Git exercises

0. If you haven't already, make sure Git (https://git-scm.com/) is installed on your computer.

1. Configure Git for use on your computer by providing your name and email address, to identify yourself in commits:

   ```
   git config --global user.name "First Last"
   git config --global user.email my@email.com
   ```

   In windows, use the Git Bash command line or Anaconda prompt instead of the usual windows command line.

2. Create an empty folder somewhere on your computer. Call it `testproject` . Go to that folder in the command line. Create a new git repository within that folder using `git init` . See if you can locate the hidden `.git` folder that was just created (try `ls -al` in Mac/Linux, or `dir /ah` in Windows).

3. Check the status of the repo with `git status` . Now create an empty text file named `test.py` (`touch test.py` in bash). Use `ls -al` to check that it's there. Now check the `git status` again. What do you notice?

4. Mark/stage the new `test.py` file as ready for committing: `git add test.py` . How has `git status` changed? Now commit the new file to the repository with `git commit -m "Add test.py"` . How has `git status` changed?

5. Now there should be one commit in the log. Have a look using `git log` .

6. Open your `test.py` file and add a line that prints a message 10 times. Test the script by launching IPython or Jupyter in the same folder and typing `run -i test.py` . What's the `git status` ? Have a look at the changes in more detail with `git diff` . Now stage and commit this change as above, but with a different descriptive commit message. Check `git status` again!

7. Write a new function for printing messages a given number of times. It should take two arguments: the string to print, and the number of times to print it. Put this function in a new file called `mylib.py` . Then change your `test.py` script to import that function, and then call the function to get the same output as before. Test it. Check your changes with `git status` and `git diff` , stage them, and commit them. Now check your commit history one last time.