**What is a variable**

* Variable is a standard building block in scripting languages, They allow you to store information, so that we can retrieve/change multiple times during the execution of script.
* Assigning the value variable <variable\_name>=<variable\_value>

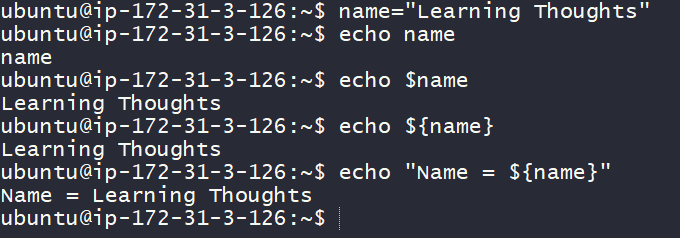
name="Learning Thoughts"

* Referencing the variable ${variable\_name} or $variable\_name

echo $name

echo ${name}

echo "Name == ${name}"



* Now lets try to create a simple shell script

#!/bin/bash <This is called Shebang>

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

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 25-Aug-2020 <Script header>

# Description: This is shell script demonstrating variables

# Usage: ./variabledemo.sh

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

name="Khaja Ibrahim"

org\_name="Learning Thoughts"

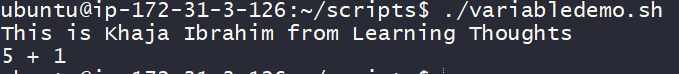
# Here we are interpolating the variables

echo "This is ${name} from ${org\_name}"

# All the variables are treated as text by linux until and unless you are explicit

number=5

echo $number + 1

* Now execute the script 
* Now shell script is treating numbers as text, if we want shell script to consider variable number as integer, we need to use and expression

$(( ${number} + 1 ))

* Now considering the script to be

#!/bin/bash

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

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 25-Aug-2020

# Description: This is shell script demonstrating variables

# Usage: ./variabledemo.sh

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

name="Khaja Ibrahim"

org\_name="Learning Thoughts"

# Here we are interpolating the variables

echo "This is ${name} from ${org\_name}"

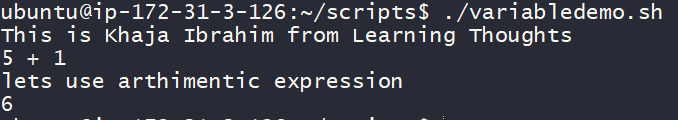
# All the variables are treated as text by linux until and unless you are explicit

number=5

echo $number + 1

echo "lets use arthimetic expression"

echo $(( ${number} + 1 ))



* In shell scripting we following naming conventions, for
  + variables: lowercase with underscores
* my\_number=6
* message="how are you"
  + Constants: UPPERCASE
* LOCATION="/home"
  + Do not create variables with names which already have some meaning PATH, USER, LANG, SHELL, HOME & so on. If the Purpose is similar try to prefix SCRIPT\_, for example define as shown below.
* SCRIPT\_PATH over PATH
* SCRIPT\_USER over USER
* So far, we have been dealing with static scripts, so let’s introduce the concept of *user inputs* in shell scripting

**Basic Inputs**

* At a very basic level, everything you enter on the command line after the script can be used as input

./<script>.sh Hello

* Lets create a script called as *printname.sh* with the following script

#!/bin/bash

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

# Author: Shaik Khaja Ibrahim

# Version: v1.0.0

# Date: 25-Aug-2020

# Description: This script demonstrates basic user inputs

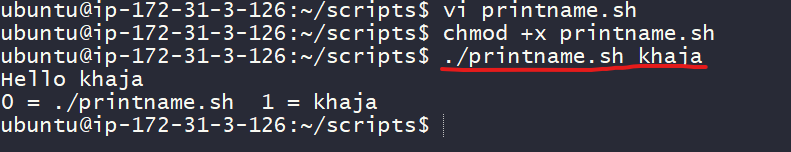
# Usage: ./printname.sh <pass name>

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

name=${1}

echo "Hello ${name}"

echo "0 = ${0} 1 = ${1}"

* Now execute the script 
* Here ${1} stands for the *first positional argument*
* Generally we will have two kinds of arguments
  + positional arguments
* cp 1.txt 2.txt # 1.txt & 2.txt are two positional arguments
  + named arguments
* ping --count 4 google.com
* # google.com is a positional argument
* # where as 4 is a named argument as it value for a name called as count