Hands-on Lab: Getting Started with Shell Scripting Variables

Skills Network

Learning Objectives

After completing this lab, you will be able to:

- Create and execute a simple Bash shell script
- Implement the shebang directive in a Bash shell script

Exercise 1 - Create and execute a basic shell script

In this exercise, you will create a simple script which will do the following:

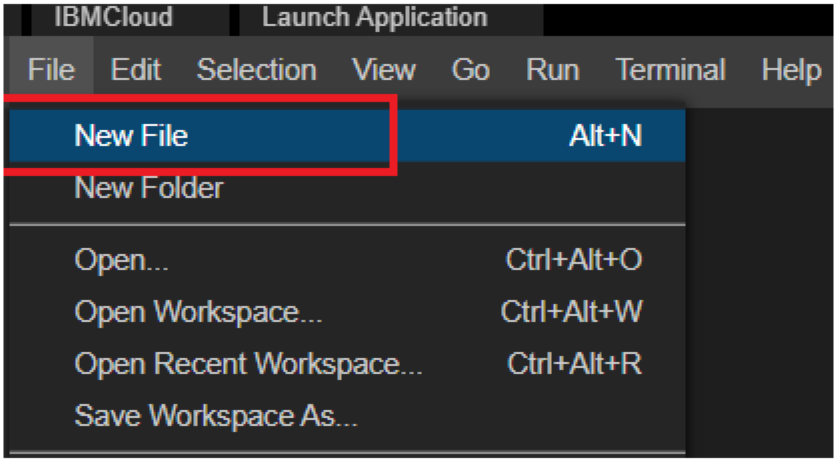
- Accept a user name
- Print a welcome message to the user

You will also add comments to the script, which are lines starting with `#`. Comments are not executed by the shell.

When used appropriately, comments can make a shell script more readable and help in debugging the script.

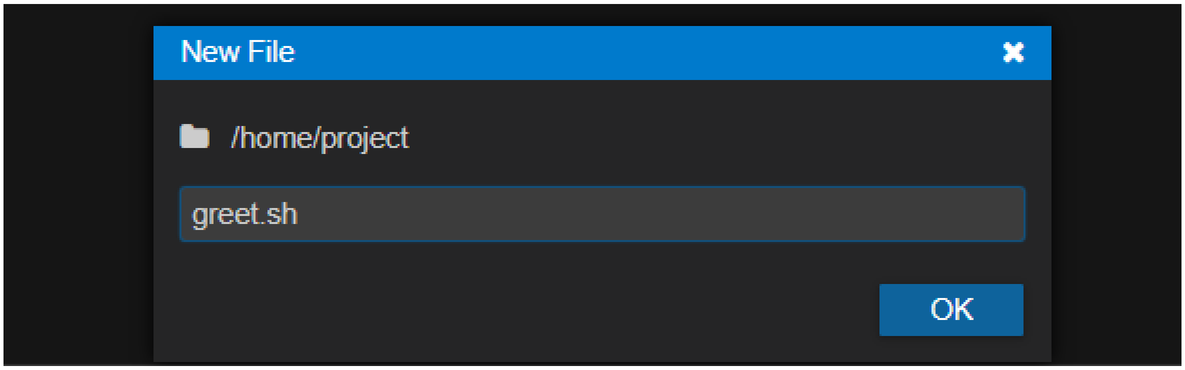
1.1. Create a new script file

Step 1: On the menu on the lab screen, use **File→New File** to create a new file.



!

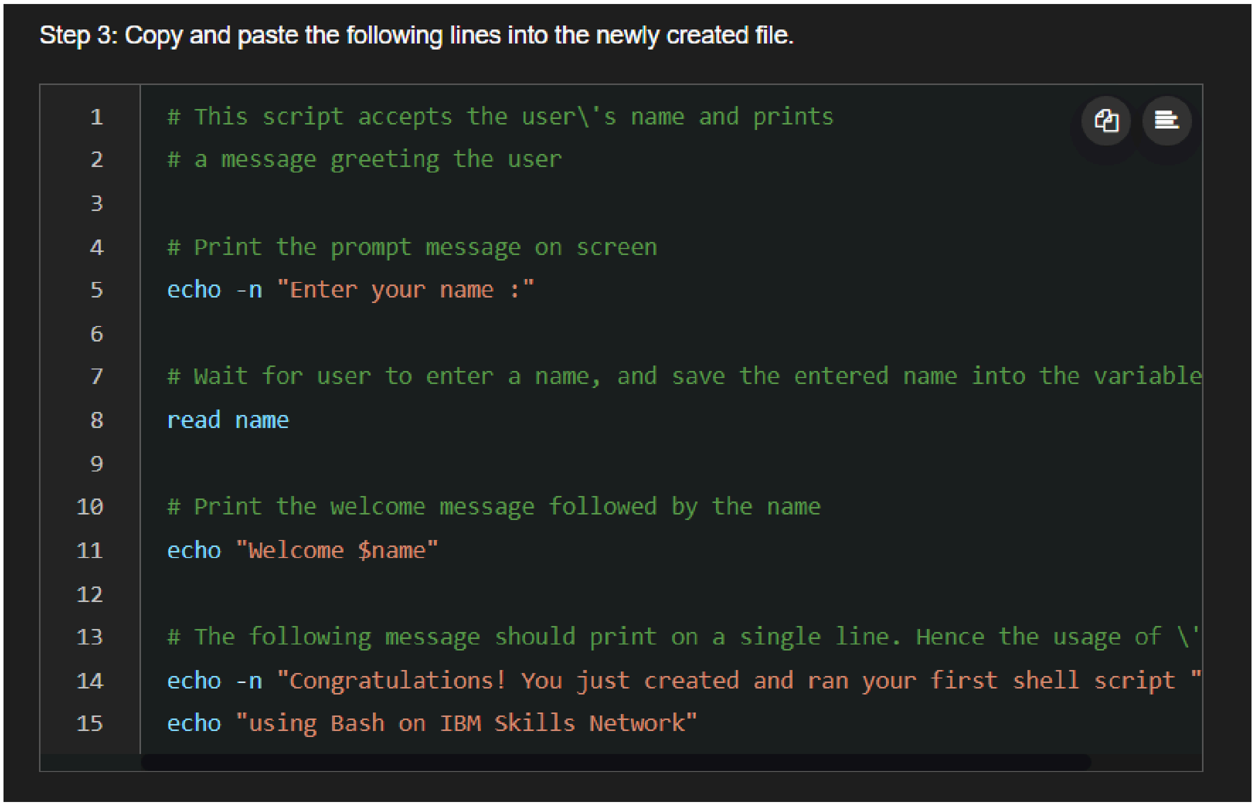
Step 2: Name it as `greet.sh` and click **OK**



/home/project

greet.sh

Step 3: Copy and paste the following lines into the newly created file.



