Colby Witherup Wood
NUIT Research Computing Services
colby.witherup@gmail.com

INTRO TO GIT AND GITHUB FOR THE AMATO LAB

Topics for today

- Git vs. GitHub
- How Git works conceptually
- The normal Git workflow
- What to post on GitHub
- What to post privately vs. publicly
- Possible guidelines for the lab GitHub site
- How to enter Git commands
- How to navigate to directories on the Command Line
- Link everyone's Git to their GitHub site
- Assign the tutorial

Preparation

- Install Git: https://git-scm.com/downloads. Select the Mac, Windows, or Linux/Unix download. We will not be using the GUI.
- Create a GitHub account if you do not already have one: https://github.com/join.
- These slides can be downloaded from: https://github.com/aGitHasNoName/amatoLab. Click the green button that says Clone or download, and then Download ZIP.

What is Git? What is GitHub?

Git

- A specialty programming language
- Tracks changes in code and other files
- "Version control" – you can always go back to a previous version
- Easily post code and files on GitHub

GitHub

- A website for posting code and other files
- Allow others to copy your code
- Allow others to contribute to your code
- Find useful code to copy

Why use Git and GitHub?

Advanced:

- Co-development of software

Basic:

- Avoid losing versions
- Back up your code and files
- Access code from anywhere
- Share code with coauthors
- Publish code following publication of research
- Organize code and files by project

vocab: REPOSITORY

A folder dedicated to one project. AKA "repo"

Concept

Project folder on your computer AKA local repo

- Data
- Code

Concept

Project folder on your computer

- Data
- Code

`git init` creates a hidden file that allows git to track changes in this folder

Concept

Project folder on your computer

- Data
- PythonCode

you make changes in the project folder

Concept

Project folder on your computer

- Data
- PythonCode

`git add` adds those changes to a git staging area

Concept

Project folder on your computer

- Data
- PythonCode

`git commit` commits all changes in the staging area in bulk as a "commit"

Concept

Project folder on your computer

- Data
- PythonCode

Repo on GitHub website

AKA remote repo

Concept

Project folder on your computer

Repo on GitHub website

Data

PythonCode

`git remote add` link the remote repo to your local repo

Concept

Project folder on your computer

- Data
- PythonCode

Repo on GitHub website

- Data
- PythonCode

`git push` push all local commits to the remote repo, so that they match

Concept

Project folder on your computer

- Data
- PythonCode

Repo on GitHub website

- ProjectData
- PythonCode

you make changes in the online repo

Concept

Project folder on your computer

- ProjectData
- PythonCode

Repo on GitHub website

- ProjectData
- PythonCode

`git pull` pull all remote changes to the local repo, so that they match

vocab: PUSH

Copy changes in a repo
from your computer
to your GitHub site

vocab: PULL

Copy changes in a repo from your GitHub site to your computer

Concept

Project folder on your computer

Repo on GitHub website

- ProjectData
- PythonCode
- .gitignore

- ProjectData
- PythonCode

you can create a hidden file called .gitignore that lists any files or folders that you don't want to track changes on and that you don't want to push to GitHub

