel:dir $ █

```


8. Create a script which take three arguments, append first argument to every line in a file and second argument to the end of every line of the same file..

```
Ashu
jay
ashish
ujjwal
shivani
divya
apoorva
nitish
~
~
~
~
~
~
```

```
sed -i "s/~/$1/; s/$/$2/" $3
~
~
~
```

```
ay@Jay-Patel:test $ ./jay123.sh jungle house names.txt
ay@Jay-Patel:test $ cat names.txt
ungle Ashuhouse
ungle jayhouse
ungle ashishhouse
ungle ujjwalhouse
ungle shivanihouse
ungle divyahouse
ungle apoorvahouse
ungle nitishhouse
ay@Jay-Patel:test $
```


