Unit 1: Introduction to Python and Github

Case Study:
*In this lecture we will "download", edit, and "submit" a **Jupyter notebook** to and from **Github repositories** located on our STAT207 **Github Enterprise organization**.*

Purpose of this Lecture:

In this lecture we will cover the following topics.

1. Overview of data science software and platforms we will use.
2. What is **version control** and why use a **version control system**, like **Git**?
3. What is a **Git repository?**
4. What is a **branch** in a Git repository?
5. What is **Github**?
6. What is a **Github Enterprise Organization**?
7. Exploring the repositories on our **STAT207 Github Enterprise Organization**
8. How to "initially download" (ie. **clone**) a virtual repository to your local computer.
9. How to "download" files/folders (or updates to files/folders) (ie. **fetch** and **merge)** from a virtual repository to a local repository on your computer.
10. What is a **Jupyter notebook** and how do you use and edit one?
11. How to "submit" materials from your local computer to your private netid remote repository on Github.

Additional resources: https://www.javatpoint.com/git

STAT107 Review about Git: https://discovery.cs.illinois.edu/learn/Basics-of-Data-Science-with-Python/Software-Version-Control-with-git/

1. Overview of data science software and platforms we will use.

The instructions in your first lab http://courses.las.illinois.edu/fall2022/stat207/labs/01-intro.html

will tell you how to download and get familiar with each of these software and platforms.

Main Purpose in Class	Software/Platform	What is it?
Coding Lab Assignments in Python	Anaconda	**Distribution** of the Python and R programming languages. Allows you to download and run popular Python packages and the Jupyter Notebook Application.
	Jupyter Notebooks	**Python application** that allows you to write data science reports that also need to be integrated with interactive Python code blocks.
	Python	**A programming language**
Version Control: practice of tracking and managing changes to code.	Git	Version control **system.**
	Github	Git repository **hosting service.**
	Github Enterprise STAT207 Organization 7-	**A collection of user accounts** (users=you, your classmates, Dr. Ellison, TAs, and Cas) **that owns Github repositories.**
	Command Line Interface	An **application** that processes commands to a computer program in lines of text.

Anaconda Prompt (Miniconda3)

(base) C:\Users\vellison>

localhost:8888/tree/Desktop/stat207/vellison

jupyter

Files   Running   Clusters

Select items to perform actions on them.

Desktop / stat207 / vellison

Test_Notebook.ipynb

README.md

jupyter Test_Notebook Last Checkpoint: Last Friday at 12:34 AM (unsaved changes)

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Markdown

Test Notebook

Let's make sure everything is working by modifying and running the code below, and the

Change my name to your name and run the code blocks below.

In [1]: student_name = 'tori_ellison'
         student_name
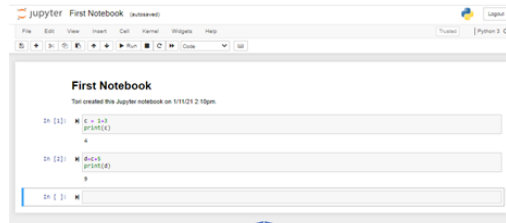
Out[1]: 'tori_ellison'

WHAT IS **VERSION CONTROL** AND WHY USE A **VERSION CONTROL SYSTEM** LIKE **GIT**?

Version control is a class of _____ responsible for managing _____ to computer programs,

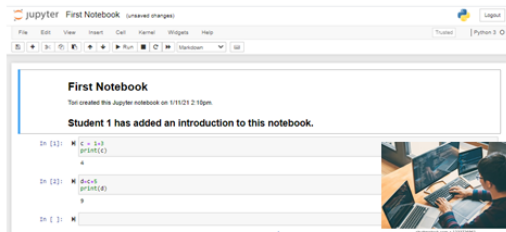documents, large web sites, or other collections of information.

Git is an open-source distributed version control system.
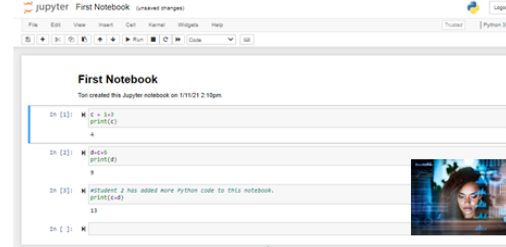
Working in Groups

Initial Version of Notebook
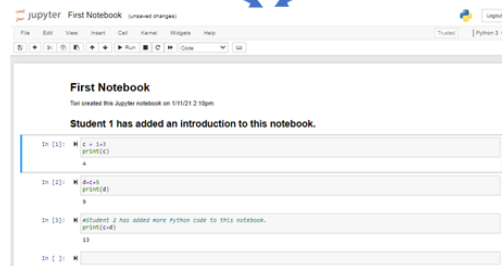

Student 1 downloads and makes edits to this notebook

Student 2 downloads and makes edits to this notebook

Git

Edits from Students 1 and 2 have been automatically combined

Tracking changes and being able to revert back to old versions

https://opensource.com/article/18/6/git-reset-revert-rebase-commands

WHAT IS A GIT REPOSITORY?