Step 3: Copy and paste the following lines into the newly created file.

```
# This script accepts the user\'s name and prints
# a message greeting the user

# Print the prompt message on screen
echo -n "Enter your name :"

# Wait for user to enter a name, and save the entered name into the variable
read name

# Print the welcome message followed by the name
echo "Welcome $name"

# The following message should print on a single line. Hence the usage of \'
echo -n "Congratulations! You just created and ran your first shell script "
echo "using Bash on IBM Skills Network"
```

1. `# This script accepts the user\'s name and prints`
2. `# a message greeting the user`
3.
4. `# Print the prompt message on screen`
5. `echo -n "Enter your name :"`
6.
7. `# Wait for user to enter a name, and save the entered name into the variable \'name\'`
8. `read name`
9.
10. `# Print the welcome message followed by the name`
11. `echo "Welcome $name"`
12.
13. `# The following message should print on a single line. Hence the usage of \'-n\'`

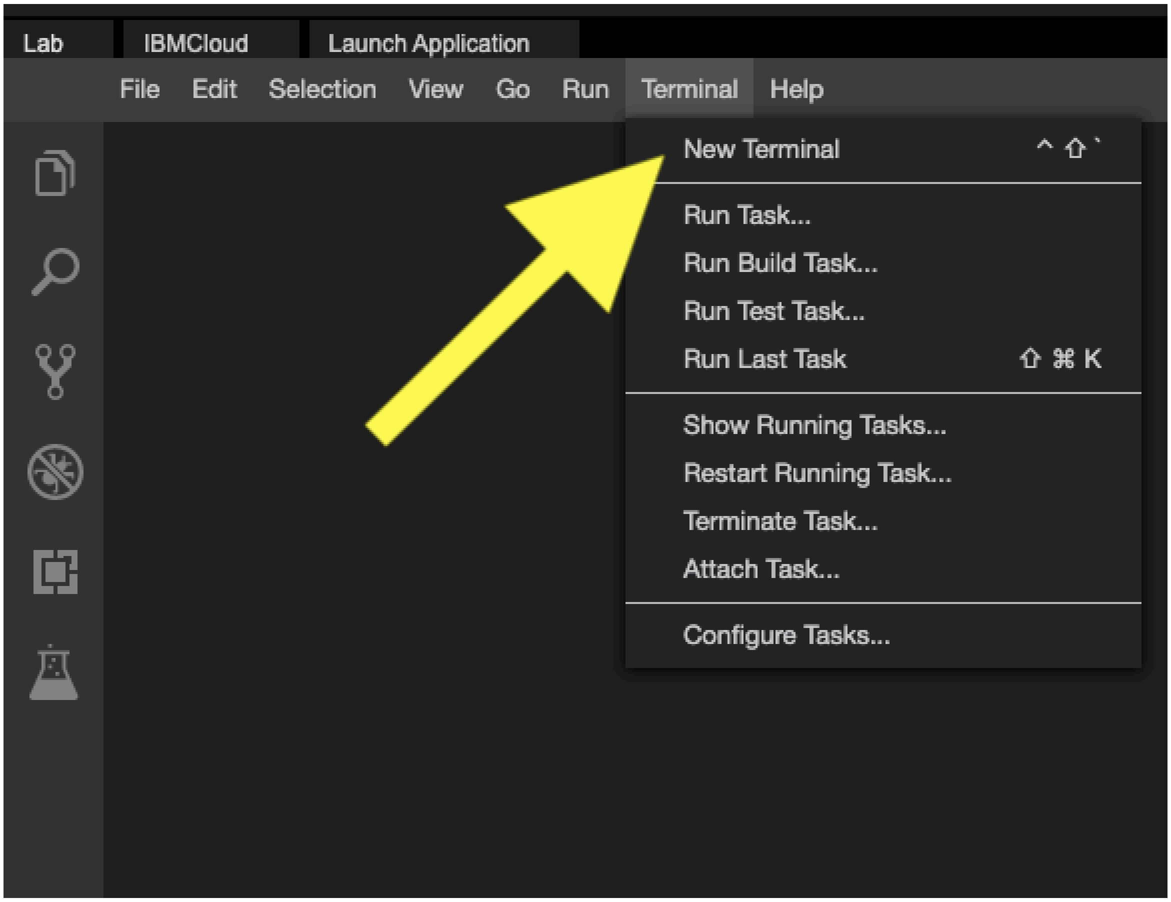
14. `echo -n "Congratulations! You just created and ran your first shell script "`

