



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.

- Overview of data science software and platforms we will use.
- What is **version control** and why use a **version control system**, like **Git**?
- What is a **Git repository**?
- What is a **branch** in a Git repository?
- What is **Github**?
- What is a **Github Enterprise Organization**?
- Exploring the repositories on our **STAT207 Github Enterprise Organization**
- How to “initially download” (ie. **clone**) a virtual repository to your local computer.
- 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.
- What is a **Jupyter notebook** and how do you use and edit one?
- How to “submit” materials from your local computer to your private netid remote repository on Github.

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

Overview of data science software and platforms we will use.

The instructions in your first lab <http://courses.las.illinois.edu/fall2021/stat207/labs/01-intro.html>

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

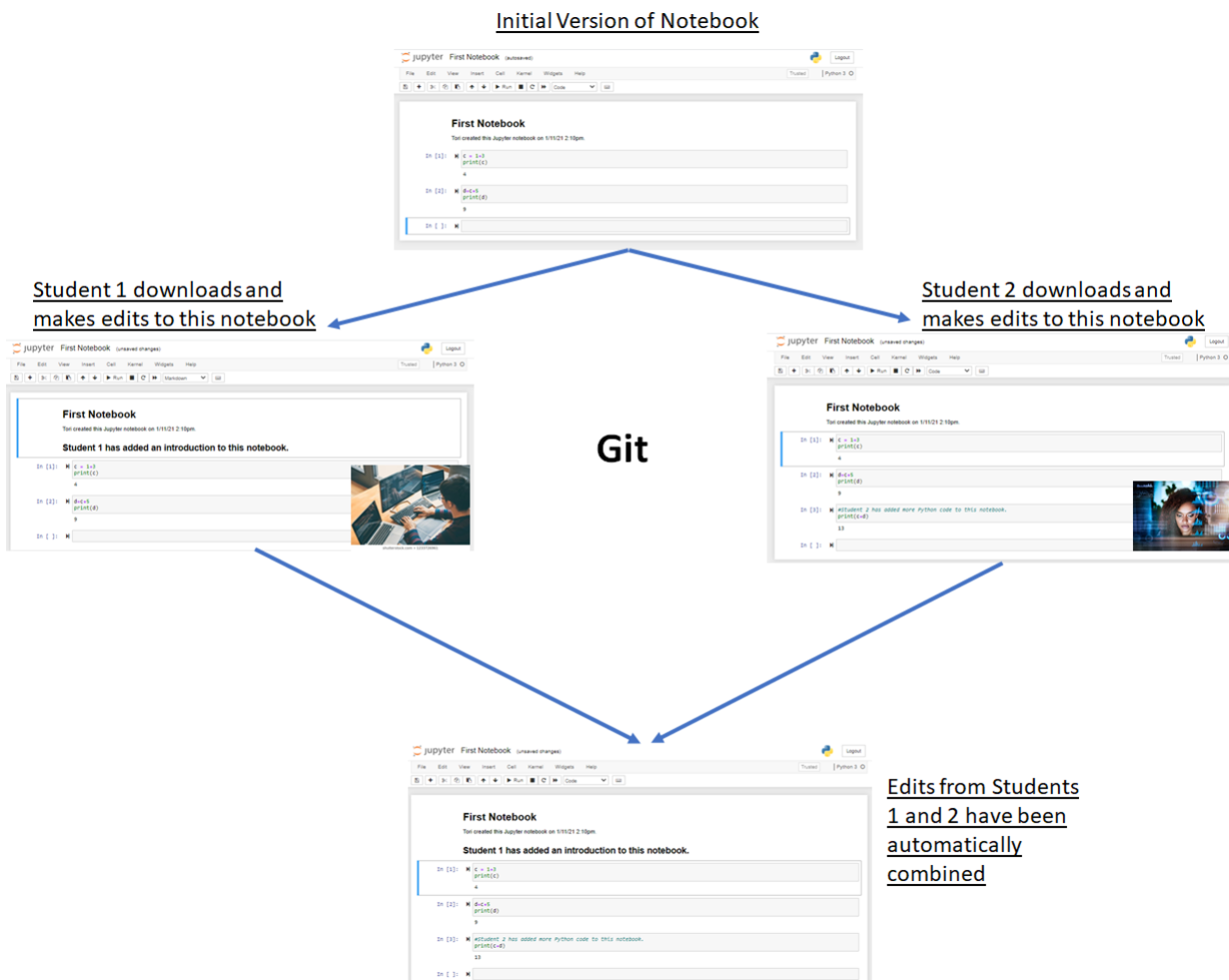
Main Purpose in Class	Software/Platform	What is it?
Coding Lab Assignments in Python	Anaconda	Distribution of the Python and R programming languages. Allows you to download and run popular Python packages and the <u>Jupyter Notebook Application</u> .
	Jupyter Notebooks	Python application that allows you to <u>write data science reports</u> 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 https://github-dev.cs.illinois.edu/stat207-sp21	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.

