BASIC SHELL SCRIPT

```
#
# My first shell script
#
clear
echo "Knowledge is Power"
~
~
~
~
~
~
```

```
Knowledge is Power
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % vi first
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % chmod +x first
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % ./first
```

```
# Script to print user information who currently login, current date and time
#
clear
echo "Hello $USER"
echo "Today is \c";date
echo "Number of user login : \c";who | wc -l
echo "Calendar"
cal
exit 0
~
~
```

```
Hello bhagyashree
Today is \c
Mon Feb  6 18:28:26 IST 2023
Number of user login : \c
       2
Calendar
   February 2023
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

(base) bhagyashree@Bhagyashrees-MacBook-Air shell %
```

VARIABLES

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % a=4
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % b=8
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $a
4
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $b
8
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo$[$a+$b]
zsh: command not found: echo12
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $[$a+$b]
12
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % vech=""
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % echo $vech

(base) bhagyashree@Bhagyashrees-MacBook-Air shell %
```

Backslash – used for execution (for arithmetic operations)

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % expr 3 + 4
7
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % echo `expr 3 + 4`
7
(base) bhagyashree@Bhagyashrees-MacBook-Air shell %
```

Exit status

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ls -l foobar
ls: foobar: No such file or directory
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % touch foobar
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ls -l fooblar
ls: fooblar: No such file or directory
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ls -l foobar
-rw-r--r--  1 bhagyashree  staff  0 Jan  9 00:04 foobar
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
0
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ls -l fooblar
ls: fooblar: No such file or directory
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
1
```

Reading content from keyboard

```
#Script to read your name from keyboard

echo "Your first name please:"
read fname
echo "Hello $fname, Lets be friend!"
~
~
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % vi sayH
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % chmod +x sayH
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % ./sayH
Your first name please:
bhagyashree
Hello bhagyashree, Lets be friend!
(base) bhagyashree@Bhagyashrees-MacBook-Air shell %
```

Finding file using *

```
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % find DTL -type f -name "*.sh"
DTL/comparison.sh
DTL/variable.sh
DTL/z2/first.sh
DTL/xp.sh
DTL/hello-world.sh
DTL/conditional.sh
DTL/backup.sh
DTL/x.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % find DTL -size 50c
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % find DTL -size -50c
DTL/stdout.txt
DTL/z2/first.sh
DTL/foobar
DTL/stderr.txt
DTL/stdoutandstderr.txt
DTL/file1
DTL/x.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % find DTL -size +700M
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % find DTL -mtime 1
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % find DTL -atime -1
DTL
DTL/z3
DTL/z4
DTL/stdout.txt
DTL/comparison.sh
DTL/z5
DTL/variable.sh
DTL/z2
DTL/z2/first.sh
```

Command line arguments

```
# Script that demos, command line args

echo "Total number of command line argument are $#"
echo "$0 is the scipt name"
echo "$1 is first argument"
echo "$2 is second argument"
echo "All of them are :- $* or $@"
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % vi demo
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % chmod +x demo
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % ./demo Hello World
Total number of command line argument are 2
./demo is the scipt name
Hello is first argument
World is second argument
All of them are :- Hello World or Hello World
```

Changing standard I/O

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % cat>sname
vivek
ashish
zebra
babu
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % sort<sname>sorted_names
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % cat sorted_names
ashish
babu
vivek
zebra
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % tr "[a-z]" "[A-Z]"<sname>cap_names
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % cat cap_names
VIVEK
ASHISH
ZEBRA
BABU
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % sort > new_sorted_names < sname
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % cat cap_names
VIVEK
ASHISH
ZEBRA
BABU
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % cat new_sorted_names
ashish
babu
vivek
zebra
(base) bhagyashree@Bhagyashrees-MacBook-Air shell %
```

Usage of pipes (output of first is given as input for second)

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % ls | more
cap_names
demo
ginfo
new_sorted_names
sayH
sname
sorted_names
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % who|sort
bhagyashree console  Feb  5 11:22
bhagyashree ttys000  Feb  6 18:21
(base) bhagyashree@Bhagyashrees-MacBook-Air shell %
```

Basic calculator program in linux:

```
(base) bhagyashree@Bhagyashrees-MacBook-Air shell % bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
5+2
7
5-2
3
5/2
2
5>2
1
5<2
0
5>12
0
5==10
0
5!=2
1
5==5
1
12<2
0
```

If-then statement

```
cat>trmif
#
# Script to test rm command and exit staus
#
if rm $1
then
echo "$1 file deleted"
fi
~
```

If else command

```
#!/bin/sh
# Script to see whether argument is positive or negative
#
if [ $# -eq 0 ]
then echo "$0 : You must give/supply one integer"
exit 1
fi

if test $1 -gt 0
then
echo "$1 number is positive"
else
echo "$1 number is negative"
fi
~
~
```

Nested if else statements

```
osch=0

echo "1. Unix (Sun Os)"
echo "2. Linux (Red Hat)"
echo -n "Select your os choice [1 or 2]?"
read osch

if [ $osch -eq 1 ] ; then

        echo "You Pick up Unix (Sun Os)"

else #### nested if i.e. if within if #####

        if [ $osch -eq 2 ] ; then
                echo "You Pick up Linux (Red Hat)"
        else
                echo "What you don't like Unix/Linux OS."
        fi
fi
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air ~ % cd DTL
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi nestedif.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x nestedif.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./nestedif.sh
1. Unix (Sun Os)
2. Linux (Red Hat)
-n Select your os choice [1 or 2]?
1
You Pick up Unix (Sun Os)
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./nestedif.sh
1. Unix (Sun Os)
2. Linux (Red Hat)
-n Select your os choice [1 or 2]?
2
You Pick up Linux (Red Hat)
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./nestedif.sh
1. Unix (Sun Os)
2. Linux (Red Hat)
-n Select your os choice [1 or 2]?
3
What you don't like Unix/Linux OS.
```

Nested if else statements