15. `echo "using Bash on IBM Skills Network"`

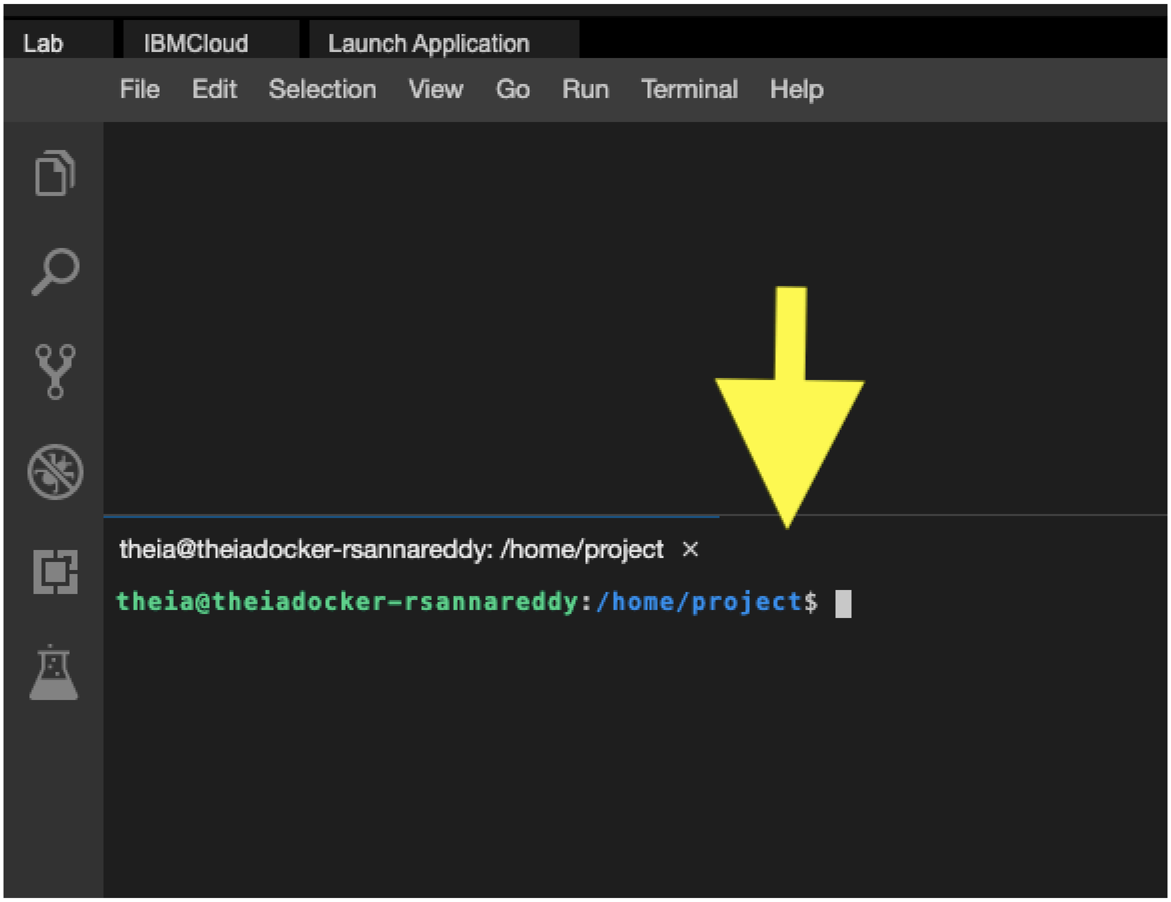
Step 4: Save the file using the **File→Save** menu option.

1.2. Execute the script

Open a new terminal by clicking on the menu bar and selecting **Terminal→New Terminal**, as in the image below.



This will open a new terminal at the bottom of the screen.



```
theia@theiadocker-rsannareddy:/home/project$
```

Run the commands below in the newly opened terminal.

Let's check the permissions for this new file by entering the following:

1. `ls -l greet.sh`

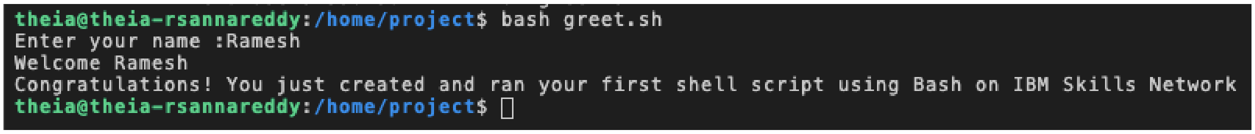
If the file exists and has read permissions, run the following command to execute it:

1. `bash greet.sh`

The message `Enter your name :` appears on screen.

