How to write a Shell Script?

What is a Shell script?

A Shell Script *is* ***series of commands*** *written* ***in plain text file****.*

So, How to create a text file ?

We have two ways to create a text file :

1) Use the "**touch**" command ex: **touch file_name**
2) Use the "**Vi editor**" ex: **vi file_name**

The difference between these two ways is that:

- "**Touch**" Command : creates an initially empty text file only.
- "**vi editor**" : creates a new text file (if the file name was used for the first time) and opens a text file (if the file name was used before, which means that the file was created previously) and enables you to ***write*** and/or ***edit*** in this file.

So in order to write a script in a flexible way, use the Vi editor so that you can open the file (if it was previously created) or create a file and open it too (if it wasn't created before).

How to use the "Vi editor" ?

Remember that the Vi editor enables you to ***write*** in a text file and also ***edit*** it, so we can say it's somehow like any text editor "ex : Microsoft Word".

It has two modes :

- Command mode
- Text input mode

The command mode : is the mode that makes the vi editor ready to take your order "command" . in this mode you have to tell the vi editor what do you want it to do. Ex : open a file, replace a text with other text …. Etc.

Ms. Eman Al-Moaili **Ms. Maram Al-Shablan** **Ms. Shiyam Al-Almaei**

The text input mode : is the mode that you can write in, inside the text file.

Notice that in order to activate this mode you have to give the editor a "command" in the "command mode" which tells the vi editor that you want to write some text right now. This command is either **i** or **a**.

Once you are done with writing your text or if you want to edit it or if you want to exit the file you need to go back to the "command mode" by hitting the "**ESC**" button in the keyboard.

1- Syntax : **vi** File_Name ex: **vi Script1**
2- Once you hit "**Enter**" the vi editor will open the "Script1" text file
3- You will notice that each line in this file is started with a blue " **~** " , that disappears if you write any text in it.
4- By default , once you opened the file with the vi editor you will be in the ***command mode***. To enter the text input mode press either **i** or **a**.
5- Now you can type your text which can be a normal free text or a comment or a whole shell script.
6- If you want to ***edit*** (modify) the written text you need first to go to the command mode by pressing "**ESC**"
7- If you're done writing the script in this file you have to save it and exit it by several ways either by typing **ZZ** or **:wq** in the command mode

Writing the shell script using the vi editor is the first step in writing scripts.

Secondly, you have to make this file an **executable** file . by default any text file doesn't have the execute permission. So you have to grant it this permission by using the "**chmod** command".

Ex: **chmod 755 Script1**

Now the file can be executed. So all you have to do now is to execute it (in other words run it).

To ***run*** any text file that already has the execute permission you have different ways:

Sh file_name

Bash file_name

./ file_name

Ex : **bash Script1**

note: in third way there is no space between ./ and the file name

Sending variables to your shell script:

You can write a script and pass some variables to it. You can pass them while (calling) ***running*** the script .

Suppose you wrote a script that adds two numbers and you want to pass them to you script, for example the numbers are **3** and **5**.

[run command] [script name] [variable1 variable2 variable3 ….. variable n]

Bash Script1 3 5

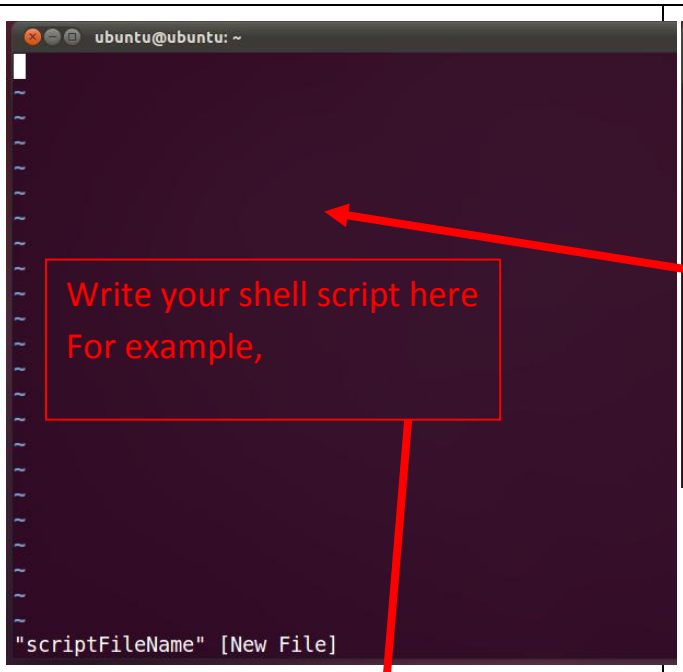
How to use the passed variables in your script?

