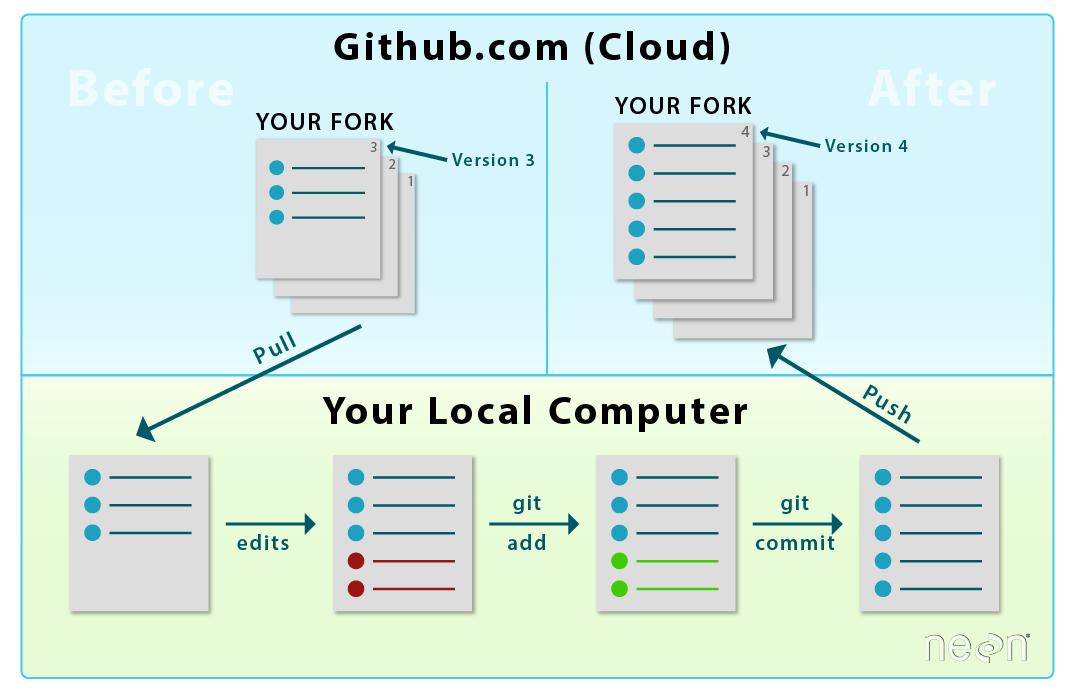
Git Homework 1

**What is Git**

Git is the most widely used version control system in the world today. Knowing how to use Git is essential for any software development role you may come across in the real world. There are several Git applications with user interfaces, but learning it through the command line is the best way to thoroughly understand what Git does and how it works. Below is a diagram demonstrating a basic workflow of Git.



Git is important for developing software because it prevents things from going horribly wrong (e.g. someone accidentally created a null pointer which breaks the entire codebase) and provides accountability for the contributions being made to a repository (e.g. seeing who create the null pointer exception). If something does go wrong and you cannot find the change is causing a detrimental error, you can simply revert the repository back to when it was fully working. Sharing files back and forth is not effective in the real world because codebases in production will be massive and many people will be working on them at once. Merge conflicts may arise if people are working on the same file. However, these conflicts can easily be solved. When a merge conflict occurs, you must correct it or else your changes will not be merged.

A repository is a codebase that will house your files. A repository has branches attached to it. The first branch is the master branch. Master should reflect your current working codebase and will be referenced in production. Then there is a develop branch that you should create but is not necessary. Develop will contain changes that will be merged into master for new patch or update/release. Then there will be branches for fixes or new features. Below is a standard workflow for branching in Git.



Some useful links to learn more about Git:

ARTICLE: <https://hackernoon.com/understanding-git-fcffd87c15a3>

VIDEO: <https://youtu.be/r63f51ce84A>

**Basic Command Line Commands**

Below is a list of commands that may be necessary for moving around your file directory. Feel free to explore more commands.

cd <directory> : changes the directory

mkdir <directory name> : creates a new folder

ls (for Mac) or dir (for Windows): lists all files and directories in current location

mv <from> <to> (for Mac) or move <from> <to> (for Windows): moves a file or directory to a new location

cd .. : moves back up a folder in the directory

**Exercise 1.1**

1. Create a Github account using your TU email
2. Download Git. For Mac, use <http://git-scm.com/download/mac> and for Windows, use <http://git-scm.com/download/win>.
3. Set up local configurations to attach your Github account to your commits:

git config --global user.name "Firstname Lastname"

git config --global user.email "yourGithubEmail@email.com"

I highly recommend applying for the Github Student Developer Pack. It provides a ton of free resources including courses and also gives you the ability to create private repositories. Apply for that here: <https://education.github.com/pack>

**Exercise 1.2**

1. Cloning makes a copy of an existing repository. You can clone a repository using its link on Github.

git clone [\*\*url of repository\*\*]

You should now fork the repository at <https://github.com/ryfellow/cis4296-starter>. There is “Fork” button in the top right of the page after clicking the link. Forking creates an exact copy of a repository and displays it on your Github. Forking is useful for proposing fixes/features in open source repositories. Clone the forked repository to your computer using the command above. (I recommend that you clone it into your desktop for ease of access. Use the command cd desktop to change your directory to be your desktop and then the clone the repo.)

1. Create a new file in a text editor or via the command line. Make sure it is in the folder of the repo you cloned. You can use the command line to create a new file if you do not feel like leaving the command line. The Mac command to create a new file is touch [\* new file name with extension \*] and with Windows it is echo “” > [\* file name with extension \*]. If you are using the command line, again make sure you are in the cloned repo’s folder, if you aren’t just locate the file and drag and drop it into the folder. You can add that new file for staging in the command line using the commandgit add [\*\*file with extension e.g. (file.js)\*\*]. Track your first file using command above.
2. Checking the status of your repository is crucial to know what has yet to be staged or what is currently staged. It also provides information on what branch you are currently on. Use the commandgit statusto check the status of your repository. You should currently have one file staged.
3. In your editor or IDE, edit the new file that you created and save the changes. You can also do this through the command using Vim. Check the status of your repository to see that it needs to be staged. Use the commandgit add –p to go through each of the changes and answer ‘y’ for yes and ‘n’ for no. If you make every change needed for a branch without committing along the way, the PR will contain 1 commit which may be tough for a reviewer to follow. Going through each change and picking if you want it to be staged or not is useful for organizing each step you made along the way. Use the command git commit –m “your message” to commit these changes. The message will be what is displayed for each commit. A Git commit message should be short but still effectively describe the functionality the changes in the commit will bring.
4. To push these changes to the server, use the command

git push to do so. The reason you may simply use this command is because you are pushing directly to the master branch which your local repository knows about. If you create a new branch on your own, you will need to let Github know to track it. The command for that is

git push --set-upstream origin [\*\*your branch name\*\*] **.**

1. Congratulations, you have successfully created your first Git commit and sent the changes to the server!

*SUBMISSION*

1. Submit the link to the forked repository on Canvas.