**Positional arguments in shell scripts**

* The arguments which are passed to the shell script after the filepath of shell becomes positional arguments

./myscript.sh 12 13

* The above example consider

$0 = ./myscript.sh

$1 = 12

$2 = 13

* Let’s develop a shell script which takes name and location as positional argument 1 and 2 and prints the message

#!/bin/bash

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

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 26-Aug-2020

# Description: This script demonstrates positional arguments

# Usage: ./positionalargdemo.sh <name> <location>

# 1 positional argument is considered as name

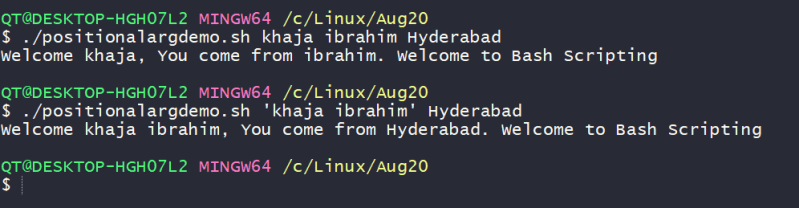
# 2 positional argument is considered as location

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

name=$1

location=$2

echo "Welcome ${name}, You come from ${location}. Welcome to Bash Scripting"

* Execution Result Preview

**Parameters & arguments**

* Let’s understand the terminology of parameter & arguments. They are slightly different
* Argument is something which you pass to a script.
* **What you define in a script to hold argument value is parameter**
* Let’s write a simple script to demonstrate the difference

#!/bin/bash

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

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 25-Aug-2020

# Description: This is shell script demonstrating parameters and arguments

# Usage: ./argumentparameter.sh <arg-1> <arg-2>

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

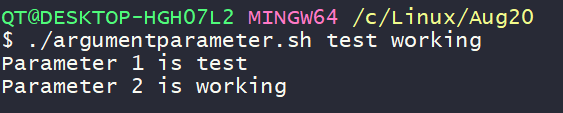
# parameter is used to define argument value

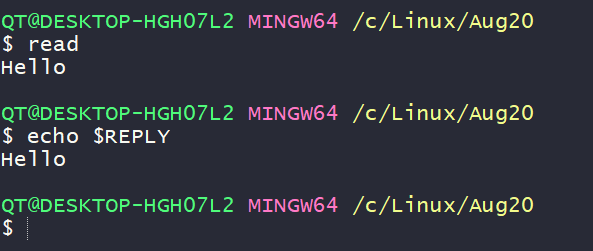
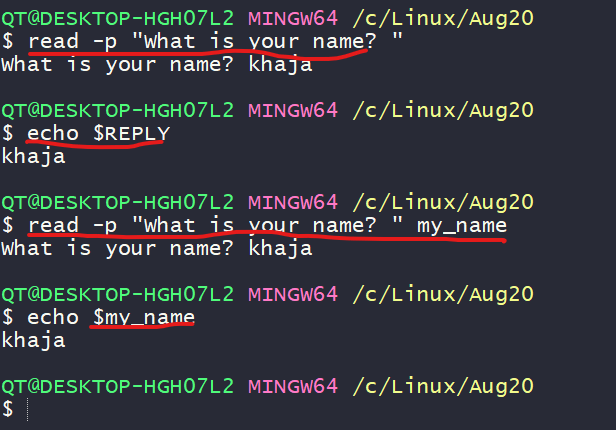
parameter\_1=$1

parameter\_2=$2

echo "Parameter 1 is ${parameter\_1}"

echo "Parameter 2 is ${parameter\_2}"



* The scripts which we have developed so far are non-interactive scripts, because once you pass the values the execution happens
* Interactive scripts take inputs from users while executing
* Now let’s look at a simple *read* command which reads the input from the user and stores in a variable called as REPLY 
* Now let’s execute one more simple variation of the *read* command 
* Lets build some interactive script asking questions to the user

#!/bin/bash

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

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 26-Aug-2020

# Description: This is shell script demonstrating interactive scripts

# Usage: ./interactivescriptdemo.sh

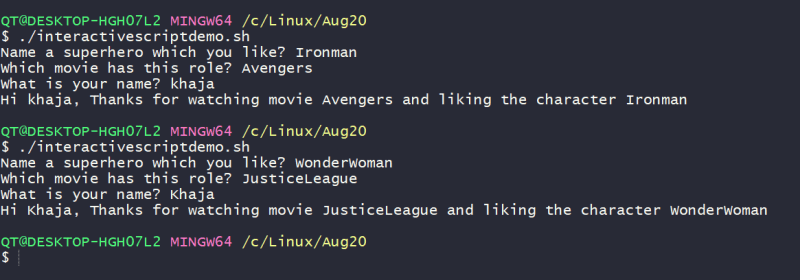
#####################################################################

read -p "Name a superhero which you like? " super\_hero

read -p "Which movie has this role? " movie\_super\_hero

read -p "What is your name? " name

echo "Hi ${name}, Thanks for watching movie ${movie\_super\_hero} and liking the character ${super\_hero}"

* Now lets execute the script 
* Let’s try to come up with script design which acts as a calculator by taking two values and operation as input?
  + Script will be *calculator.sh*
  + operations that will be supported are *add,sub,mul,div*
  + It should take two numbers to perform math.