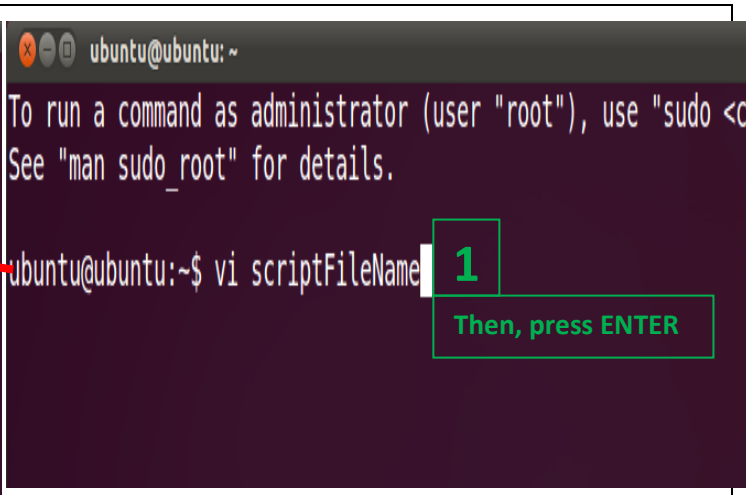
Notice that you can send 3 types of variables :

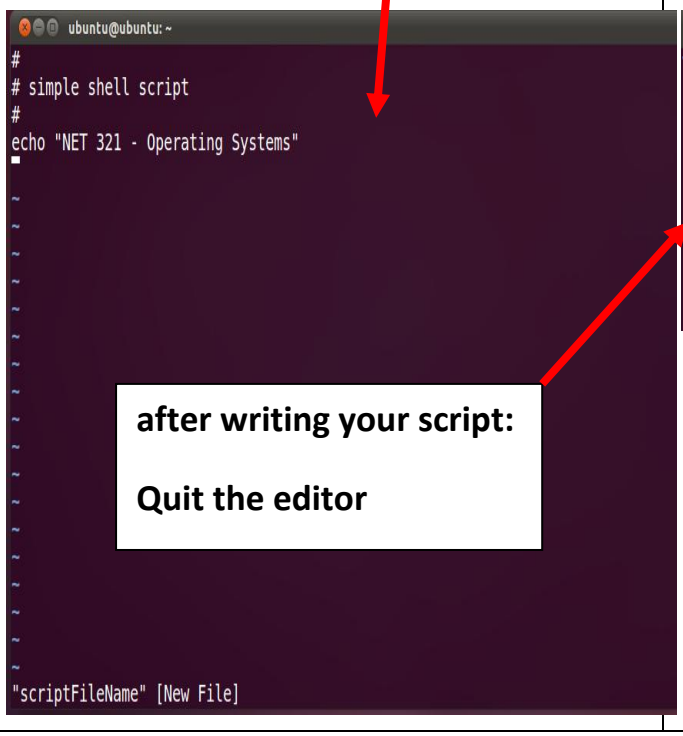
1- Direct value either a ***number*** like (**3**) or a ***character*** like (string or special character like **+**, *****, **%** … etc)
2- Shell variables that contain a value like any user defined variable (**$x, $y**, … etc)
3- File or directory names

