

Git/GitHub Usage

TL;DR

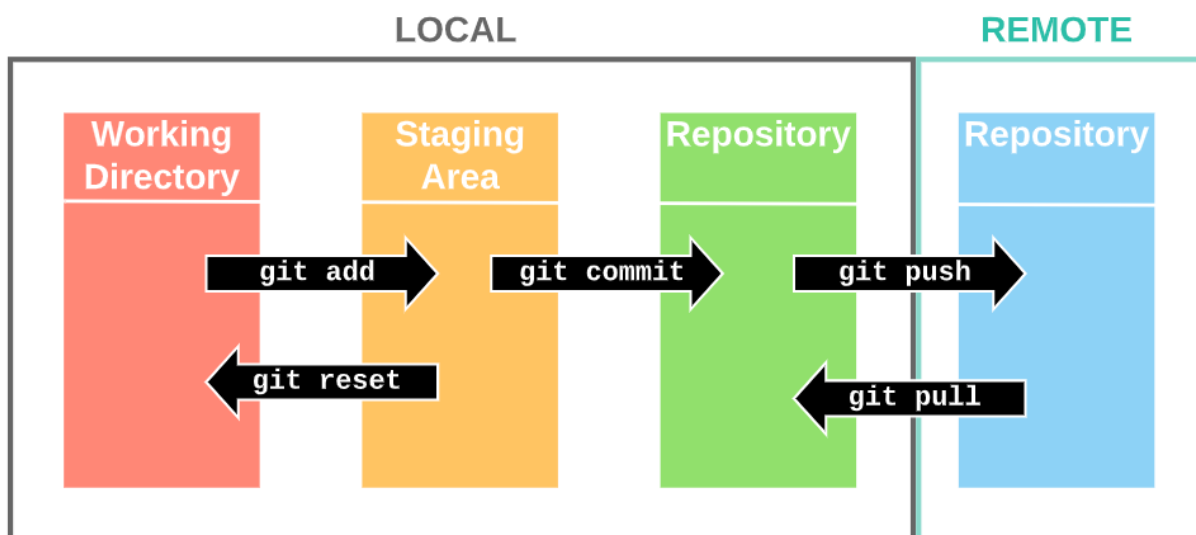
As the lab increases in size and we have multiple Research Assistants (RAs) that come and go, it's important to have all the code stored on GitHub for future reference. It's important because once the code is on GitHub, the staff can further do code reviews and inspect any code for the future. For any RAs, please follow these guidelines:

1. Please document your most important scripts that you have according to the example script provided and these guidelines.
2. Please put your code on GitHub in a repository assigned to your work. If you already have a repository, confirm with Eric Shapiro shape@seas.upenn.edu what repository to work out of.
3. Please commit your code to GitHub with the instructions attached below.
4. Please send an email to yli12313@seas.upenn.edu when you are finished.
5. If you have any access problems, network issues, and can't push/pull/fetch from GitHub, please contact yli12313@seas.upenn.edu.

- Best, Y (12/16/21)

What is Git/GitHub

Git is a distributed version control system used to store code and helps software engineers, data engineers, and data scientists to store their code in a central repository as well as collaborate on projects. GitHub is an online code repository owned by Microsoft that will host code and documentation, so that people can collaborate on projects. The intricacies of GitHub can be pretty unintuitive for a beginner, but using GitHub is not hard. You can use the command line in order to use Git or use any Graphical User Interface (GUI) tool if that's easier. There are really only two concepts that need to be understood to really use Git/GitHub.



Concept #1: Adding, Staging, Pushing/Pulling Files

Please see the diagram for reference. It shows the workflow of most Git processes. It's a very easy process with two directions, moving files from the local computer to GitHub and moving changes from GitHub to your local computer.

Moving files from Local Computer to GitHub

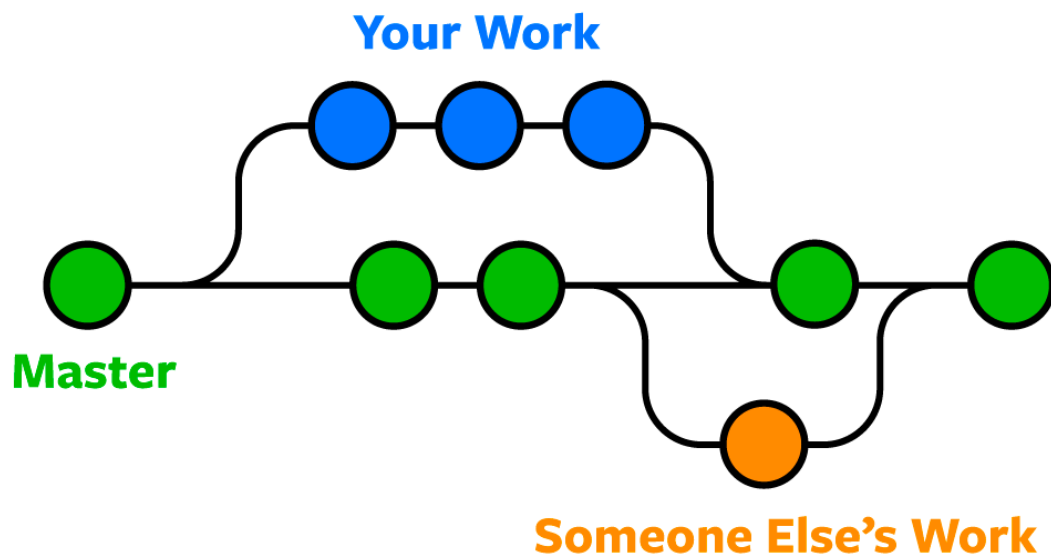
- Use *git add* to stage a file/files.
- Use *git commit* to commit the file/files to your local git repository.
- Use *git push* to push the code from the local git repository (your computer) to the remote git repository hosted on GitHub.

Moving files from GitHub to Local Computer

- Use *git pull* to pull files from the remote git repository to your local computer.

Concept #2: Work out of Your Own Branch

When you are working with multiple people, it's important to segment and differentiate your work from other peoples' work if both of you are working out of the same repository. For those reasons, GitHub has the concept of **branches** that segment your work from the work of other people. This way two people can work on a set of code and develop features independently, but merge all the code together when it's ready. Working out of your own branch is almost always a good idea so that you are not working out of the **main** branch, which is the code that is production ready.



Branching Commands

- Use *git branch -a* to list all branches.
- Use *git checkout [branch name]* to switch to a branch.
- Use *git checkout -b [branch name]* to create a branch and switch to it.
- Use *git push -u origin [branch name]* to push changes to remote repository and remember the branch.

- Use *git push* to push further changes to the branch specified above.

Resources to Practice Using Git

- https://learngitbranching.js.org/?locale=en_US
- <https://lab.github.com/>
- <http://gitready.com/>