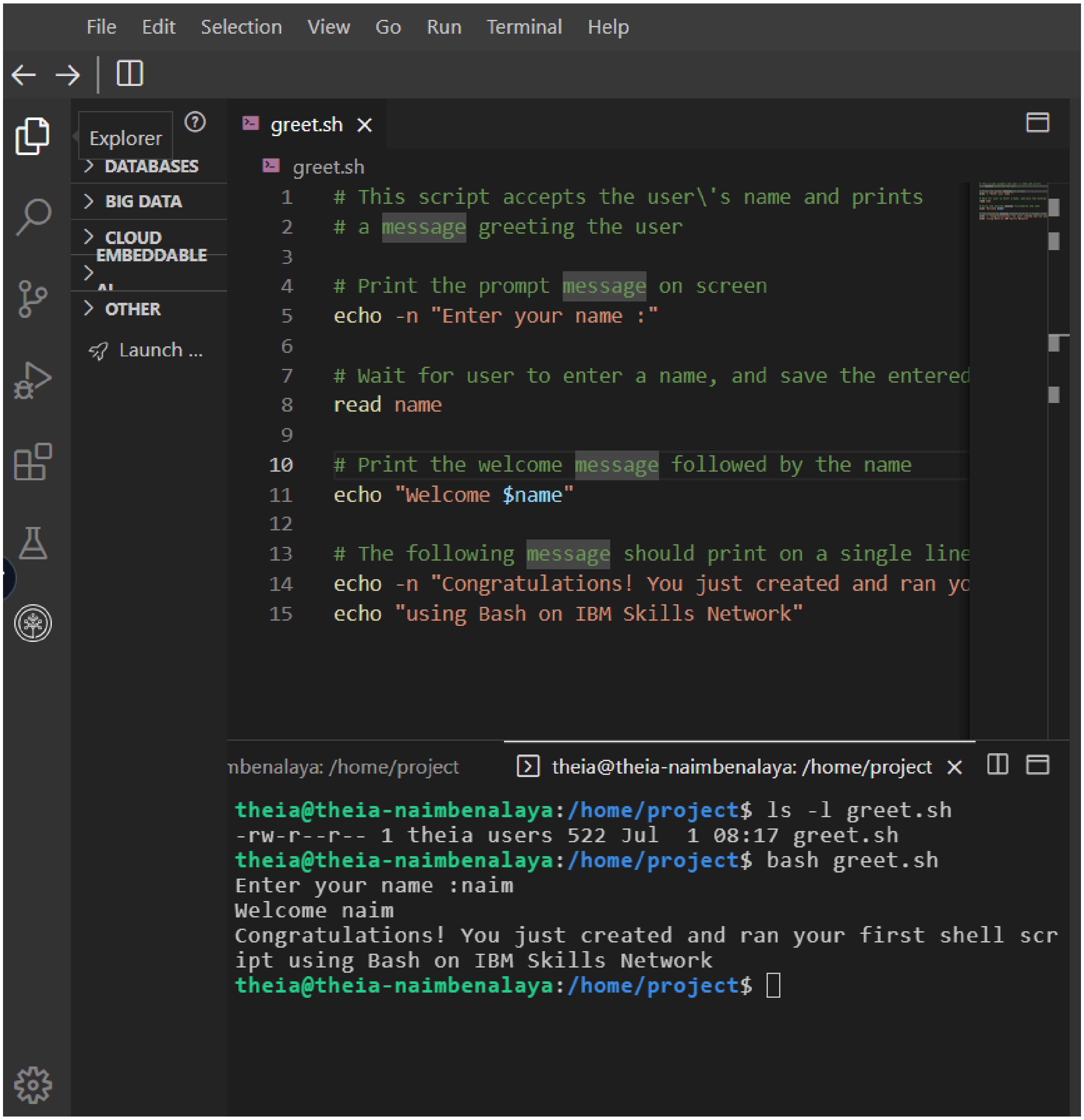
Type your name and press the `Enter` key.

You should now see the welcome messages displayed on screen with your entered name.



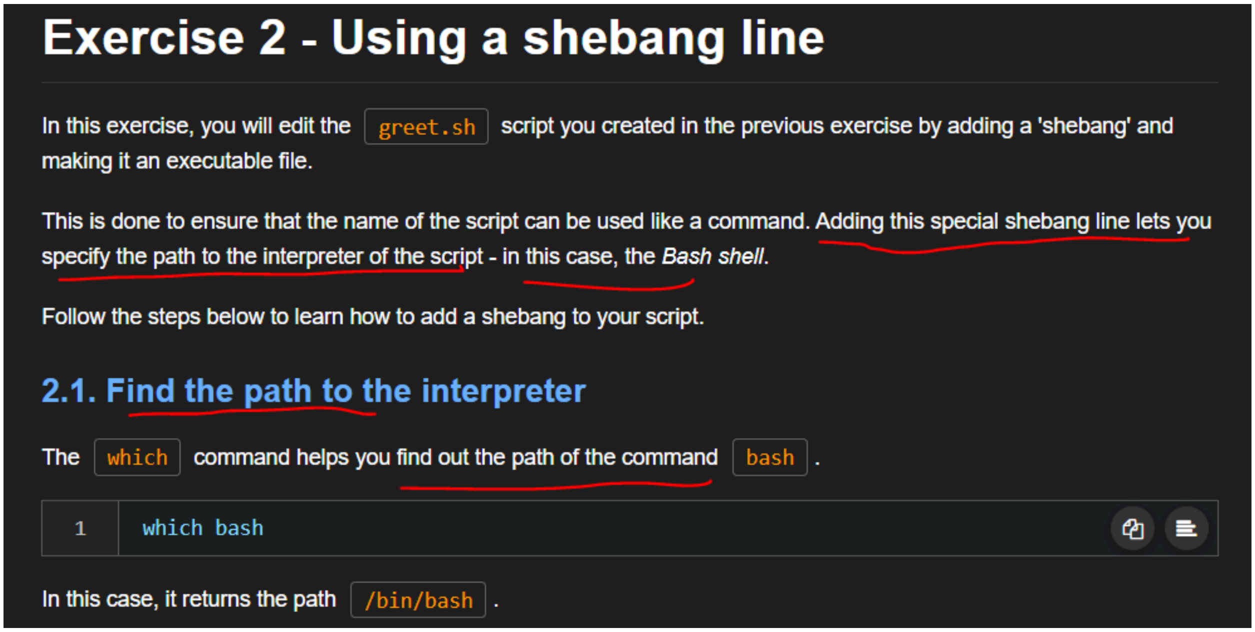
```
theia@theia-rsannareddy:/home/project$ bash greet.sh
Enter your name :Ramesh
Welcome Ramesh
Congratulations! You just created and ran your first shell script using Bash on IBM Skills Network
theia@theia-rsannareddy:/home/project$ 
```

Congratulations! You have successfully executed your first Bash shell script.



```
# This script accepts the user\'s name and prints
# a message greeting the user

# Print the prompt message on screen
echo -n "Enter your name :"

# Wait for user to enter a name, and save the entered
read name

# Print the welcome message followed by the name
echo "Welcome $name"

# The following message should print on a single line
echo -n "Congratulations! You just created and ran yo
echo "using Bash on IBM Skills Network"
```

```
theia@theia-naimbenalaya:/home/project$ ls -l greet.sh
-rw-r--r-- 1 theia users 522 Jul  1 08:17 greet.sh
theia@theia-naimbenalaya:/home/project$ bash greet.sh
Enter your name :naim
Welcome naim
Congratulations! You just created and ran your first shell scr
ipt using Bash on IBM Skills Network
theia@theia-naimbenalaya:/home/project$ 
```



Exercise 2 - Using a shebang line

In this exercise, you will edit the `greet.sh` script you created in the previous exercise by adding a 'shebang' and making it an executable file.