A git **repository** is a collection of _____ as well as a history of _____ made to these files and subfiles. You can

think of a repository as a _____

Example of a Git Repository:

The _*release repository*, found here https://github.com/illinois-cs-coursework/fa22_stat207_.release, will contain the materials that you need to "download" to work on the lab assignments in this class.
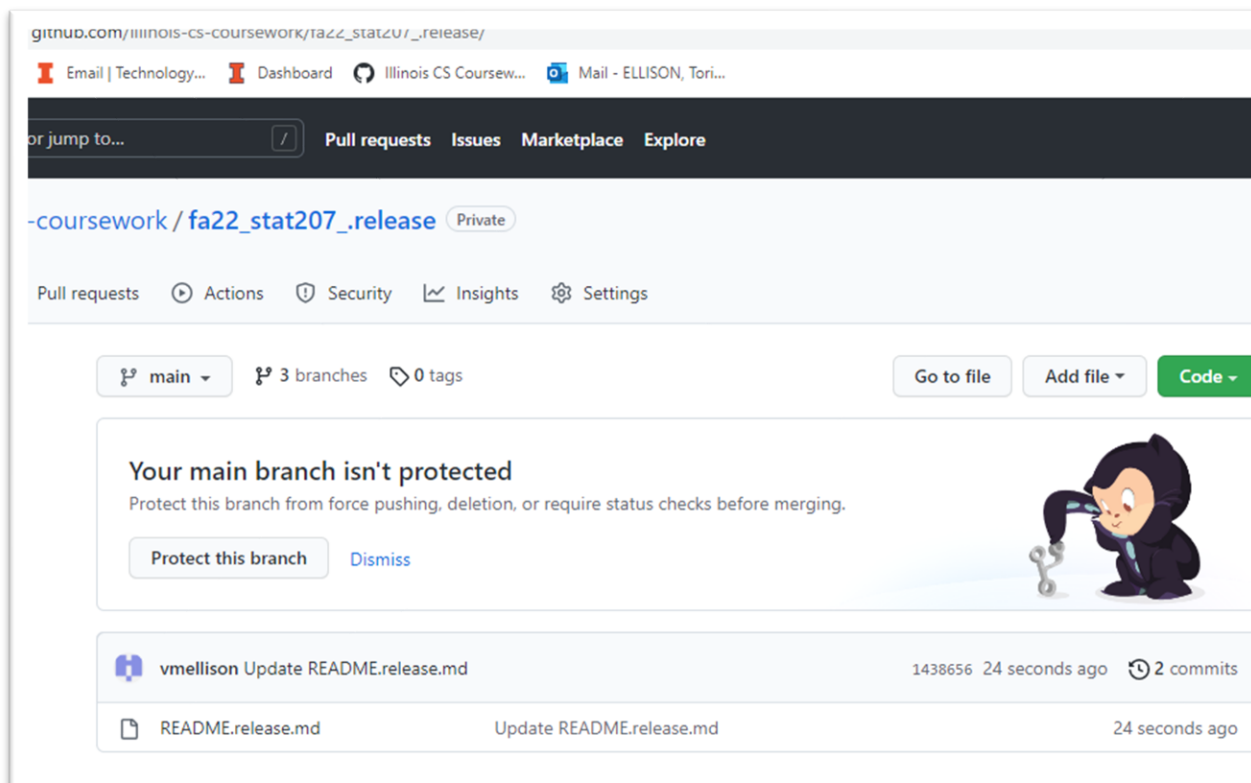
What is a Git Repository **Branch?**

A git repository **branch** is a _____ of the repository that diverges from the <u>main</u> working project. You can

think of a git repository branch as a _____.

Example: Check out the branches we have so far in the https://github.com/illinois-cs-coursework/fa22_stat207_.release repository. Each lab assignment will show up in it's own branch.
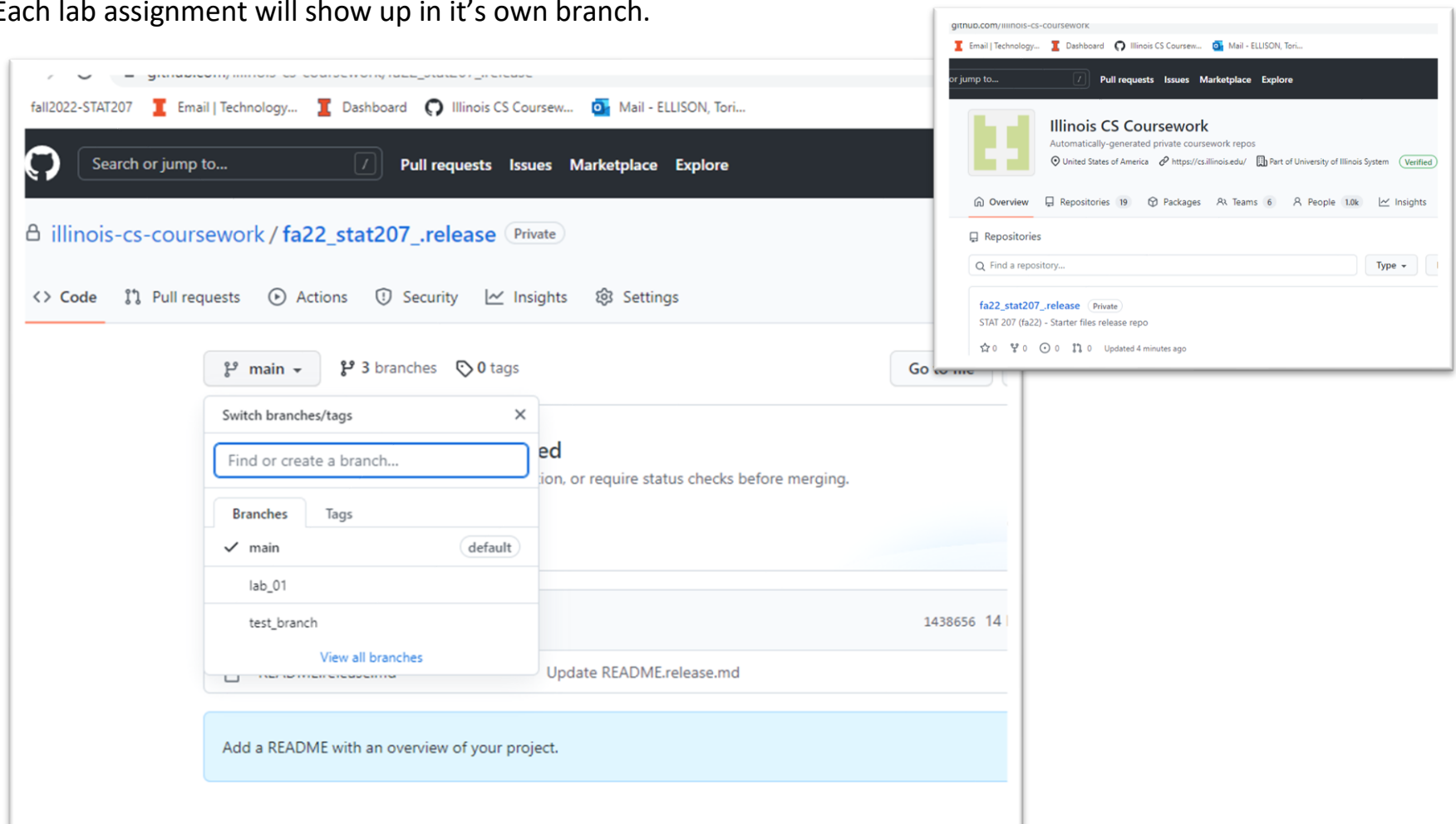
Example: Notice how the lab_01 has different files in it than the main branch. The main branch is the default, and Git will assume that this is the branch you are referring to unless you specify otherwise.
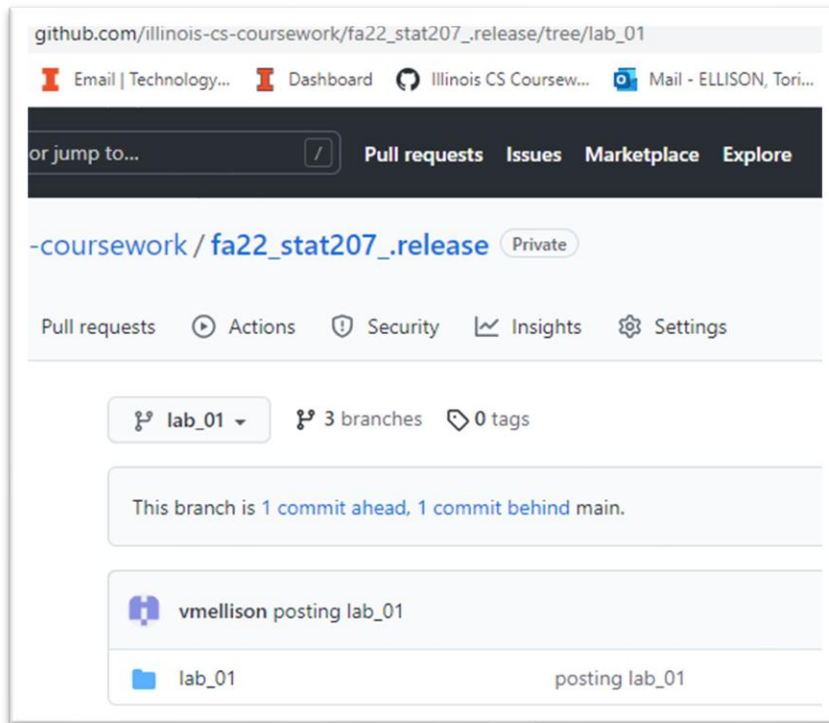
WHAT IS GITHUB?

Github is a web-based service that hosts _____.
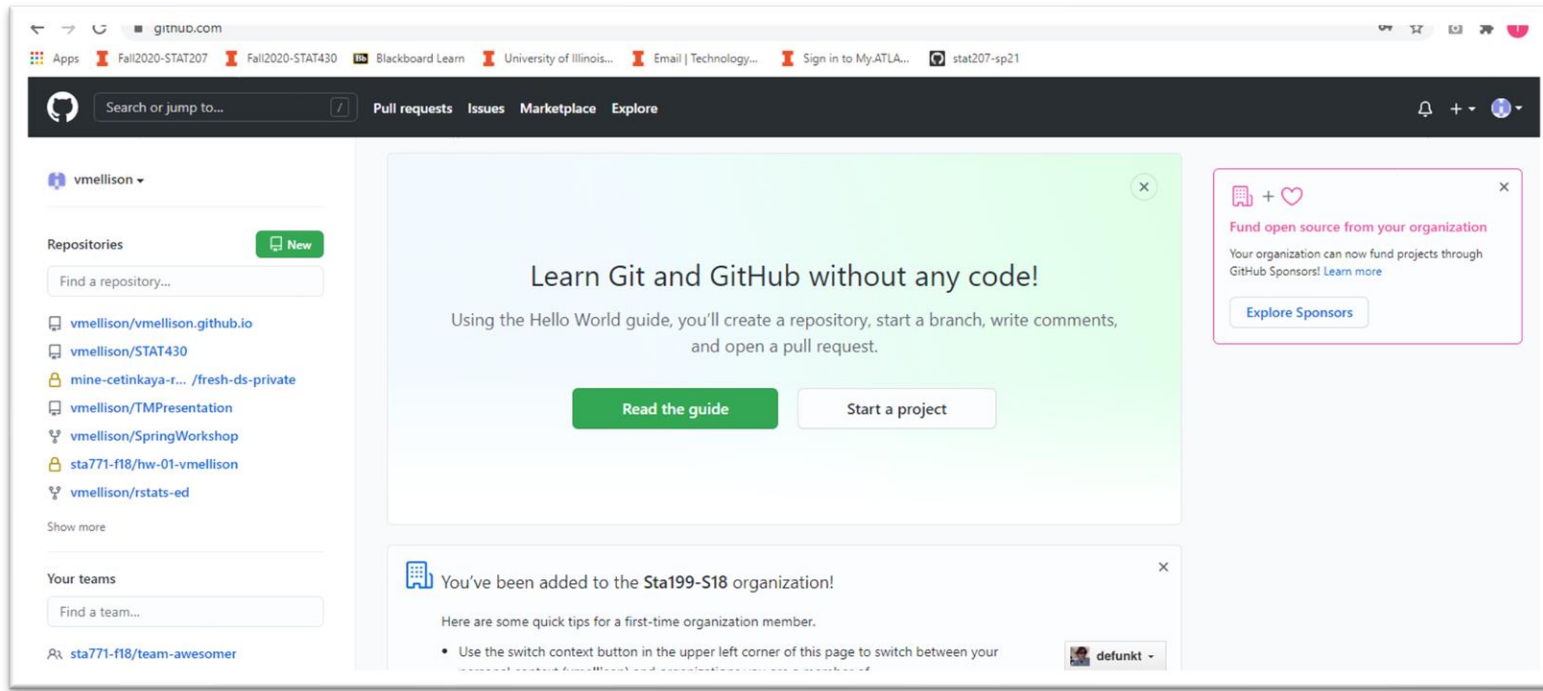
WHAT IS A GITHUB ENTERPRISE ORGANIZATION?

A **Github Enterprise Organization** is a set of shared accounts and repositories, that is managed by owners and administrators.

For example: You can find the UIUC Coursework Github Enterprise organization here: https://github.com/illinois-cs-coursework

Below is an example of what students will see (after you have clicked https://edu.cs.illinois.edu/create-gh-repo/fa22_stat207/ to create your **private netid repository** in the UIUC Coursework Github Enterprise organization).

What the TAs and I will see...

How do these two class repositories work and what will they be used for?
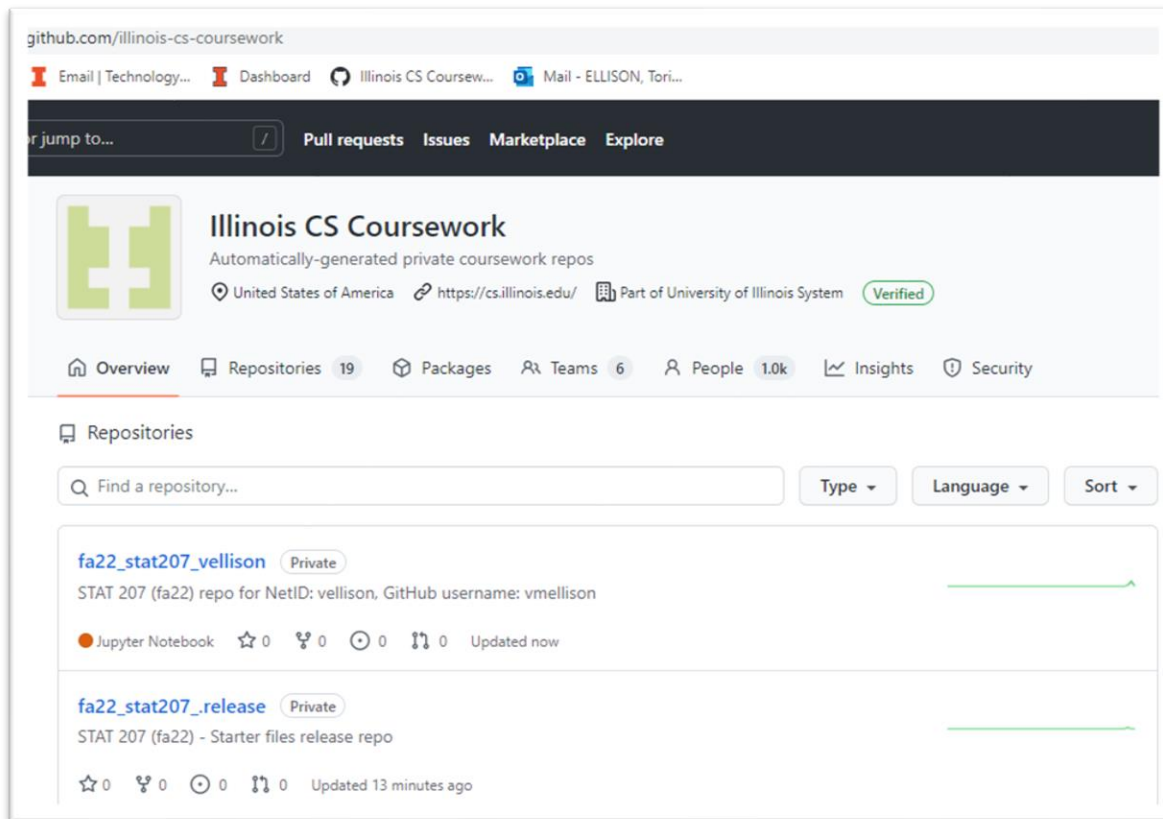
Note: Your private netid repository starts out with just one branch: **main** (ie. the default). To keep things easier in this class, you should probably not try to create additional branches in your netid repository. The TAs will be looking in your **main branch** for your lab assignment submissions.

In general:

To make a copy of a repository on your local computer you can use

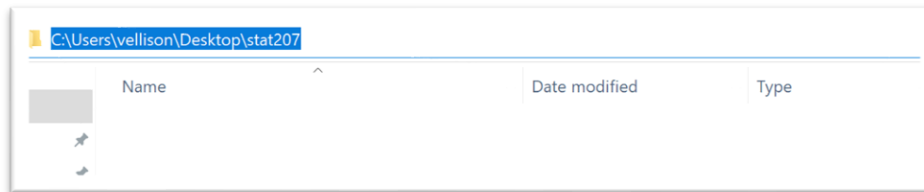**git clone <insert repository link> <insert "nickname" for repository>**

in the command line in the folder you would like to store it in.

Example: Clone your *netid repository*

The instructions in your first lab assignment will discuss this in more detail (http://courses.las.illinois.edu/fall2022/stat207/git.html).

Local folder

You will edit and download your STAT207 materials in this folder on your local computer here.

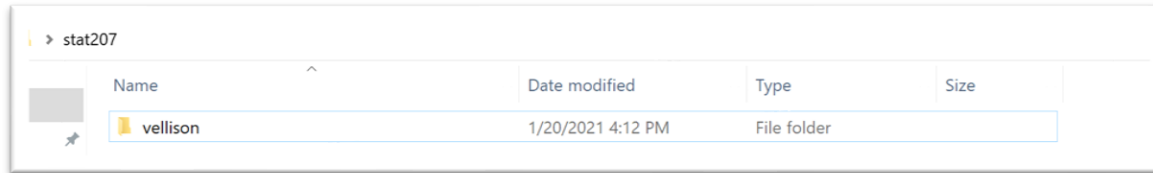|			
C:\Users\vellison\Desktop\stat207			
	Name	Date modified	Type

In Command Line…

Git CMD — □

```
C:\Users\vellison>cd desktop

C:\Users\vellison\Desktop>cd stat207

C:\Users\vellison\Desktop\stat207>git clone https://github.com/illinois-cs-coursework/fa22_stat207_vellison vellison
```

After running the command…

Whatever was in your *netid repository* (ie. nothing if you just created it) will show up in this folder named with your netid.

CREATING A NICKNAME FOR ANOTHER REPOSITORY

The instructions in your first lab assignment will discuss this in more detail
(http://courses.las.illinois.edu/fall2022/stat207/git.html).

In Command Line...

```
C:\Users\vellison\Desktop\stat207>cd vellison

C:\Users\vellison\Desktop\stat207\vellison>git remote add release https://github.com/illinois-cs-coursework/fa22_stat207_.releas
e.git
```

Once you are in the cloned local repository, the

    git remote add <insert nickname> <insert organization link>/<repo_name>.git

command creates a nickname for the specified repository.

HOW TO "DOWNLOAD" FILES/FOLDERS OR UPDATES TO FILES/FOLDERS (IE. **FETCH** AND **MERGE)** FROM A VIRTUAL REPOSITORY TO YOUR LOCAL REPOSITORY ON YOUR COMPUTER.

General Commands: (these commands only work after you have **cloned** a repository and your git command line is in that repository).

- **git fetch** <insert repository name>
  - Transmits the *"whole version" of the REMOTE repository* to your LOCAL computer.

- **git merge** <insert repository name>/<insert branch name> **-m** "notes about merge"
  - Merges just the *changes made to the specified **branch*** of the remote repository.

For example: In the command line code below we **fetch** the virtual **_release repository**. And then we **merge** the following:
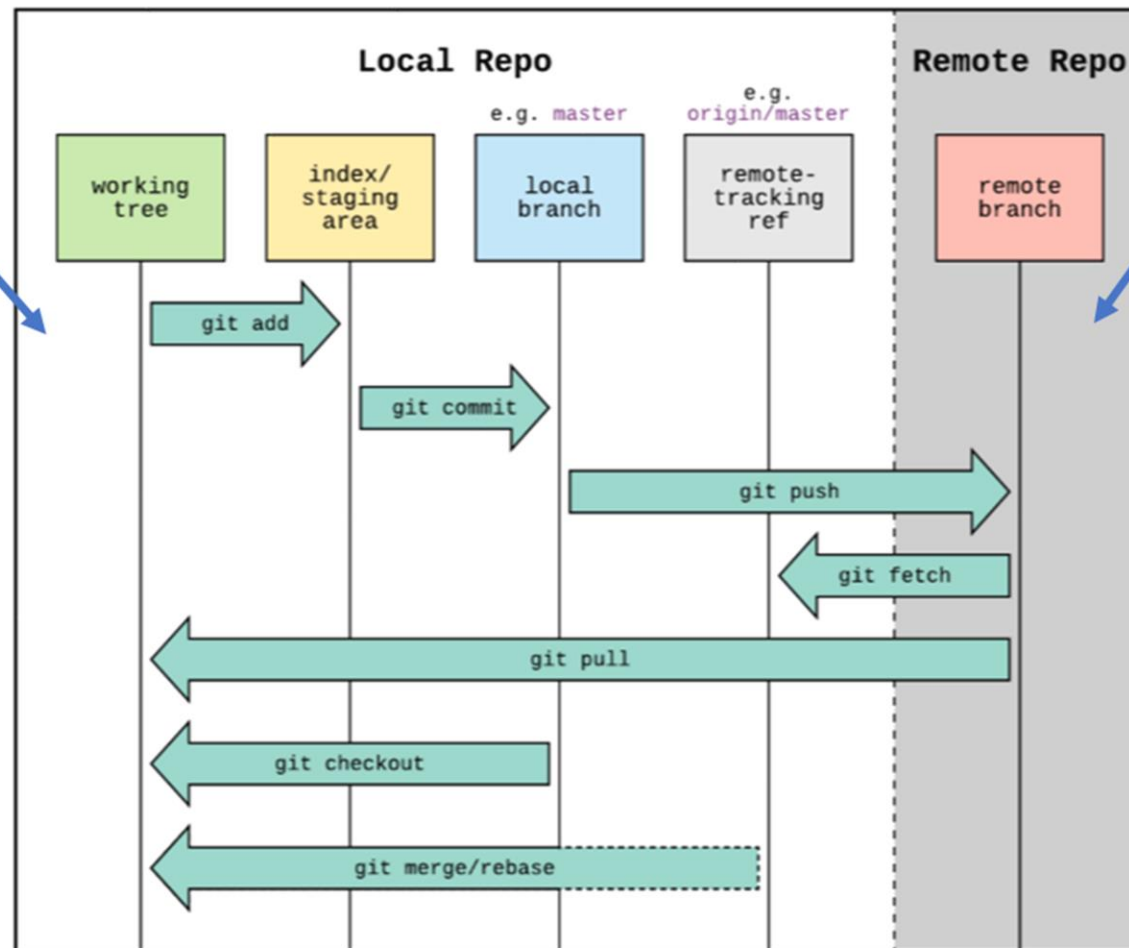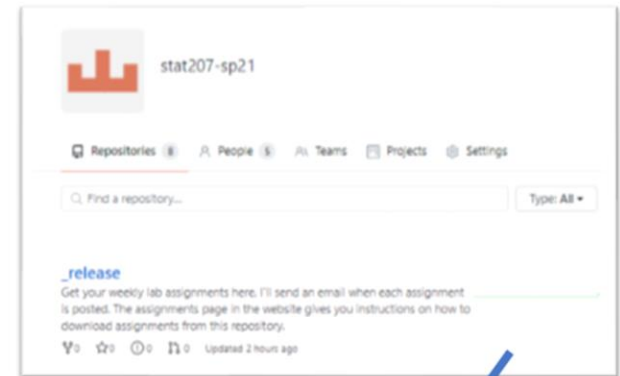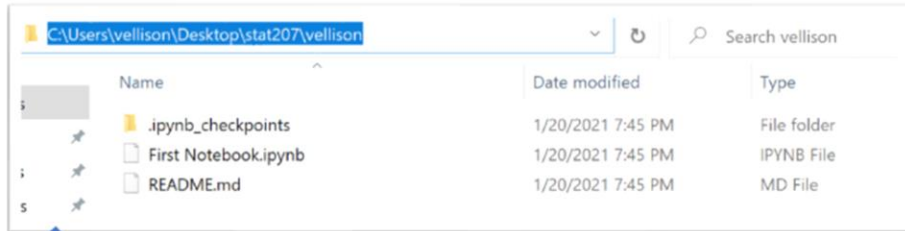
- what is in the demo_branch branch of the virtual **_release repository**
- what is in the main branch of your local **netid repository.**

These changes will show up in the folder that your local netid repository resides in.

```
Git CMD

C:\Users\vellison>cd desktop

C:\Users\vellison\Desktop>cd stat207

C:\Users\vellison\Desktop\stat207>cd vellison

C:\Users\vellison\Desktop\stat207\vellison>git fetch release
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 20 (delta 4), reused 5 (delta 1), pack-reused 0
Unpacking objects: 100% (20/20), 2.75 KiB | 8.00 KiB/s, done.
From https://github-dev.cs.illinois.edu/stat207-sp21/_release
 * [new branch]      demo_branch -> release/demo_branch
 * [new branch]      lab_01      -> release/lab_01
 * [new branch]      master      -> release/master
C:\Users\vellison\Desktop\stat207\vellison>git merge release/demo_branch -m "Merging initial files"

C:\Users\vellison\Desktop\stat207\vellison>_
```

What shows up in this local folder will be a **merging** of what is
in the remote demo_branch branch of the _release repository
and whatever you currently have in your local copy of your
netid repository (most likely master branch)

C:\Users\vellison\Desktop\stat207\vellison   Search vellison

Name	Date modified	Type
.ipynb_checkpoints	1/20/2021 7:45 PM	File folder
First Notebook.ipynb	1/20/2021 7:45 PM	IPYNB File
README.md	1/20/2021 7:45 PM	MD File

stat207-sp21

Repositories 8   People 5   Teams   Projects   Settings

Find a repository...   Type: All ▾

_release
Get your weekly lab assignments here. I'll send an email when each assignment
is posted. The assignments page in the website gives you instructions on how to
download assignments from this repository.

⑂0 ☆0 ⊙0 �𝄃0 Updated 2 hours ago

Local Repo   Remote Repo

e.g. master   e.g.
                origin/master

| working tree | index/ staging area | local branch | remote-tracking ref | remote branch |

git add →

git commit →

git push →

← git fetch

← git pull

← git checkout

← git merge/rebase

WHAT IS A **JUPYTER NOTEBOOK** AND HOW DO YOU USE AND EDIT ONE?

1. First, check out the instructions in your first lab http://courses.las.illinois.edu/fall2022/stat207/labs/01-intro.html for how to download **minicondas** software which contains the **Jupyter notebook application.**

2. Start your **Jupyter notebook application**
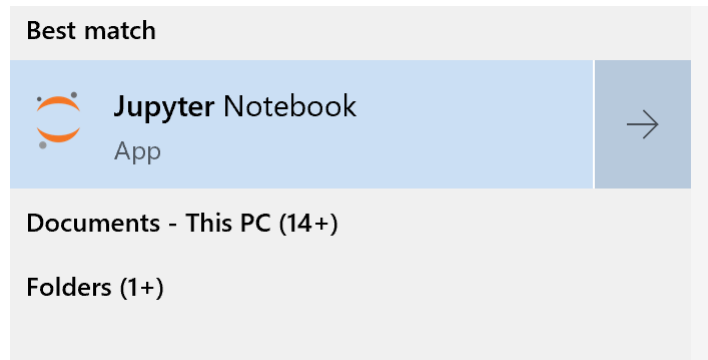
3. This will open a browser which displays your computer's file system. Use this file system display like you would on your computer to navigate to the folder on computer where your recently downloaded **Jupyter notebook (.ipynb)** is located. Click on the file in this screen to open and edit the notebook.

4. **Edit the notebook.**

5. **Save any changes made to the notebook.**
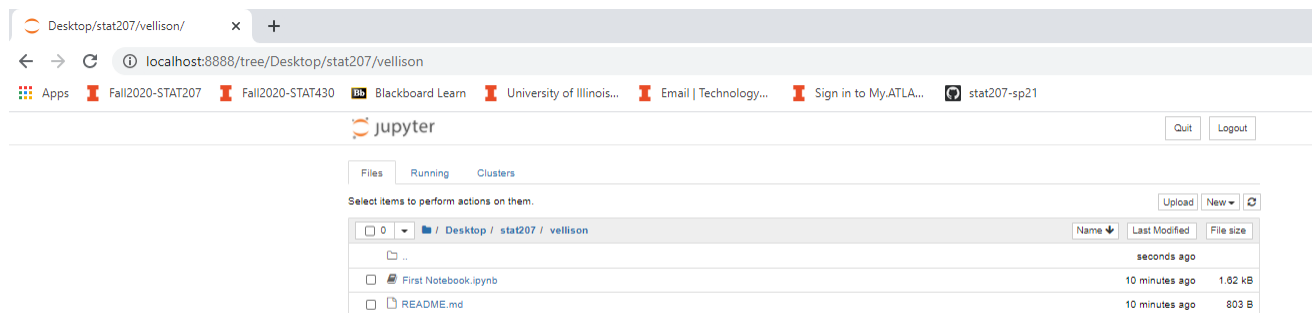
localhost:8888/notebooks/Desktop/stat207/vellison/First%20Notebook.ipynb

Apps   Fall2020-STAT207   Fall2020-STAT430   Blackboard Learn   University of Illinois...   Email | Technology...   Sign in to My.ATLA...   stat207-sp21

Jupyter   First Notebook   Last Checkpoint: 13 minutes ago   (unsaved changes)   Logout

File   Edit   View   Insert   Cell   Kernel   Widgets   Help   Trusted | Python 3 ○

New Notebook ▸
Open...

Make a Copy...
Save as...
Rename...
Save and Checkpoint   Ctrl-S

Revert to Checkpoint ▸

Print Preview
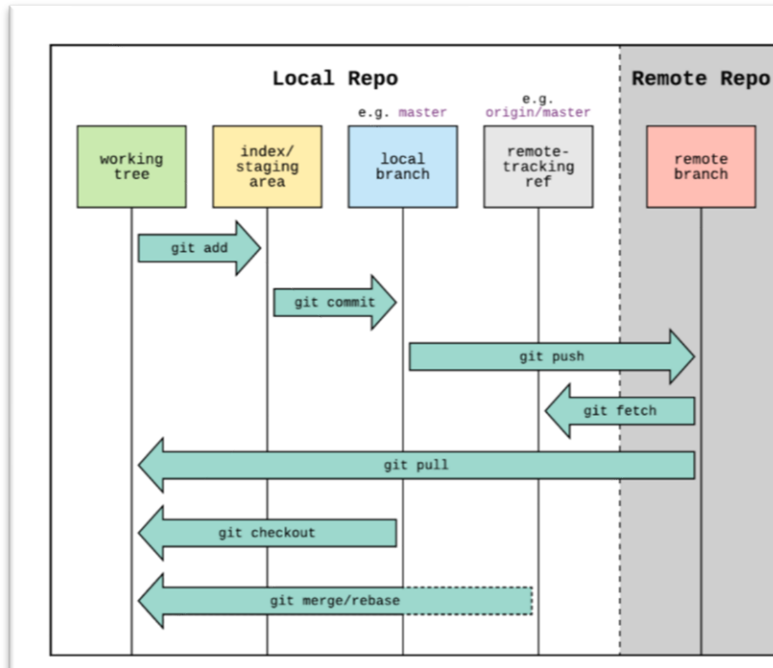Download as ▸

Trusted Notebook

Close and Halt

▶ Run ■ C ▶▶   Code ▾

otebook

s Jupyter notebook on 1/11/21 2:10pm.

In [3]: ▸   #Student 2 has added more Python code to this notebook.
print(c+d)

13

In [1]: ▸   print('hello world')

hello world

In [ ]: ▸

HOW TO "SUBMIT" MATERIALS (OR CHANGES TO MATERIALS) FROM YOUR LOCAL COMPUTER BACK TO A REPOSITORY.
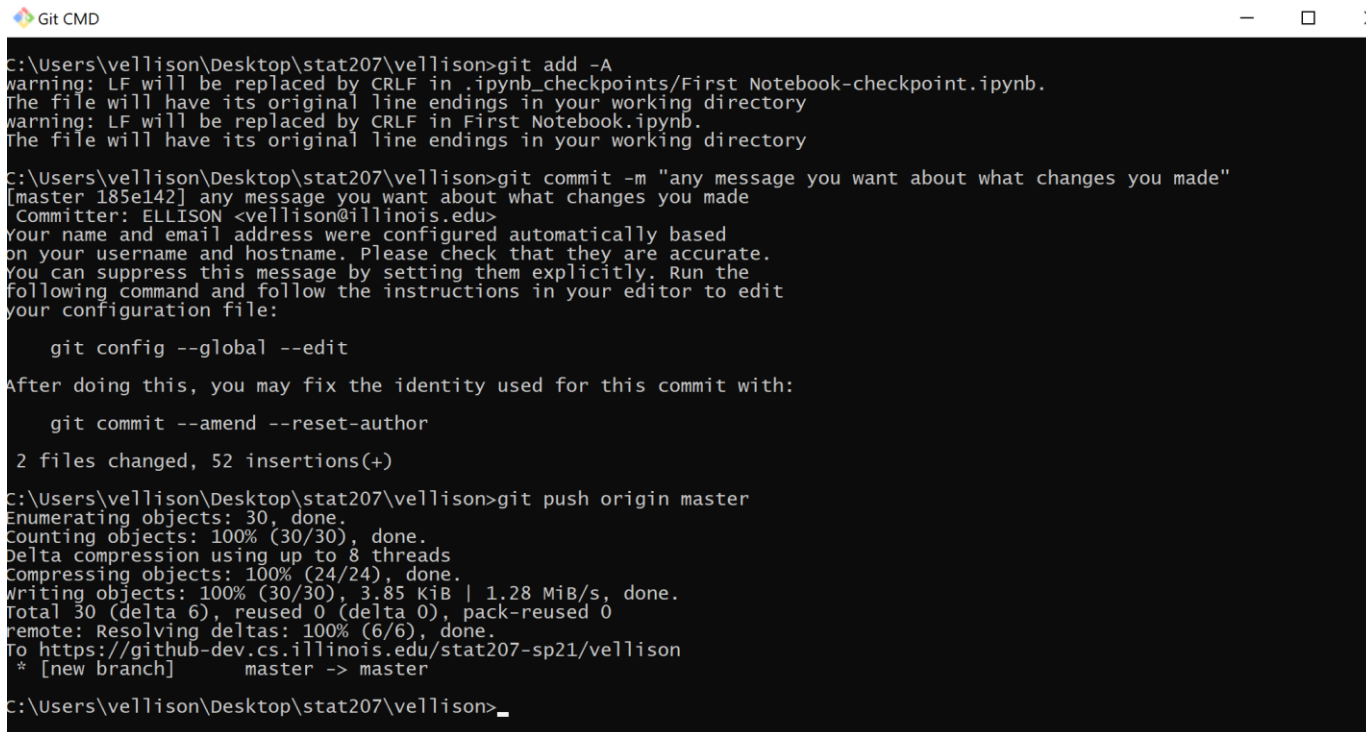
In general

Commands:

- **git add –A**
  - Adds your changes to the **index/staging area** (this is a "rough draft space."

- **git commit –m "message explaining your changes"**
  - Adds a "snapshot of your project" (with changes) to the local branch along with your message.

- **git push origin master**
  - Pushes/overwrites these changes you made back to the REMOTE repository.

Example: **Push your local changes back to your remote netid repository.**

Your first lab http://courses.las.illinois.edu/fall2022/stat207/labs/01-intro.html talks about how to do this in more detail.

```
Git CMD                                                                    —  □  ✕

C:\Users\vellison\Desktop\stat207\vellison>git add -A
warning: LF will be replaced by CRLF in .ipynb_checkpoints/First Notebook-checkpoint.ipynb.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in First Notebook.ipynb.
The file will have its original line endings in your working directory

C:\Users\vellison\Desktop\stat207\vellison>git commit -m "any message you want about what changes you made"
[master 185e142] any message you want about what changes you made
 Committer: ELLISON <vellison@illinois.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 2 files changed, 52 insertions(+)

C:\Users\vellison\Desktop\stat207\vellison>git push origin master
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (30/30), 3.85 KiB | 1.28 MiB/s, done.
Total 30 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), done.
To https://github-dev.cs.illinois.edu/stat207-sp21/vellison
 * [new branch]      master -> master

C:\Users\vellison\Desktop\stat207\vellison>_
```

Check your remote repository to double check that your changes were submitted.