.gitignore

```
ProjectData
```

Concept

Project folder on your computer

Repo on GitHub website

ProjectData

PythonCode

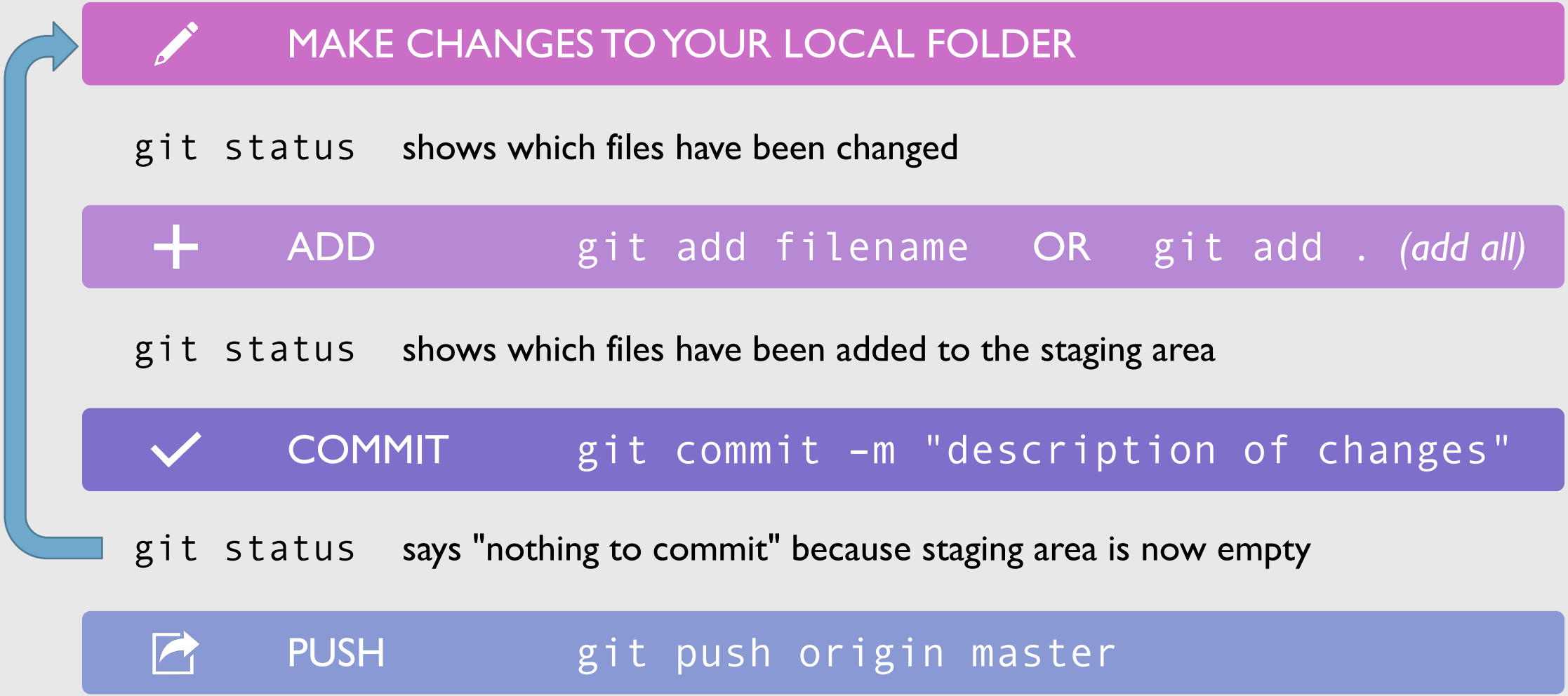
.gitignore

PythonCode

.gitignore

`git push` add, commit, and push all local commits to the remote repo

Normal workflow


MAKE CHANGES TO YOUR LOCAL FOLDER
git status shows which files have been changed
ADD git add filename OR git add . (add all)
git status shows which files have been added to the staging area
COMMIT git commit -m "description of changes"
git status says "nothing to commit" because staging area is now empty
PUSH git push origin master

What to post on GitHub

Post publicly:

- Code (scripts, Jupyter notebooks) used in research and side projects
- Markdown files (table of contents, readmes)
- Published papers

Post privately:

- Code and manuscripts related to unpublished research
- You can still grant access to other individuals

Don't post:

- Large data files
- API keys, log-in credentials, or other private data

What to post on GitHub

Your GitHub

Private

- unpublished research

Public

- published research

Lab GitHub

Private

- lab policies and protocols
- Table of Contents

Public

- published research
- Table of Contents

What to post on GitHub

Lab GitHub

Private

- lab policies and protocols (unlike box or google drive, no one can permanently change these, everyone can access them from web without NU log-in)
- Table of Contents – links to lab members' projects

Public

- published research (forked repos from individual's sites)
- Table of Contents – links to each project, links to lab webpage, contact info, links to any policies you want to be public

LET'S LOOK AT MY GITHUB:
HTTPS://GITHUB.COM/AGITHASNONAME

SEARCHING FOR HELPFUL CODE ON GITHUB

vocab: FORK

Copy a repo

from someone else's GitHub site

to your GitHub site

(Allows you to collaborate with the original user's version of the repo)

vocab: CLONE

Copy a repo

from a GitHub site

to your computer

(Does not allow interaction with the original repo)

README.md

- A README file is expected to be included with each repo (in reality, most people don't take the time to make one for each repo, but you should)
- It is written in markdown.
- You can create it and edit it on GitHub or in your own text editor.
- It contains information about your repo – often this means stating the purpose of the repo and/or including a Table of Contents to the items in your repo. Can also include links of interest, contact information, installation instructions, etc.

Licenses

If you do not choose a license for your GitHub repo, no license technically means no one should use it without getting your permission.

https://choosealicense.com/

Where to enter git commands

We use the command line. You can access command line in a shell.

Mac users: open Terminal

PC users: open Anaconda prompt (if you have it) or Windows PowerShell

Where to enter git commands

You will need to know some basic "command line" to move around between folders on your computer

The command line language on Macs is Bash and on PCs is PowerShell. There are some similar commands between the two.

Where to enter git commands

print your current working directory:

```
pwd
```

change to a lower directory:

```
cd Documents
```

change to a directory one level higher:

```
cd ..
```

Git and GitHub tutorial

https://swcarpentry.github.io/git-novice/

Git and GitHub tutorial

Today we are doing Lesson 2: Setting Up Git

Git and GitHub tutorial

Please complete Lessons 1 and Lessons 3-8 on your own.

For Lesson 9: Conflicts, you will need to work with a partner.

Lessons 10-14 are optional

2. Setting up Git

```
git config --global user.name "Vlad Dracula"
git config --global user.email vlad@tran.sylvan.ia

Mac and Linux users:
git config --global core.autocrlf input
Windows users:
git config --global core.autocrlf

git config --global core.editor "nano -w"
```

(Full list of text editors: https://git-scm.com/book/en/v2/Appendix-C%3A-Git-Commands-Setup-and-Config)