Shell scripting

CREATING VARIABLES

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ cat variable2.sh
#!/bin/bash
#author: ram kumar
#pourpose: learning variables
#usage: ./variable.sh

var1=10
var2="Hello world"
hostname=`hostname`
date=`date`
#the following variable definitions are allowed
1value=100
false@linux=false

echo "var1=$var1"
echo "var2=$var2"
echo "hostname=$hostname"
echo "date=$date"
echo "1value=$1value"
echo "false@linux=$false@linux"


USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./variable2.sh
./variable2.sh: line 11: 1value=100: command not found
./variable2.sh: line 12: false@linux=false: command not found
var1=10
var2=Hello world
hostname=G6DR0F3
date=Tue Jan 28 12:53:13 IST 2025
1value=value
false@linux=@linux
```

CREATING VARIABLES WHICH READ INPUT FROM USER

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ cat input_variable.sh
#!/bin/bash
#author: ram kumar
#pourpose: learning variables
#usage: ./input_variable.sh


echo "This is first value i got=$1"
echo "Second value=$2"


USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./input_variable.sh
This is first value i got=
Second value=

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./input_variable.sh 10 20
This is first value i got=10
Second value=20
```

IF CASE WITH SINGLE BRACKET

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi if_square_bracket.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cat if_square_bracket.sh
#!/bin/sh

file=variable2.sh
if [ -f $file ]; then
        echo "file exists"
fi



USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./if_square_bracket.sh
file exists
```

-f is used to search the file

IF DOUBLE QUOTE [HERE WE ARE MORE BETTER ADVANCED] HERE WE CAN USE ONLY IN BASH

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cp if_square_bracket.sh if_double_square_bracket.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi if_double_square_bracket.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./if_double_square_bracket.sh "file with space .sh"
file exists

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./if_double_square_bracket.sh "file with space not .sh"
Does not exists

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$  cat if_double_square_bracket.sh
#!/bin/sh

file=variable3.sh

if [[ -f $1 ]]; then
        echo "file exists"
else
        echo "Does not exists"
fi
```

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi for.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./for.sh
I like to eat apple
I like to eat banana
I like to eat cherry
I like to eat mango
fruit ate 0 i like apple
fruit ate 1 i don't like banana
fruit ate 2 i like cherry
fruit ate 3 i don't like mango

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi for.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cat f
file with space .sh  for.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cat for.sh
#!/bin/bash
#author: ram kumar
#pourpose: learning foe loop
#usage: ./for.sh

fruits=("apple" "banana" "cherry" "mango")
for fruit in "${fruits[@]}"; do
        echo "I like to eat $fruit"
done


fruits=("apple" "banana" "cherry" "mango")
for i in "${!fruits[@]}"; do
        if [ `expr ${i} % 2 ` == 0 ]; then
                echo "fruit ate $i i like ${fruits[${i}]}"
        else
                echo "fruit ate $i i don't like ${fruits[${i}]} "
        fi
done


USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$
```

29-01-2025

$ linuxshell.sh

```shell
#!/bin/bash
#author: dev
#pourpose: learning fillinux commandse
#usage: ./linuxshell.sh

echo "iam $USERNAME and my home directory is $HOME"
echo "My current working directory is $PWD or `pwd` "
echo "`whoami`"
echo "`date`"

ls

command=`ls -ltr /etc`
echo "$command"
eval $command
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
-rw-r--r-- 1 USTR+290398 4096  1744 Nov 25 12:13 docx2txt.config
-rw-r--r-- 1 USTR+290398 4096  4351 Nov 25 12:13 DIR_COLORS
-rw-r--r-- 1 USTR+290398 4096  2497 Nov 25 12:13 bash.bashrc
-rw-r--r-- 1 USTR+290398 4096   623 Nov 25 12:13 bash.bash_logout
-rw-r--r-- 1 USTR+290398 4096  3886 Nov 25 12:27 package-versions.txt
drwxr-xr-x 1 USTR+290398 4096     0 Jan 14 09:46 pki
drwxr-xr-x 1 USTR+290398 4096     0 Jan 14 09:46 pkcs11
drwxr-xr-x 1 USTR+290398 4096     0 Jan 14 09:46 ssh
drwxr-xr-x 1 USTR+290398 4096     0 Jan 14 09:49 profile.d
-rw-r--r-- 1 USTR+290398 4096   366 Jan 14 09:50 install-options.txt
```

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi while.sh
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./while.sh
Please enter the number:
10
Multiplication table of 10
Mult of 10 * 1 = 10
Mult of 10 * 2 = 20
Mult of 10 * 3 = 30
Mult of 10 * 4 = 40
Mult of 10 * 5 = 50
Mult of 10 * 6 = 60
Mult of 10 * 7 = 70
Mult of 10 * 8 = 80
Mult of 10 * 9 = 90
Mult of 10 * 10 = 100

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi while.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi while.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cat while.sh
#!/bin/bash
#author: ram kumar
#pourpose: learning while loop
#usage: ./while.sh

echo "Please enter the number: "
read -r var1
echo Multiplication table of $var1
counter=1
while [ $counter != 11 ]
do
        echo "Mult of $var1 * $counter = `expr $var1 \* $counter`"
        counter=`expr $counter + 1`
done
```