This is done to ensure that the name of the script can be used like a command. Adding this special shebang line lets you specify the path to the interpreter of the script - in this case, the *Bash shell*.

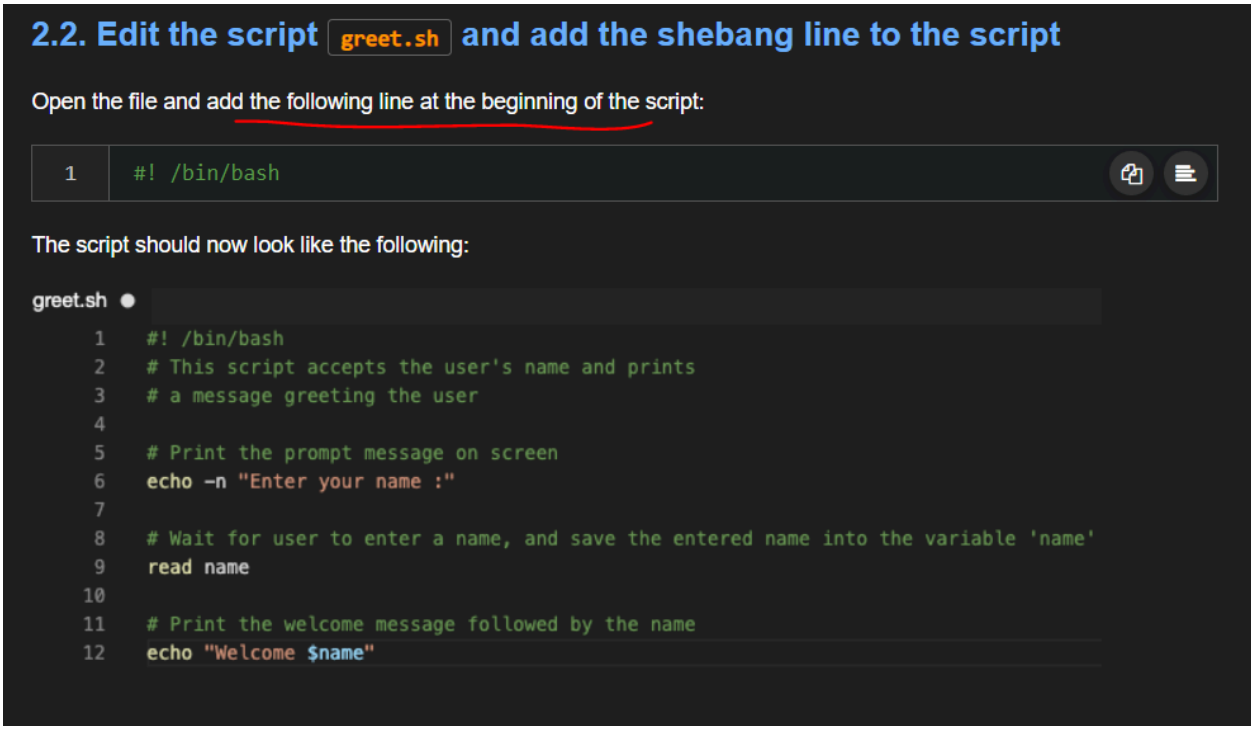
Follow the steps below to learn how to add a shebang to your script.

2.1. Find the path to the interpreter

The `which` command helps you find out the path of the command `bash` .

```
which bash
```

In this case, it returns the path `/bin/bash` .



2.2. Edit the script `greet.sh` and add the shebang line to the script

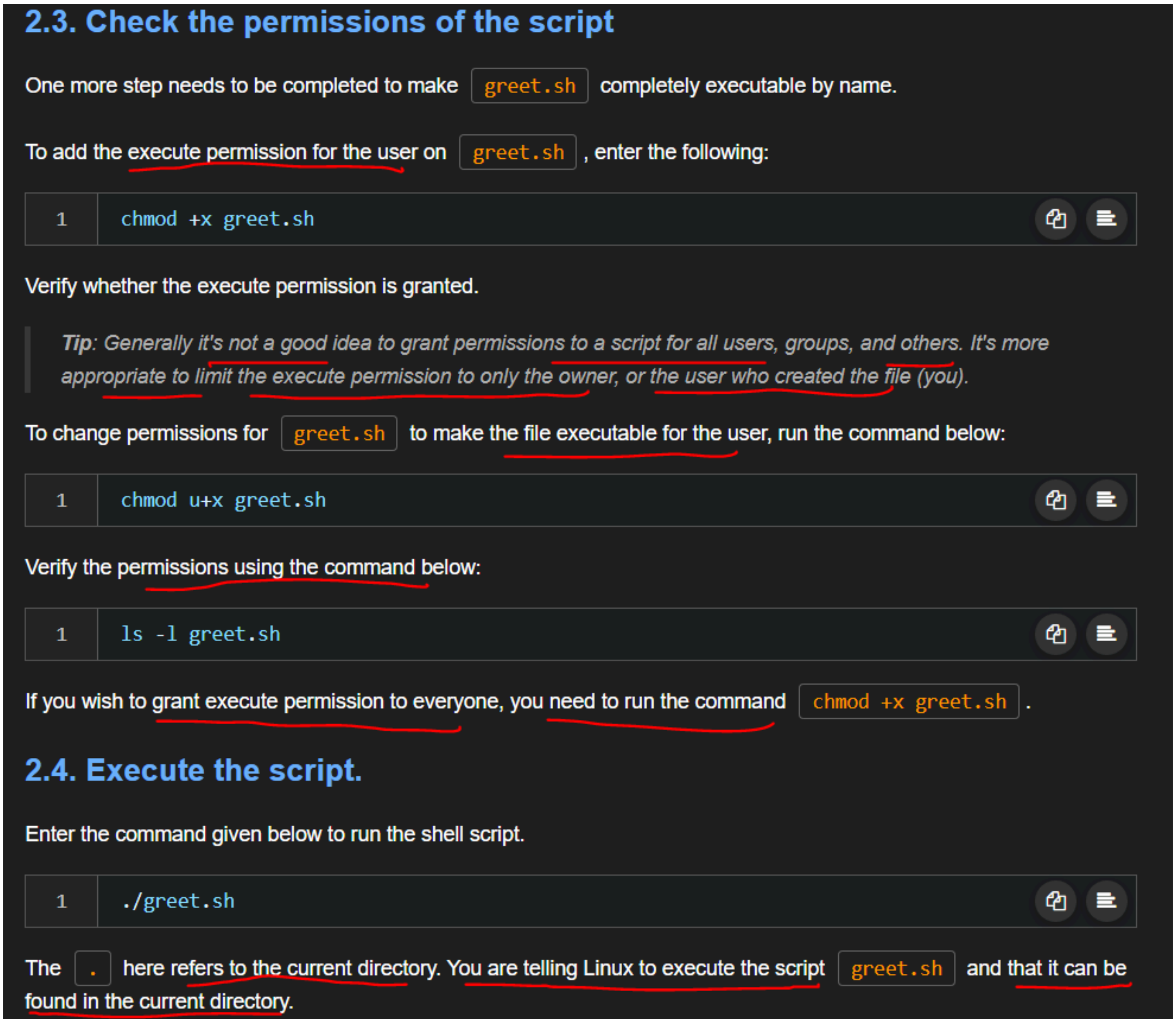
Open the file and add the following line at the beginning of the script:

```
#! /bin/bash
```

