

Goals of the Assignment

The goal of this homework assignment is to give you more practice with the basic Git workflow (status, add, commit, push), working with remote Git repositories, and more practice viewing, changing, and creating environment variables.

Activities

1. If you are working on a computer that is different from the one that you used in class, you will need to clone your repository onto the new computer:
 - a. Open the repository in your browser (if you need to, you can find it by clicking the original GitHub Classroom Invitation again).
 - b. Create a directory in your user directory: `SoftDevI\Unit01\`
 - c. Change into the directory and use `git clone` with your repository URL to download the repository to your new computer.
2. Download and save this PDF file in the repository.
3. Use notepad to create a new file called `questions.txt` and type your full name on the first line of the file. As you edit the file, make sure to include a blank line each time you add text to the file (e.g. answer a question) so that it is easy to tell when one answer ends and the next begins. Make sure to save the file each time you type an answer.
4. Run `git status` and redirect it to a file called `status.txt`.
5. Use the Git workflow (status, add, commit, push) to push the new files into your repository. Note that, because the PDF file name contains spaces, you will need to use quotes to stage it, e.g. `git add "A filename with spaces.txt"` *Hint*: use tab to complete the filename.
 - a. Remember, if you forget to use `-m` when committing, you will need to use **vim** to enter a comment. Type `i` (to insert) and type your comment on the first line. Then press **ESC** to open the vim command prompt (at the bottom of the window). Type `:x` into the prompt to save and quit.

6. Use your browser to open your repository on <https://www.github.com> and verify that the files were pushed.
7. Create a new, empty directory inside the repository directory using your first name, e.g. "SoftDevI\Week01\assignment-1-2-harry\Harry".
8. Use the Git workflow to push the empty directory to GitHub, and check your repository in a browser. What happened? Edit your "questions.txt" and type your answer. Remember to leave a blank line after your name.
9. Move (do not copy) your "questions.txt" file into the directory that you created in the previous step. Use the Git workflow to push the moved file to GitHub, and check your repository in a browser. What happened? Edit your "questions.txt" and type your answer. Is your new directory present? How many copies of the "questions.txt" file are present? Explain why you think it happened (feel free to Google).
10. Change into the top directory for your Assignment 1.2 repository (the one with the .git directory and most of your files) and check the status of your files. You should see that the original "questions.txt" file has been deleted but is listed under "Changes not staged." This is the way that Git works: even when you **delete** files, you must use the Git workflow to delete them from your repository. Do so now, and then verify that the old "questions.txt" file has been removed from your repository on GitHub.
11. If you have not already, change into the directory with your name. Display the environment variables on your computer and redirect the output to a file called "before.txt". Use the Git workflow to push the new file to GitHub. Close your command prompt.
12. Use the environment variable editor in Windows to create a new environment variable with the name "SOFT_DEV_HOME" and a value that is the absolute path to your SoftDevI directory, e.g. "C:\Users\Harry\SoftDevI". You **do not** need to include quotes around the value.
13. New environment variables will not appear in a command prompt that is already opened, so you should **close any open command prompts** and launch a new one. Navigate to the directory with your name inside your Assignment 1.2 repository. Display the environment variables on your computer and redirect the output to a file called "after.txt". Use Notepad to verify that your new environment variable appears in the file! Then use the Git workflow to push the new file to GitHub.
14. Don't forget to push **all** of your files to GitHub. Use a browser to open your repository on <https://www.github.com> and verify that all of your files are there.

Submission Instructions & Grading

Be sure that you have pushed all of your assignment files to your GitHub repository **before** the start of the next class. Your grader will verify that you finished your solution before the deadline.

See the course syllabus for the rubric that is used for grading homework assignments.