**MNWD Git Demo**

9/21/2016

[Presentation Link](https://docs.google.com/presentation/d/1MvXSs9CyIMcYBFhHiUVGY6I_x6rwo9TEOsSVwf9edUI/edit?usp=sharing)

**Basics**

**clone**, **add**, **commit**, **push**, **pull**, **status**, **log**

1. Go to your personal Github page and click the **New repository** button. Name your repository **testRepo** and make it **Public**. Also select the option to **Initialize this repository with a README**. Finally, click the **Create repository** button.

2. You will now see the page for your repository, with the contents of the README shown. You can now clone the repo to your local machine by opening your Terminal and running the following, substituting in your username.

**git clone https://github.com/username/testRepo.git**

3. Move into the directory of the cloned repository using the “change directory” command.

**cd testRepo**

4. View the contents of the directory using “ls”.

**ls**

5. Create a new file using the “echo” command with the redirect option.

**echo “This is my test file” > test.txt**

6. Use git status to see that the file exists, but is currently unstaged.

**git status**

7. Use git add to stage the file.

**git add .**

8. Run git status again to see that the file is now staged.

**git status**

9. Now it is time to commit the changes.

**git commit -m “Added my a new file called test.txt”**

10. You can see that your commit has been made by using git log.

**git log**

11. Now, check the Github page and notice that your file isn’t there yet. That’s because you have to push your changes. Do so by running git push.

**git push**

12. Refresh the page, and you should now see your file listed.

13. Now, click the README.md link on your Github page, then click the Edit button (pencil icon). Add a line of text to the README. Then in the **Commit changes** box below, add a short description in the Update field such as “Added info to README”, then click the **Commit changes** button.

14. If you open the README in a text editor on your local machine, you’ll notice your last change isn’t there. This is because you need to pull down your changes. Do this by running git pull from your command line.

**git pull**

15. Now if you open the README, you should see your changes.

**More Advanced**

**branch, checkout, merge**

Github Repository: <https://github.com/California-Data-Collaborative/mnwd_test>

1. From the previous exercise, return to the parent directory by running the following:

**cd ..**

2. From the command line, clone the repository to your local machine.

**git clone https://github.com/California-Data-Collaborative/mnwd\_test.git**

3. Move into the directory of the cloned repository using the cd command.

**cd mnwd\_test**

4. View the contents of the directory using “ls”.

**ls**

5. Create a new branch for developing a feature independently from the master branch. Substitute in your first name.

**git branch name\_feature**

6. Now that the branch is created, you can switch to the branch using the checkout command.

**git checkout name\_feature**

7. Running git branch with no arguments will show a list of branches with an asterisk next to your active branch.

**git branch**

8. Create a new file using the “echo” command with the redirect option. Substitute your own name for the file name.

**echo “This is my test file” > name.txt**

9. Use git status to see that the file exists, but is currently unstaged.

**git status**

10. Use git add to stage the file.

**git add .**

11. Now commit the changes.

**git commit -m “Added my a new file called name.txt”**

12. By default, git push will only push changes that occur on the master branch. To specify another branch to push, you must also provide the name of the remote server (usually origin) and the branch name.

**git push origin name\_feature**

13. Now, refresh the Github page and notice that your branch appears in the list of branches.

14. Back on your command line, it’s time to merge this feature with the master branch. First, switch back to the master branch by using the checkout command, then use the merge command to specify which branch to merge.

**git checkout master**

**git merge name\_feature**

15. Attempt to push the newly updated master branch by using git push. NOTE: You may get an error telling you that your master branch is not up-to-date. This happens if someone else has committed to the master branch and you have not yet pulled their changes. Use git pull to pull down the most recent changes to your local machine.

**git push**

**git pull**

This concludes the walkthrough. Be sure to consult the resources linked at the end of the [presentation](https://docs.google.com/presentation/d/1MvXSs9CyIMcYBFhHiUVGY6I_x6rwo9TEOsSVwf9edUI) for more examples on how to use git.