The script should now look like the following:

greet.sh ●

```
#! /bin/bash
# This script accepts the user's name and prints
# a message greeting the user

# Print the prompt message on screen
echo -n "Enter your name :"

# Wait for user to enter a name, and save the entered name into the variable 'name'
read name

# Print the welcome message followed by the name
echo "Welcome $name"
```



2.3. Check the permissions of the script

One more step needs to be completed to make `greet.sh` completely executable by name.

To add the execute permission for the user on `greet.sh` , enter the following:

```
chmod +x greet.sh
```

Verify whether the execute permission is granted.

> *Tip: Generally it's not a good idea to grant permissions to a script for all users, groups, and others. It's more appropriate to limit the execute permission to only the owner, or the user who created the file (you).*

To change permissions for `greet.sh` to make the file executable for the user, run the command below:

```
chmod u+x greet.sh
```

Verify the permissions using the command below:

```
ls -l greet.sh
```

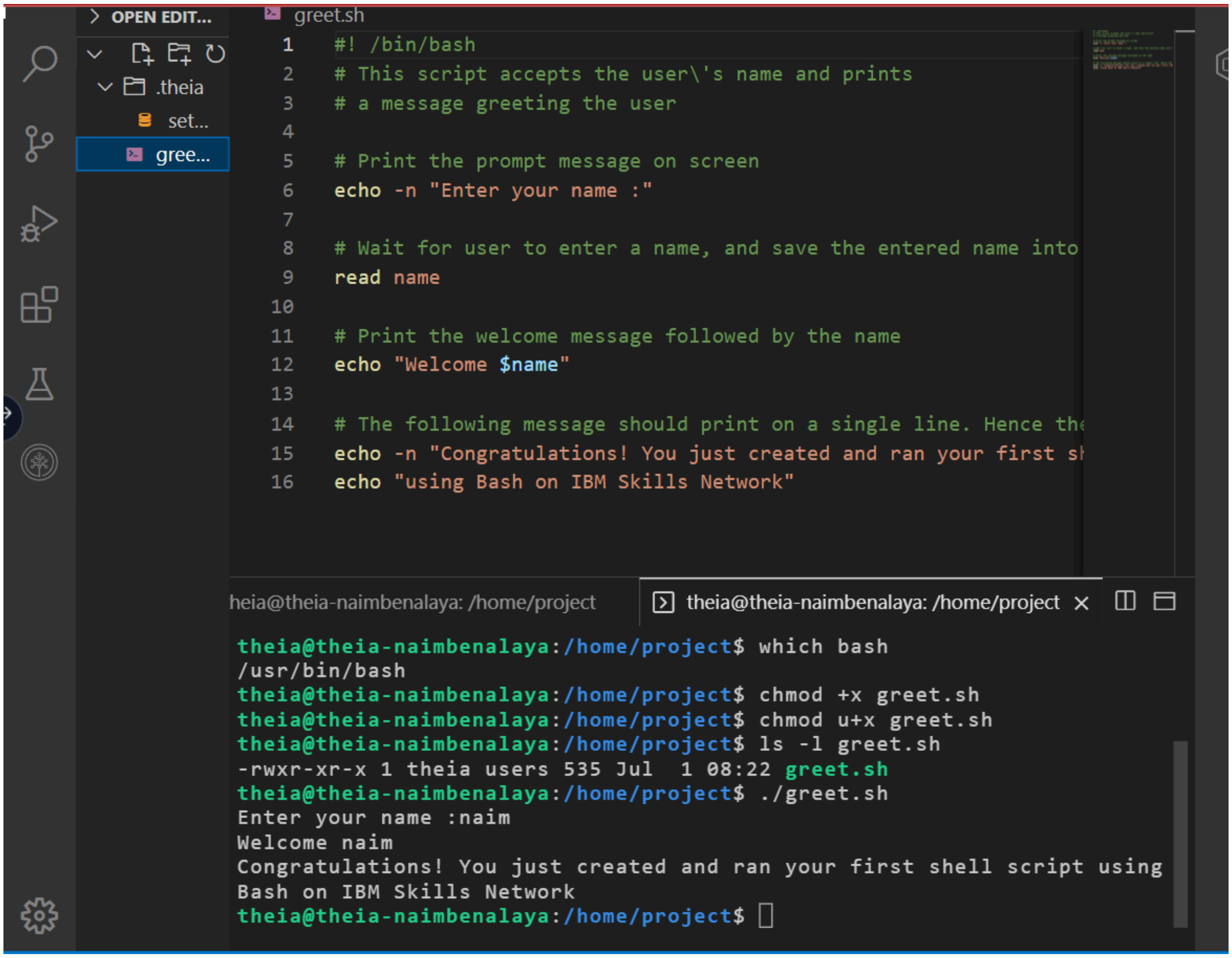
If you wish to grant execute permission to everyone, you need to run the command `chmod +x greet.sh` .

