**Some examples of popular conditions**

1. if [[ -f /tmp/file ]]; then do-something => do something if the file /tmp/file exists
2. if [[ ! -f /tmp/file ]]; then do-something => do something if the file /tmp/file does not exists
3. if [[ -n ${variable} ]]; then do-something => do something if the ${variable} is not empty
4. if [[ !-n ${variable} ]]; then do-something => do something if the ${variable} is empty
5. if [[ -z ${variable} ]]; then do-something => do something if the ${variable} is empty

**Applying above conditions to make script handle errors**

* The ifthenrcdemo.sh file will have a script

#!/bin/bash

mkdir temps

mkdir\_rc=$?

# Test if the directory creation is success

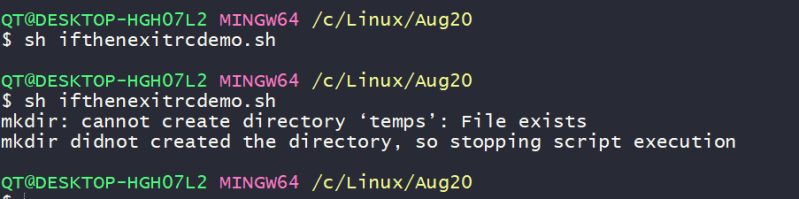
if [[ ${mkdir\_rc} -ne 0 ]]; then

echo "mkdir didnot created the directory, so stopping script execution"

exit 1

fi

touch temps/tempfile.txt

* Lets execute this script with a positive & negative condition 
* So now lets adopt if-then-else

#!/bin/bash

FILE=randomfile.txt

# check if the file exists

if [[ ! -f ${FILE} ]]; then

echo "File mentioned as ${FILE} doesn't exist so exiting"

exit 1

else

echo "Printing contents of file at ${FILE}"

cat ${FILE}

fi

* Lets write one more if-then-else script which accepts parameters (positional parameters)

#!/bin/bash

### Usage ./ifthenelsedemo2.sh <path-of-file>

file\_name=$1

# user might enter empty values

if [[ -z ${file\_name} ]]; then

echo "Incorrect usage: ./ifthenelsedemo2.sh <filename>"

exit 1

fi

if [[ ! -f ${file\_name} ]]; then

echo "Please correct the file path and re-execute."

exit 1

else

echo "Contents of the file are"

cat ${file\_name}

fi

**Checking my arguments**

* Lets try to write a very simple shell scrip which creates a specified file in the specified directory with specified content
* So argument list should be

./createfile.sh <directory\_path> <filename> <file contents>

* The script will be as shown below

#!/bin/bash

####################################################################

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 01-Sep-2020

# Description: This script demonstrates basic user inputs

# Usage: ./createfile.sh <directory-name> <file-name> <file-content>

#####################################################################

# We need three arguments, so checking if the arguments passed count

# is 3 or not

if [[ $# -ne 3 ]]; then

echo "Incorrect number of arguments passed"

echo "Usage: ./createfile.sh <directory-name> <file-name> <file-content>"

exit 1

fi

# create parameters with argument values

directory\_name=$1

file\_name=$2

file\_content=$3

# check if the directory exists, if it doesnot exist create directory

if [[ ! -d ${directory\_name} ]]; then

mkdir ${directory\_name} || { echo "Cannot create directory"; exit 1; }

fi

# lets create absolute file path

abs\_file\_path=${directory\_name}/${file\_name}

# Try to create a file if the file doesnot exist

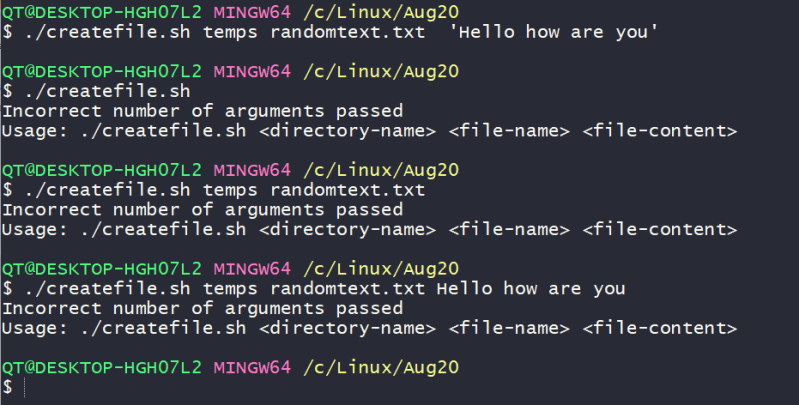
if [[ ! -f ${abs\_file\_path} ]]; then

touch ${abs\_file\_path} || { echo "Cannot create a file"; exit 1; }

fi

# Since file is created or present add the contents to it

echo ${file\_content} > ${abs\_file\_path}

* Now execute the script 
* In command line usage when arguments are in **<>** they are required arguments and if the arguments are in **[]** they are optional.

**Dealing with y/n options in interactive scripts**

* When we ask input from the user, user might enter many possibilities for yes (YES,yes,YeS,y,Y) and same for no.
* How can write a script which is case sensitive to yes value

#!/bin/bash

read -p "Do you like linux? " reply

if [[ ${reply,,} = 'y' ]] || [[ ${reply^^} == 'YES' ]]

echo "Great, Continue your learning journey"

exit 0

fi