```
if [ $1 -gt 0 ] ; then
        echo "$1 is positive"
elif [ $1 -lt 0 ]
then
        echo "$1 is negative"
elif [ $1 -eq 0 ]
then
        echo "$1 is zero"
else
        echo "Opps! $1 is not number, give number"
fi
~
~
~
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi nestedif.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi elf.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x elf.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./elf.sh 1
1 is positive
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./elf.sh -2
-2 is negative
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./elf.sh 0
0 is zero
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./elf.sh a
./elf.sh: line 1: [: a: integer expression expected
./elf.sh: line 3: [: a: integer expression expected
./elf.sh: line 6: [: a: integer expression expected
Opps! a is not number, give number
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

For loop

```
for((i=0;i<=5;i++))
do
        echo "Welcome $i times"
done
~
~
~
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi for.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x for.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./for.sh
Welcome 0 times
Welcome 1 times
Welcome 2 times
Welcome 3 times
Welcome 4 times
Welcome 5 times
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Nested for loop

```
for ((i=1;i<=5;i++))
do
        for((j=1;j<=5;j++))
        do
                echo -n "$i"
        done
        echo " " #### print the new line ###
done
~
~
~
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi for2.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x for2.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./for2.sh
-n 1
-n 1
-n 1
-n 1
-n 1

-n 2
-n 2
-n 2
-n 2
-n 2

-n 3
-n 3
-n 3
-n 3
-n 3

-n 4
-n 4
-n 4
-n 4
-n 4

-n 5
-n 5
-n 5
-n 5
-n 5
```

While loop

```
n=$1
i=1
while [ $i -le 10 ]
do
        echo "$n * $i = `expr $i \* $n`"
        i=`expr $i + 1`
done
~
~
~
```

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi while.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x while
chmod: while: No such file or directory
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x while.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./while.sh 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % vi while.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Using case instead of if else

```
rental = $1
case $rental in
        "car") echo "For $rental Rs. 20 pen km";;
        "van") echo "For $rental Rs. 10 per km";;
        "jeep") echo "For $rental Rs. 5 per km";;
        "bicycle") echo "For $rental 20 paisa per km";;
        *) echo "Sorry, I cannot get a $rental for you";;
esac
~
~
~
```

Debug shell script

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % cat>dsh1.sh
#
# Script to show debug of shell
#

tot=`expr $1 + $2`
echo $tot
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % chmod +x dsh1.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ./dsh1.sh 4 5
9
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % sh -x dsh1.sh 4 5
++ expr 4 + 5
+ tot=9
+ echo 9
9
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % sh -v dsh1.sh 4 5
#
# Script to show debug of shell
#

tot=`expr $1 + $2`
expr $1 + $2
echo $tot
9
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Array

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % array=(a b c)
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo ${array[*]}
a b c
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo ${array[2])
braceparam> }
zsh: bad substitution
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo ${array[2]}
}
zsh: parse error near `}'
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo ${array[2]}
b
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo ${#array[*]}
3
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % for chr in ${array[@]}; do
for> echo $chr
for> done
a
b
c
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % for chr in ${array[@]}; do
for> done
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % for chr in ${array[*]}; do
for> echo $chr
for> done
a
b
c
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Comparison of numbers

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % a=1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % b=2
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [$a -lt $b]
zsh: bad pattern: [1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [ $a -lt $b ]
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
0
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [ $a -gt $b ]
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [ $a -eq $b ]
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [ $a -ne $b ]
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
0
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Comparison of strings

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [ "apples" = "oranges" ]
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % str1="apples"
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % str2="oranges"
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % [ str1 = str2 ]
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $?
1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Shell expansion

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo a{b,d,e}c
abc adc aec
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo {A..Z}
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % mkdir z{1..10}
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ls
hello-world.sh  z1              z3              z6              z9
x.sh            z10             z4              z7
xp.sh           z2              z5              z8
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % a=4
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % b=8
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $((a=a+b))
12
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $a
12
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ((a++))
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $a
13
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % b=8
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % unset b
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % ((b--))
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $b
-1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % a=1
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % let a++
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % echo $a
2
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```

Special parameters

```
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % nano x.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % bash x.sh {A..Z}
D O G C A T
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % nano x.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % bash x.sh {A..Z}

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Represents all args as sing string

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All args are represented as an array

26 is the total number of arguments

hB current option flag

3121 is the PID of the shell

is the PID of the last executed BG command

x.sh is the script name

echo final arg of last executed Foreground command

0 Represents the exit code of the last command

(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % nano x.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % bash x.sh {A..Z}
D O G C A T
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % nano xp.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % bash xp.sh {A..Z}
```

```
echo final arg of last executed Foreground command

0 Represents the exit code of the last command

(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % nano x.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % bash x.sh {A..Z}
D O G C A T
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % nano xp.sh
(base) bhagyashree@Bhagyashrees-MacBook-Air DTL % bash xp.sh {A..Z}

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Represents all args as sing string

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All args are represented as an array

26 is the total number of arguments

hB current option flag

3126 is the PID of the shell

is the PID of the last executed BG command

xp.sh is the script name

echo final arg of last executed Foreground command

0 Represents the exit code of the last command

(base) bhagyashree@Bhagyashrees-MacBook-Air DTL %
```