2.4. Execute the script.

Enter the command given below to run the shell script.

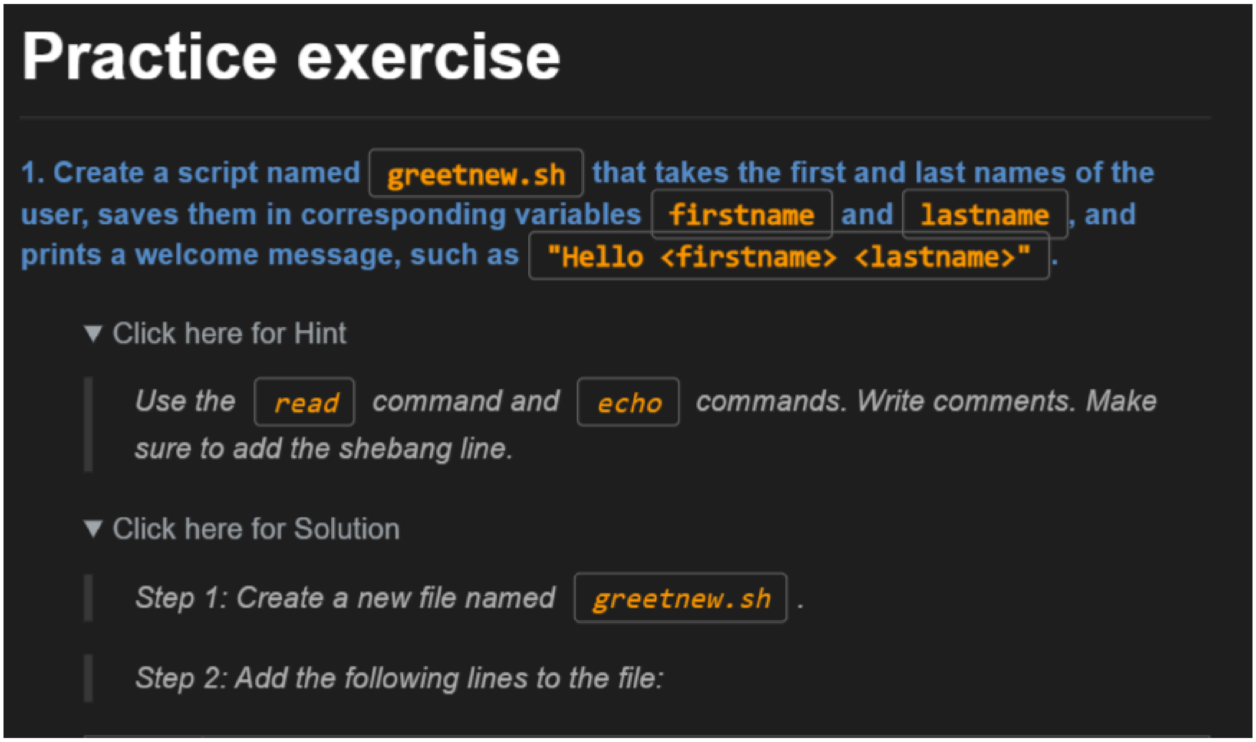
```
./greet.sh
```

The `.` here refers to the current directory. You are telling Linux to execute the script `greet.sh` and that it can be found in the current directory.


OPEN EDIT...
.theia
set...
gree...
greet.sh
heia@theia-naimbenalaya: /home/project
theia@theia-naimbenalaya: /home/project

```bash
#! /bin/bash
# This script accepts the user\'s name and prints
# a message greeting the user

# Print the prompt message on screen
echo -n "Enter your name :"

# Wait for user to enter a name, and save the entered name into
read name

# Print the welcome message followed by the name
echo "Welcome $name"

# The following message should print on a single line. Hence the
echo -n "Congratulations! You just created and ran your first sh
echo "using Bash on IBM Skills Network"
```

```
theia@theia-naimbenalaya:/home/project$ which bash
/usr/bin/bash
theia@theia-naimbenalaya:/home/project$ chmod +x greet.sh
theia@theia-naimbenalaya:/home/project$ chmod u+x greet.sh
theia@theia-naimbenalaya:/home/project$ ls -l greet.sh
-rwxr-xr-x 1 theia users 535 Jul  1 08:22 greet.sh
theia@theia-naimbenalaya:/home/project$ ./greet.sh
Enter your name :naim
Welcome naim
Congratulations! You just created and ran your first shell script using
Bash on IBM Skills Network
theia@theia-naimbenalaya:/home/project$ 
```



Practice exercise

1. Create a script named `greetnew.sh` that takes the first and last names of the user, saves them in corresponding variables `firstname` and `lastname`, and prints a welcome message, such as `"Hello <firstname> <lastname>"`.

