Intro to Github

Jessica Young
jyoung22@nd.edu
GitHub Basics

1. Installing GitHub and Git
2. Connecting Git and GitHub
3. Why use Git?
Installing GitHub

If you haven’t already, create an account on GitHub and install Git and GitHub desktop on your machine.

https://git-scm.com/downloads

https://desktop.github.com
Setting Up GitHub

Once installed, you’ll want to connect your GitHub desktop to your account. To do so in the desktop client, follow these steps

1. Click on ‘GitHub desktop’ menu in the top left
2. Click ‘Preferences’
3. Select ‘Accounts’
4. Under “GitHub.com” click ‘Sign In’ and add in your credentials
5. Voila!
Configure Git for GitHub Desktop

This will connect your email to your GitHub for when you make commits.

1. Signin to GitHub (online)
2. Click on your profile picture (top right) and click on ‘Settings’
3. Click on ‘Emails’
4. Put in the email you want to use (probably your ND email!)
5. In your GitHub Desktop application, click on ‘GitHub desktop’ menu in the top left
6. Click ‘Preferences’
7. Select ‘Git’
8. Make sure your Name and Email are filled in properly!
Why Use Git?

Version Control!

Ever create good code and then change something and everything breaks? Using GitHub allows you to *revert* back to when your code worked and shows you exactly what you changed to make it break!

Also very useful for if you make multiple versions of the same code. This way you can see what your most recent version is compared to past versions.

Allows you to determine what changes collaborators make on your code.
Git Repositories

1. What are they?
2. How to create them on GitHub
3. Connecting to Already Existing repos on GitHub
What is a Repository?

A folder.
Creating Repos on GitHub

On your homepage, click the green ‘New repository’ button!

** ALWAYS click to Initialise this repository with a README

This allows you to immediately clone the repository to your computer.
A blank repository on GitHub should resemble this!

**IntroToGit**

Wow it's an intro!
Add a GitHub Repository to Your GitHub Desktop

1. Click the green ‘Clone or Download’ button
2. Copy the link that appears to your clipboard
3. In the GitHub Desktop application, Click ‘File’
4. Click ‘Clone Repository’
5. Go to the URL tab and paste in the URL from your clipboard
   a. Note: you can change the local path to be wherever you want to store this repo
6. Now click ‘Clone’
7. KABLAM!
Interacting With Repos

1. Documenting Your Project
2. Pushing to GitHub
3. Pulling from GitHub
Documenting Your Project

The key to success is to always document your projects. This can be as simple as creating a text file that declares what the project is, what it aims to do, and what your next steps.

This is usually what the README file in GitHub is for!
Documenting Your Project

1. In Github Desktop, click on the blue link that says ‘open this repository’
2. In your local folder, open up your README.md file using your preferred text editor (Notepad, Notepad ++, Atom, TextWrangler, etc.)
3. Write a few lines- it can be literally anything since this is practice! And save your file!

** Note, these steps should also be followed for when you’re adding new files. I.E. place your files to add inside this local folder when you want to add new files! Alternatively, brute force it and add the file directly using GitHub online 🤷 **
Pushing to GitHub

Back in your GitHub Desktop, you can see that it’s reflecting these new changes in your MD file. You need to make sure that this changed file has a checkmark to the left of it like so. This yellow circle symbol indicates that changes were made that haven’t been pushed yet.
Pushing to GitHub

1. Before you can push, aka upload this file to Github and merge it with the old one, you have to tell Git you want to do that (this is called adding and committing which are normally 2 separate steps if using the command line). This is accomplished by having that checkmark shown!

2. Now you need to fill in the summary or else you cannot push to Github. This is required so that GitHub can show why these changes occurred. This is useful for you to look back and see why you made certain changes in your code. Furthermore, you can then use this to go back to certain commits if your code every breaks. A more detailed description can be filled in, but is not necessary.
Pushing to GitHub

3. Click the blue ‘Commit to master’ button when ready to merge online.
Pushing to GitHub

4. Once you’ve made the commit, now you can push your updated file on line. To do this, click where it says ‘Push Origin’ in the top right of the Desktop application.

5. Now if you refresh your online GitHub repo, you’ll see your new commit!
Pulling From GitHub

This is only useful if you are working from multiple computers or with multiple collaborators.

There are 3 ways to pull using GitHub Desktop,

1. clicking on the ‘Fetch Origin’ button (which only appears if you have no commits pending)
2. by going into the Repository menu and clicking ‘Pull’
3. Using your machine’s hotkeys (i.e. shift, command P for Mac).
Advice to Live By Forever

❖ At the beginning of every day, when you’re about to work on a project that is hosted in GitHub, PULL! You never know which collaborator got inspired at 4am and made new commits online!
❖ At the end of every day, when you’re about to stop working on a project hosted in GitHub, PUSH your changes to the online so that others know your progress and can work off it when they get inspired at 4 am.
❖ All of this avoids merge conflicts! *SCARY!*
❖ Don’t be afraid of branching. It’s just fancy talk for folders. And time. (like if everything is already good, and you want to add more but don’t want to break the Master, that is the finalized product): good for coding new features
Interactions!

Now stop sleeping, and try pulling your neighbor’s/friend’s GitHub projects.

** Try to figure out how to add one another as collaborators to the repo. If they aren’t a contributor then they can pull and then make a push request which requires approve to finalize, not ideal if you’re actually collaborating!**
Interactions! V 2.0!

Now try forking another friend/neighbors repository.

** Discussion: What is forking? Benefits? Drawbacks? Problems with the name?**
Interactions! V 3.0!

Now try forking another friend/neighbors repository.

** Discussion: What is forking? Benefits? Drawbacks? Problems with the name?**
Try adding topics to your projects to make them easier to find!

Issue tracking is useful for collaborators or non-collaborators to remark on bugs ("features") and opens up discussion threads. Can be marked as resolved when the bugs have officially become features.