-r used to read from user

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi batterscore.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./batterscore.sh
Enter the score of batsman
33
Good

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./batterscore.sh
Enter the score of batsman
89
Excellent

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./batterscore.sh
Enter the score of batsman
2
Bad

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cat batterscore.sh
#!/bin/sh
#Author: DEv
#Pourpose: learning if else

echo "Enter the score of batsman"
read score
if [ $score -gt 40 ]; then
        echo "Excellent"
elif [ $score -lt 20 ]; then
        echo "Bad"
else
        echo "Good"
fi
```

For loop varieties

```
#for i in 1 2 3 4 5
#do
#       echo "$i"
#done


#for i in {1..5};
#do
#       echo "$i"
#done

#for i in $(seq 1 5);
#do
#       echo "$i"
#done



for (( i=1; i<10; i++));
do
        echo "$i"
done

~
~
```

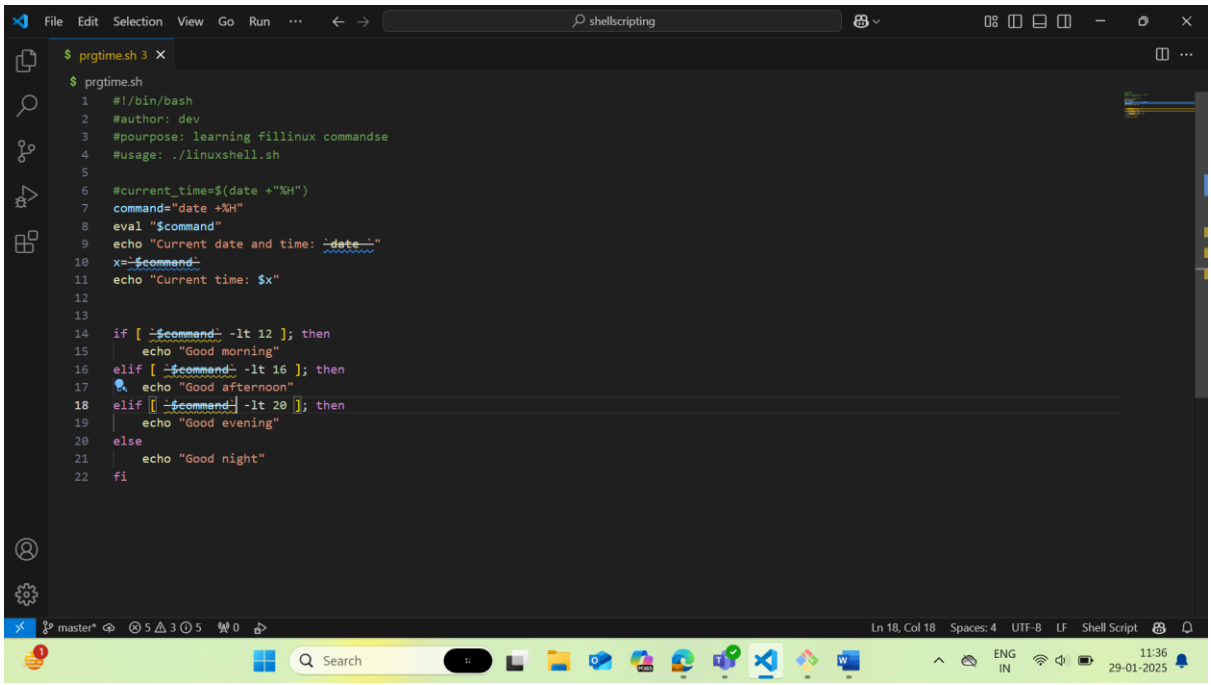
UNTIL

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./until.sh
please enter the ip addressto ping\c
yahoo.com

Pinging yahoo.com [74.6.231.21] with 32 bytes of data:
Reply from 74.6.231.21: bytes=32 time=275ms TTL=52
Reply from 74.6.231.21: bytes=32 time=344ms TTL=52
Reply from 74.6.231.21: bytes=32 time=277ms TTL=52
Reply from 74.6.231.21: bytes=32 time=277ms TTL=52

Ping statistics for 74.6.231.21:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 275ms, Maximum = 344ms, Average = 293ms
Host in yahoo.com is up

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ cat until.sh
#!/bin/bash

echo "please enter the ip addressto ping\c"
read -r ip
until ping $ip
do
        echo "Host in $ip is down"
        sleep 1
done
echo "Host in $ip is up"
```

```
MINGW64:/c/Users/290398/shellscripting
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ vi until.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./until.sh
please enter the ip addressto ping\c
127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Host in 127.0.0.1 is up

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting (master)
$ ./until.sh
please enter the ip addressto ping\c
150.0.0.1

Pinging 150.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 150.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Host in 150.0.0.1 is down

Pinging 150.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 150.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Host in 150.0.0.1 is down

Pinging 150.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
```