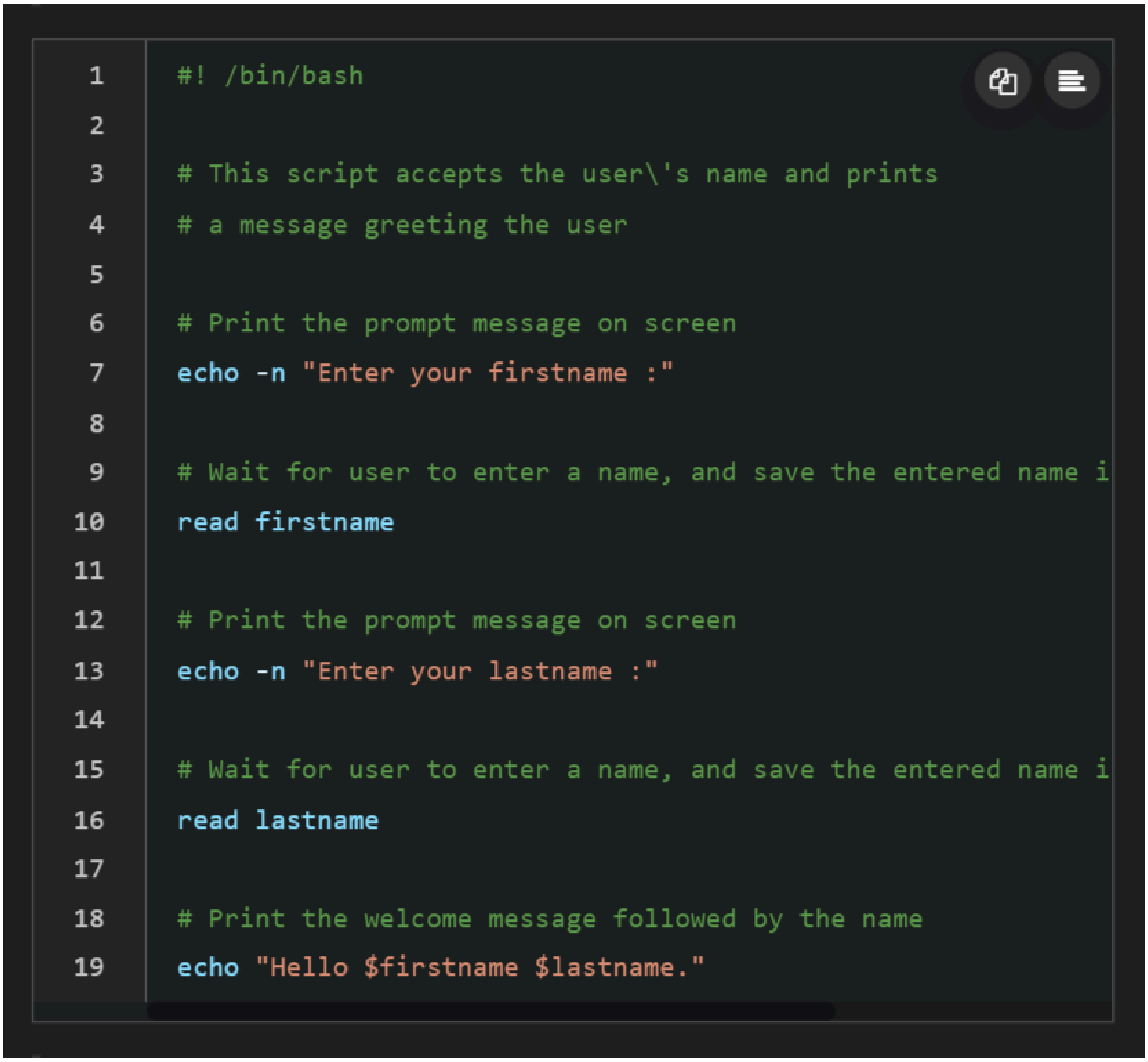
▼ Click here for Hint

> *Use the `read` command and `echo` commands. Write comments. Make sure to add the shebang line.*

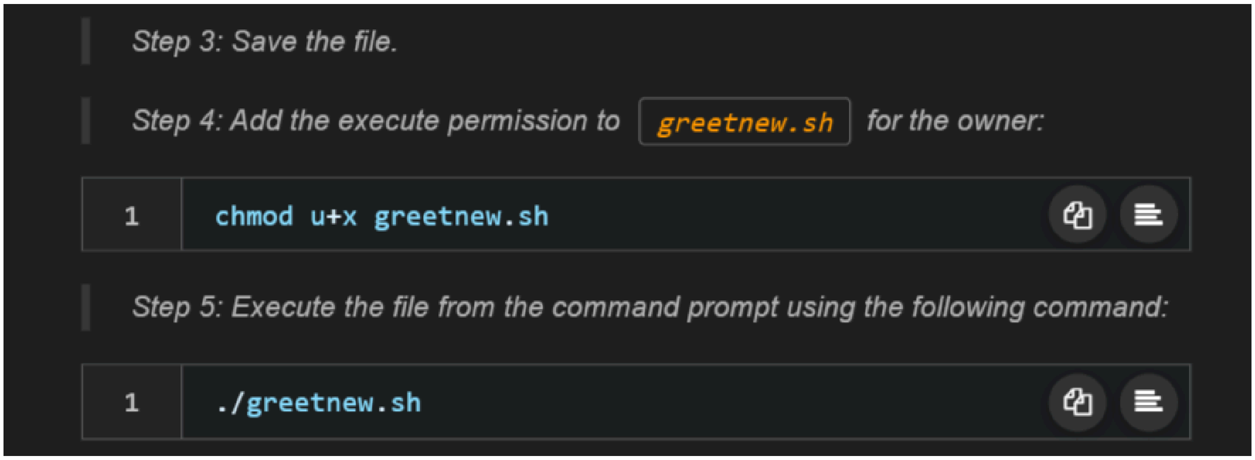
▼ Click here for Solution

> *Step 1: Create a new file named `greetnew.sh`.*

> *Step 2: Add the following lines to the file:*



```
#! /bin/bash

# This script accepts the user\'s name and prints
# a message greeting the user

# Print the prompt message on screen
echo -n "Enter your firstname :"

# Wait for user to enter a name, and save the entered name i
read firstname

# Print the prompt message on screen
echo -n "Enter your lastname :"

# Wait for user to enter a name, and save the entered name i
read lastname

# Print the welcome message followed by the name
echo "Hello $firstname $lastname."
```



> *Step 3: Save the file.*

> *Step 4: Add the execute permission to* `greetnew.sh` *for the owner:*

```
chmod u+x greetnew.sh
```

> *Step 5: Execute the file from the command prompt using the following command:*

```
./greetnew.sh
```

Summary

In this lab, you learned how to:

- Create and execute a simple Bash shell script
- Implement the shebang directive `#! /bin/bash` in a Bash shell script