

Assignment:3

Task:1

Tasks:

1: Basic Functions

Objective: Write a shell script that uses a function to print a greeting.

Certainly! Writing a shell script with a function to print a greeting is a simple task. Below is an example of a basic shell script that accomplishes this:

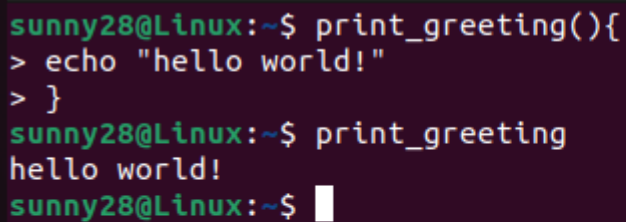
```
#!/bin/bash
```

```
# Define a function for printing a greeting
```

```
print_greeting() {  
    echo "hello world!"  
}
```

```
# Call the function to print the greeting
```

```
print_greeting
```

A terminal window with a dark purple background. The prompt is 'sunny28@Linux:~\$'. The user enters 'print_greeting(){' and the prompt changes to '>'. They enter 'echo "hello world!"' and the prompt changes to '>'. They enter '}' and the prompt returns to 'sunny28@Linux:~\$'. Then they enter 'print_greeting' and the output 'hello world!' is displayed. Finally, they enter a blank line and the prompt returns to 'sunny28@Linux:~\$' with a white cursor.

```
sunny28@Linux:~$ print_greeting(){  
> echo "hello world!"  
> }  
sunny28@Linux:~$ print_greeting  
hello world!  
sunny28@Linux:~$
```

Here's a breakdown of the script:

#!/bin/bash: This line specifies the shebang, indicating that the script should be interpreted using the Bash shell.

`print_greeting() { ... }`: This defines a function named `print_greeting`. Inside the curly braces `{ ... }`, you place the code that the function executes. In this case, it's a simple `echo` statement.

`echo "hello world!"`: This line within the function prints the greeting message.

`print_greeting`: This line calls (or invokes) the `print_greeting` function, causing it to execute and print the greeting.

Save the script with a `.sh` extension (e.g., `greeting_script.sh`) and make it executable using the following command:

`chmod +x greeting_script.sh`

Then, you can run the script using:

`./greeting_script.sh`