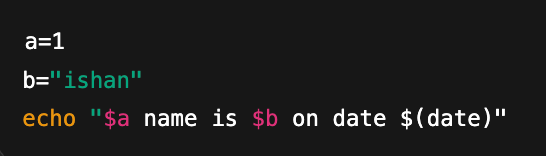
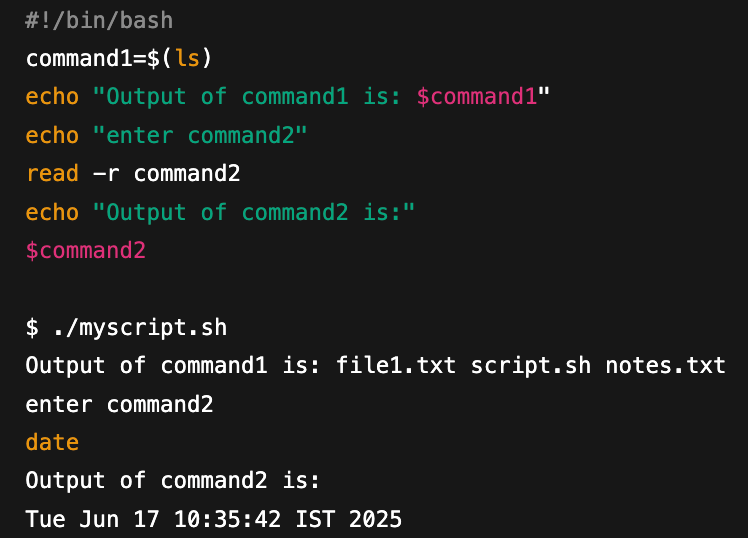
**Variables**

Use command in script as $(command).



**Take input from user**

1. Use read <variable>



1. Take arguments with ./script.sh $1 $2

#!/bin/bash

username=$1

password=$2

sudo useradd "$username"

echo "$username:$password" | sudo chpasswd

echo "User $username created with password"

$ ./create\_user.sh ishan Ishan@123

User ishan created with password

**If Else**

#!/bin/bash

read -p "Enter a number: " num

if [[ "$num" -gt 0 ]]; then

echo "Number is positive."

elif [[ "$num" -lt 0 ]]; then

echo "Number is negative."

else

echo "Number is zero."

fi

**For Loop**

#!/bin/bash

for (( i=1; i<=5; i++ ))

do

echo "Number: $i"

done

$ ./for\_c\_style.sh

Number: 1

Number: 2

Number: 3

Number: 4

Number: 5

**While Loop**

#!/bin/bash

i=1

while (( $i -le 10 && $((i % 2)) -eq 0 ))

do

echo "Even number: $i"

((i++))

done

echo "Loop ended at i=$i"

$ ./while\_and.sh

Loop ended at i=1

1. Script to print only errors from a remote log

- curl http://example.com/logs/app.log | grep "error" > errors.log

1. Numbers divisible by 2,3 but not 15

for (( i=1; i<=100; i++ ))

do

if (( (i % 3 == 0 || i % 5 == 0) && i % 15 != 0 )); then

echo "$i"

fi

done

1. Find number of s in mississippi or any other word

echo -n "Enter a word: "

read word

echo -n "Enter a character to count: "

read char

count=$(echo "$word" | grep -o "$char" | wc -l)

echo "$count"

*# -o tells grep to output only the matching characters, not the full line.  
# In a file,* ***wc*** *count no. of words (wc -w), no. of lines (wc -l). In our case, grep print each letter ‘s’ in a new line*

1. How to debug a shell script?

- Put “set -x” at the top of script. The script will be executed in debug mode.