# Shell Scripts

[1. Read 2](#_Toc363977438)

[2. echo 2](#_Toc363977439)

[3. more 2](#_Toc363977440)

[4. less 2](#_Toc363977441)

[5. uniq 2](#_Toc363977442)

[6. sort 3](#_Toc363977443)

[7. gzip 3](#_Toc363977444)

[8. bzip2 3](#_Toc363977445)

[9. rm 3](#_Toc363977446)

[10. mkdir 3](#_Toc363977447)

[11. rpm 3](#_Toc363977448)

[12. ifconfig 4](#_Toc363977449)

[13. shutdown 4](#_Toc363977450)

[14. String Concatenation 4](#_Toc363977451)

[15. String Comparison 4](#_Toc363977452)

[16. Nested If Loop 5](#_Toc363977453)

[17. For Loop 5](#_Toc363977454)

[18. While Loop 5](#_Toc363977455)

[19. Case 6](#_Toc363977456)

[20. Arithmetic Expression 6](#_Toc363977457)

[21. Function 6](#_Toc363977458)

[22. File Present or not 7](#_Toc363977459)

[23. Directory Present or not 7](#_Toc363977460)

## Read

* read –t 3 text
  + Terminal waits for 3 seconds to get user input. If time elapsed, it exists.
* Read –s text
  + User inputs are not displayed

## echo

* echo –n “Enter some text > “
  + The cursor is placed in the same line.
* echo –e “Hello world\t\t Linux\n”
  + enable the backslash-escaped characters

## more

* Displays one page at a time
* No backwards scrolling
* Space bar -> One screen up
* Enter -> One line up
* q -> quit
* Example
  + ls –l /usr/bin | more

## less

* More sophisticated solutions than more
* j/k -> movement one line
* f/b -> Movement by one window at a time
* h -> help window
* q -> quit

## uniq

* Removes the duplicate entries
* Compares only if entries are adjacent to each other

## sort

* sort names.txt
  + Arranges the contents of the file in Ascending order.
* sort –r names.txt
  + Arranges the contents in Descending order

## gzip

* gzip test.txt
  + To create a \*.gz compressed file
* Gzip –d text.txt.gz
  + To uncompress a \*.gz file

## bzip2

* bzip2 test.txt
  + To create a \*.bz2 compressed file
* bzip2 –d test.txt.bz2
  + To uncompress a \*.bz2 file

## rm

* rm –i filename.txt
  + Get confirmation before removing the file

## mkdir

* mkdir –p dir1/dir2/dir3/dir4
  + To create the nested directory with one command

## rpm

* rpm –ivh httpd-2.23.rpm
  + To install apache using rpm.
* rpm –uvh httpd-2.23.rpm
  + To upgrade apache using rpm.
* rpm –ev httpd
  + To uninstall/remove apache using rpm.

## ifconfig

* View Network Settings of an Ethernet Adapter
  + ifconfig eth0
* Display Details of All interfaces Including Disabled Interfaces
  + ifconfig -a
* Disable an Interface
  + ifconfig eth0 down
* Enable an Interface
  + ifconfig eth0 up
* Assign ip-address to an Interface
  + ifconfig eth0 192.168.2.2
  + ifconfig eth0 netmask 255.255.255.0
  + ifconfig eth0 192.168.2.2 netmask 255.255.255.0

## shutdown

* Shutdown immediately
  + shutdown –h now
* shutdown after 10 minutes
  + shutdown –h +10
* Reboot the System
  + shutdown –r now

## String Concatenation

#!/bin/bash

x="Hello"

y=" World"

z=$x$y

echo $z

## String Comparison

#!/bin/bash

echo -n "Enter text compare -> "

read text

if [ $text = "yes" ]; then

echo "Strings are equal"

else

echo "Strings are not equal"

fi

## Nested If Loop

#!/bin/bash

echo -n "Enter a number between 1 and 3 inclusive > "

read character

if [ "$character" = "1" ]; then

echo "You entered one."

elif [ "$character" = "2" ]; then

echo "You entered two."

elif [ "$character" = "3" ]; then

echo "You entered three."

else

echo "You did not enter a number"

echo "between 1 and 3."

fi

## For Loop

#!/bin/bash

for i in word1 word2 word3; do

echo $i

done

## While Loop

#!/bin/bash

number=0

while [ $number -le 10 ]; do

echo "Number = $number"

number=$((number+1))

done

## Case

#!/bin/bash

echo -n "Type a digit or a letter> "

read character

case $character in

[a-z] | [A-Z] ) echo "you typed the letter $character"

;;

[0-9] ) echo "you tyed the digit $character"

;;

\* ) echo "you did not type a letter"

esac

## Arithmetic Expression

#!/bin/bash

first\_num=0

second\_num=0

echo -n "Enter the first number --> "

read first\_num

echo -n "Enter the second number -> "

read second\_num

echo "first number + second number = $((first\_num + second\_num))"

echo "first number - second number = $((first\_num - second\_num))"

echo "first number \* second number = $((first\_num \* second\_num))"

echo "first number / second number = $((first\_num / second\_num))"

echo "first number % second number = $((first\_num % second\_num))"

echo "first number raised to the"

echo "power of the second number = $((first\_num \*\* second\_num))"

## Function

#!/bin/bash

function hello

{

echo "Hello"

}

function world

{

echo " World"

}

#Execute the Function, Just by calling the function

hello

world

## File Present or not

#!/bin/bash

if [ $# -ne 1 ]; then

echo "Usage - $0 file-name"

exit 1

fi

if [ -f $1 ]; then

echo "$1 file exist"

else

echo "sorry $1 file doesn't exist"

fi

## Directory Present or not

#!/bin/bash

if [ $# -ne 1 ]; then

echo "Please Enter the Directory Name"

exit 1

fi

if [ -d $1 ]; then

echo "Directory present"

else

echo "Directory Not present"

fi

echo, 6

function, 5