30°C Mostly cloudy Search ENG IN 17:17 28-01-2025

To take hour from the date then x=` date + %h`



To get the value of the variable we need to give the `` then only we get the value

? has error code is or not

learningbackupfile.sh 4

learningbackupfile.sh

```
#!/bin/bash
#author: dev
#pourpose: learning fillinux commandse
#usage: ./linuxshell.sh

function backup {
    echo "Enter the file name"
    read -r file
    if [ -f $file ]; then
        echo "file exists"
        cp $file /tmp/backup.txt
    else
        echo "file does not exist"
    fi


    if [$? -ne 0]; then
        echo "backup failed?? "
    else
        echo "backup success"
    fi
}
```

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
backup success
```

& execute as backgound process(Daemon)

```
echo "All arguments combined togther $*"
echo "No: of arguments $#"
echo "first argument $1"
echo "Expand all command line on sep words $@"
echo "Process id of current process $$"

sleep 400 &
echo "Process id of recently background process $!"
echo "file name of current program$0"
```

PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
./autopopulate.sh

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./autopopulate.sh one "two word" three
All arguments combined togther one two word three
No: of arguments 3
first argument one
Expand all command line on sep words one two word three
Process id of current process 3199
Process id of recently background process 3200
file name of current program./autopopulate.sh
```

Date set

$ learningset.sh

```
#pourpose: learning autopopulate
#usage: ./autopopulate.sh

set `date`
echo "Today is $1"
echo "month is $2"
echo "date is $3"
echo "year is $4"
echo "time is $5"
echo "am/pm is $6"
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
time is 2:26:05
timezone is PM

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./learningset.sh
Today is Wed,
month is Jan
day is 29,
year is 2025
time is 2:27:35
am/pm is PM
```

$ learningset.sh

```
#!/bin/bash
#author: dev
#pourpose: learning autopopulate
#usage: ./autopopulate.sh

set `date`
set -x

echo "Today is $1"
echo "month is $2"
echo "date is $3"
echo "year is $4"
echo "time is $5"
echo "am/pm is $6"
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./learningset.sh
+ echo 'Today is Wed,'
Today is Wed,
+ echo 'month is Jan'
month is Jan
+ echo 'date is 29,'
date is 29,
+ echo 'year is 2025'
year is 2025
+ echo 'time is 2:30:04'
time is 2:30:04
+ echo 'am/pm is PM'
```

$ regularexpression.sh

```
#!/bin/bash
#author: dev
#pourpose: learning RE
#usage: ./rexp.sh

numString="123456789"
if [[ $numString =~ ^1 ]]; then
    echo "The string starts with 1"
fi
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./regularexpression.sh
The string starts with 1

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$
```

```
numString="123456789"
charString="abcde32fghi321"

if [[ $numString =~ ^1 ]]; then
    echo "$numString The string starts with 1"
fi


if [[ $numString =~ ^[1.7] ]]; then
    echo "$numString The string starts with 1 and 7"
fi


if [[ $numString =~ ^1.*8 ]]; then
    echo "$numString The string starts with 1 and 8"
fi

if [[ $charString =~ ^[A-Za-z]+$ ]]; then
    echo "$charString The string contains only alphabets"
fi
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
$ ./regularexpression.sh
123456789 The string starts with 1
123456789 The string starts with 1 and 7
123456789 The string starts with 1 and 8

USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$
```

String upper

$ stringupper.sh

```
#!/bin/bash
#author: dev
#pourpose: string upper case conversion
#usage: ./stringupper.sh

echo "Enter a string"
read -r string1

string2=$string1

echo "The string is: ${string2^^}"

# stringupper=$(echo "$string1" | tr '[:lower:]' '[:upper:]')

# echo "The string in upper case is: $stringupper"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
USTR+290398@G6DR0F3 MINGW64 ~/shellscripting
$ ./stringupper.sh
Enter a string
fds rfed
The string is: FDS RFED
The string in upper case is: FDS RFED
```

Pallindrome

$ pallindrome.sh

```
#!/bin/bash
#author: dev
#pourpose: string pallindrome
#usage: ./palindrome.sh


echo "Enter a string"
read -r str1

for (( i=${#str1}; i>=0; i-- ))
do
    str2+=${str1:$i:1}
done
echo "The reverse of the string is: $str2"
```

Count no: of words

```
$ count_words.sh
#!/bin/bash
#author: dev
#pourpose: no: of words in a string
#usage: ./count_words.sh

echo "Enter a string"
read -r str1

count_words=$( echo "$str1" | wc -w )
echo "The no: of words in the string is: $count_words"
```

Replace word

```
$ replaceaword.sh
#!/bin/bash
#author: dev
#pourpose: replace a word in a file
#usage: ./replacewords.sh

echo "Enter the word to be replaced and the word to be replaced with"
read -r var1 var2

sed -i 's/'$var1'/'$var2'/g' file1.txt
```