Project Brief: Improved Backup Script

Scenario

You have been using your backup script at work for some time now, and you have discovered a few issues that you would like to address.

Firstly, your script is not giving any output to the user. It would be nice to tell the user that the script is running to give them a bit of reassurance that their valuable data is indeed being backed up.

Secondly, some of your colleagues struggle to use the script because they do not have a `bash_course` directory on their system. To get around this issue, you want to modify the script so that it creates a backup in whatever directory the user runs the script from.

Your Task

This project involves five steps.

Step 1: Edit the `backup_script` to print out the following two lines to the user's screen when the script is run:

```
Hello, <User>

I will now back up your home directory, </path/to/home_directory>
```

For example:

```
Hello, Simon

I will now back up your home directory, /home/simon/
```

Note: The user's username should have the first letter capitalised in the first line of the greeting message.

Hint 1: You can use the `echo` command and shell variables to achieve this.

Hint 2: Look at the lecture on "shell parameter expansion tricks" to see how to capitalise the first letter of the user's username.

Step 2: Create a variable called `currentdir` and store in it the value returned by the `pwd` command.

The `pwd` command simply shows the path to the user's current working directory (i.e. the directory that they run the script from).

Hint: You will need to use command substitution to achieve this.

Step 3:

Use the echo command and the `currentdir` variable to tell the user where they have run the script from, and thus where their backup will be saved:

```
You are running this script from <current_directory>.

Therefore, I will save the backup in <current_directory>
```

e.g.

```
You are running this script from /home/simon/bash_course

Therefore, I will save the backup in /home/simon/bash_course
```

Step 4: Edit the `tar` command so that it backs up the user's home directory in the directory held by the `currentdir` variable.

Note that the verbose (`-v`) option should also be removed from the `tar` command so that the `echo` statements are easily visible to the user.

Step 5: Add a final line so that the script tells the user that the backup has completed

e.g.

```
Backup Completed Successfully.
```