Cardiff University
School of Computer Science & Informatics
Dr Natasha Edwards

Objectives:

- To practice various Git commands.

Assumption:

- Git should be already installed and ready for you to complete today's task. If it isn't, please contact the university IT Service Desk.

Instructions:

- Complete all tasks.
- Write down the git commands you use for each task.
- When a task asks you to modify a file, just type a short phrase, e.g. "change 1", "change 2", etc. Insert and keep a newline at the end of each file (**`git diff`** doesn't like files without a newline at the end of the file).
- When you perform each commit, use "**`commit X`**" [1], or some other short phrase that will help remember what each particular commit was for.

Tasks

Part 1: Git commands

1. Create a directory **`git_practice`** in a suitable directory. *(Hint: you can use* **`mkdir`** *command to create this directory using the command prompt.)*

   ✓ Check that you now have **`.git`** folder in **`git_practice`** dir.

2. Create 5 empty files in this directory: **`README.txt`**, **`TODO.txt`**, **`hidden.txt`**, **`module1.txt`**, **`and removeme.txt`**
3. Start tracking **`git_practice`**.

   git command:

4. Run **`git status`**. What is the status of the files in the **`git_practice`** directory?

5. Stage **`README.txt`** and **`TODO.txt`** files.

   git commands:

   What is the status of the two files?

   Modify **`README.txt`** by adding a line of text, and save the file.

[1] where '*X*' represents the consecutive number of each commit

6.

a. Use "short" status to check the current state of your git repo.

git command:

b. Interpret the output:

✓ It is a good idea to run **`git status`** often to check your files' status.

7. Tell git to ignore **`hidden.txt`** file. Check this file is no longer listed.

What actions do you need to do and what git commands do you need to run do to accomplish this?

8. Modify **`TODO.txt`**, stage and commit the changes.

git command(s):

9.

a. Modify **`TODO.txt`** file again, and commit the changes without staging this file first (i.e. skip the **`git add`** command).

git command:

b. Run **`git status`**.

What is the status of your repo?

10. Modify **`TODO.txt`** again. Run **`git diff`** command. Interpret the output:

11.

a. Stage **`removeme.txt`** file. Run **`git status`**.

What is the status of **`removeme.txt`**?

b. Remove **`removeme.txt`** from git, but keep it the directory (i.e. do not delete it completely). Run **`git status`**.

git command:

What is the status of **`removeme.txt`** now?

12. View the history of the commits you made so far.

git command:

How many commits have you made so far?

13.

a. Create the tag "**`ex13a`**". Modify **`README.txt`**. Commit the changes.

git command(s):

b. Display the data associated with the **`ex13a`** tag.

git command:

14.

a. Try the **`git log --oneline --all`** command.

What does this command do?

b. Try the **`git log --oneline --decorate --graph --all`** command.

What does this command do?

What is the difference between this command and the one in 13 (a)?

c. Create alias **`pretty`** for the command in 13 (b). Check that it works as intended. (*Hint: it might be necessary to use single quotes for the command that is used for the alias)*

git command: