Git Basics

...

Objectives

- Learn the basic workflow of Git
- Learn the common commands used in Git
- Practice using these commands

What is Git?

- Git is a version control system
- Version control is a system that tracks file changes so that you can recall specific versions later

Github?

- Git ≠ Github
- Git is a version control system, written to keep track of changes to a code base. The codebase itself is referred to as a repo or repository
- Github is a web service for git repos, which doubles as a social network where developers can share code and projects

Create a New Repo

- Let's create a new directory. Inside that directory, run *git init* to create a new git repo
- Careful: Never init a repo in a repo. Ever.

Or, Clone an Existing Repo

- You can create a working local copy of a repo from github by running the command *git clone git@github.com:repo_to_clone.git*

Workflow - Adding and Committing

- Step 1: Propose changes by running the command:
  - *git add <filename>*
  - or : *git add .* (to add all files)

- To commit the changes you've made, run the command
  - *git commit -m "commit message"*

  This commits the file to the head.

Pushing Changes

- The changes we made are in the head of the **local** copy of this repo
- If we want to send these changes to the remote repo hosted on github, we run the command:
  - *git push origin master*

Updating Local Repos From Remote

- If your local repo falls behind your remote repo, you will need to pull changes
- This happens if you work on a project across multiple machines, or if someone else made changes to the remote repo
- Get the most updated version of the repo by running:
  - *git pull*

Keeping Track

- You'll forget what you modified. Check the status of your repo, use the command: *git status*
- This will tell you the current branch and if it is up to date with 'origin/master' as well as the current tracked (added) files

Fixing Mistakes

- You can replace local changes in the working directory with the copy from the HEAD by using the command:
  - *git checkout -- <filename>*

- You can roll back if you need to. You will need the commit id, that you can check for on github or on terminal. Then run:
  - *git reset -hard <commit id>*