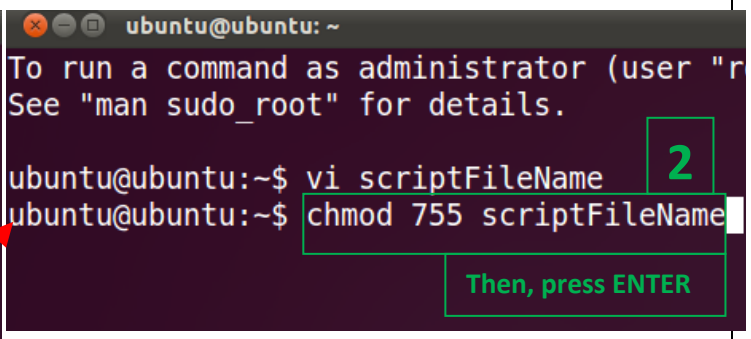
Inside the script you can name the (first passed variable) whatever it was with (**$1**). The second variable is (**$2**), the third is (**$3**) … and so on.

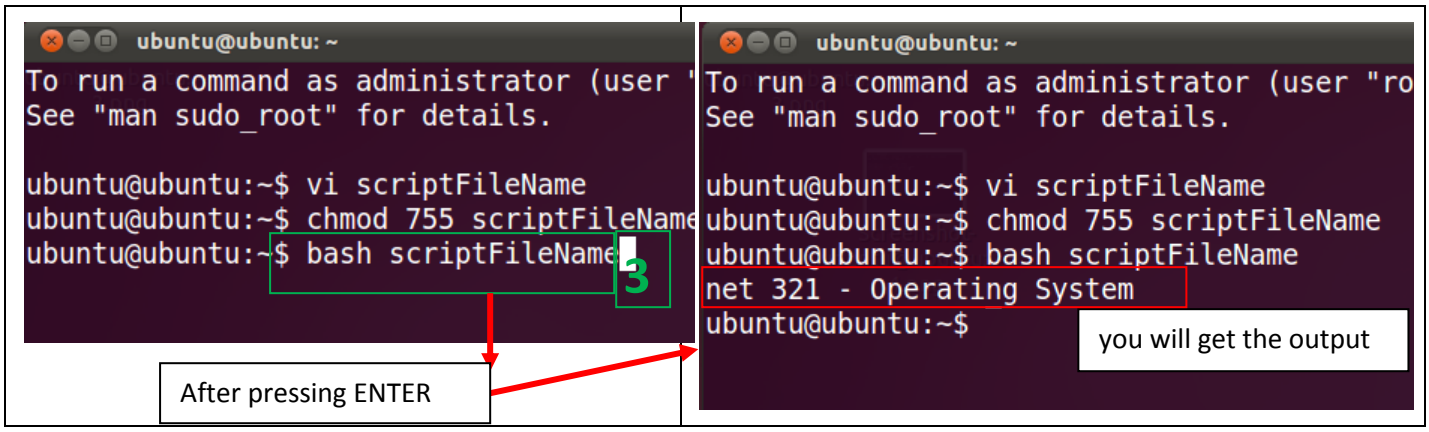
So instead of using the passed variables in the script with their exact name. use the special shell variables ($1, $2, $3 …. Etc) correspondingly.


ubuntu@ubuntu: ~
Write your shell script here
For example,
"scriptFileName" [New File]


ubuntu@ubuntu: ~
To run a command as administrator (user "root"), use "sudo <c
See "man sudo_root" for details.
ubuntu@ubuntu:~$ vi scriptFileName
1
Then, press ENTER


ubuntu@ubuntu: ~
#
simple shell script
#
echo "NET 321 - Operating Systems"
after writing your script:
Quit the editor
"scriptFileName" [New File]


ubuntu@ubuntu: ~
To run a command as administrator (user "r
See "man sudo_root" for details.
ubuntu@ubuntu:~$ vi scriptFileName
2
ubuntu@ubuntu:~$ chmod 755 scriptFileName
Then, press ENTER



NOTE:

After saving the script you will find the script file in the Home folder.