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



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:

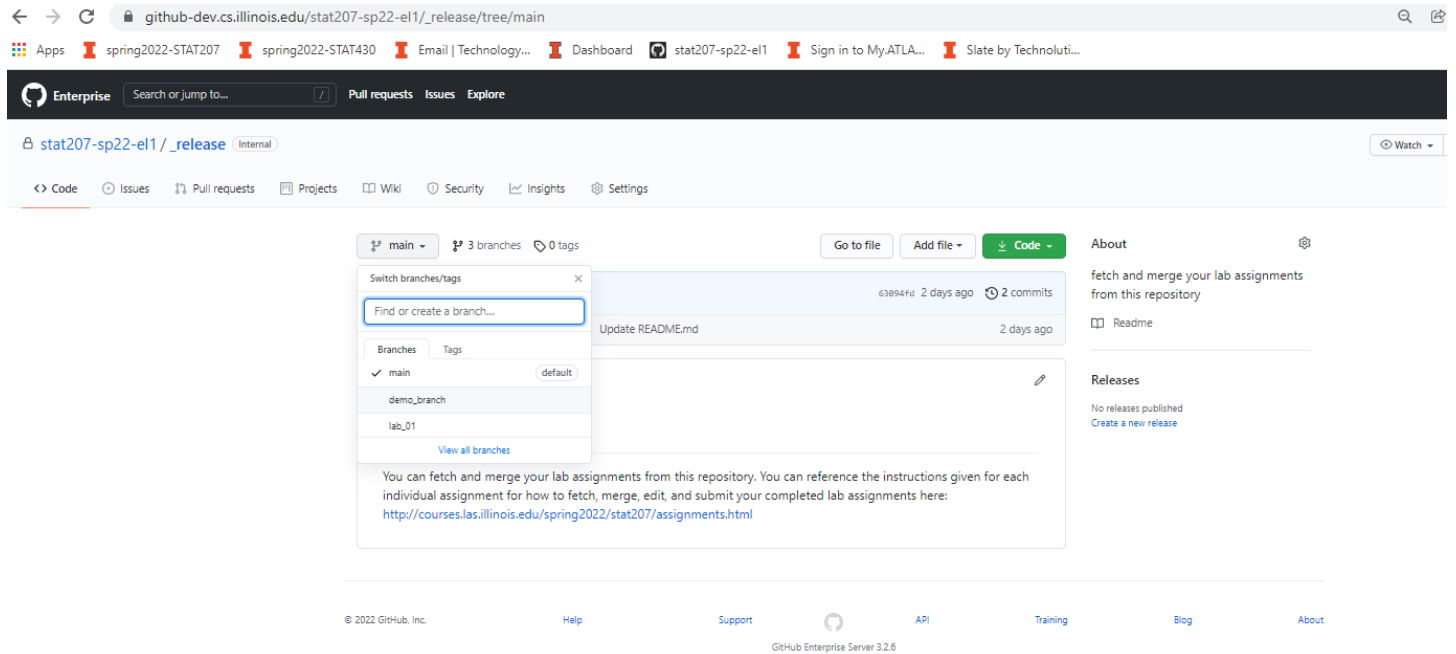
The *_release repository*, found here https://github-dev.cs.illinois.edu/stat207-sp22-el1/_release, will contain the materials that you need to “download” to work on the lab assignments in this class.

The screenshot shows a GitHub repository page for 'stat207-sp22-el1/_release'. The browser address bar shows the URL 'github-dev.cs.illinois.edu/stat207-sp22-el1/_release'. The repository name is 'stat207-sp22-el1 / _release' with an 'Internal' label. The page shows a commit history with one commit: 'vellison Update README.md' by user '63894fd' 2 days ago, with 2 commits. The file 'README.md' is shown with its content: '_release' and instructions on how to fetch and merge lab assignments, including a link to 'http://courses.las.illinois.edu/spring2022/stat207/assignments.html'. The right sidebar contains 'About' (fetch and merge your lab assignments from this repository), 'Readme', and 'Releases' (No releases published, Create a new release). The footer shows '© 2022 GitHub, Inc.', 'Help', 'Support', 'API', 'Training', 'Blog', and 'About', along with 'GitHub Enterprise Server 3.2.6'.

WHAT IS A GIT REPOSITORY BRANCH?

A git repository **branch** is a _____ of the repository that diverges from the main working project. You can think of a git repository branch as a _____.

Example: Check out the branches we have so far in the _release repository. Each lab assignment will show up in it's own branch.



The screenshot shows a GitHub repository page for 'stat207-sp22-el1/_release'. A 'Switch branches/tags' dropdown menu is open, displaying a search bar 'Find or create a branch...', a 'Tags' tab, and a list of branches: 'main' (checked), 'demo_branch', and 'lab_01'. Below the list is a link 'View all branches'. The main content area shows a commit history with 'Update README.md' and a commit ID '63894fd' from '2 days ago' with '2 commits'. The right sidebar contains 'About' (with instructions to 'fetch and merge your lab assignments from this repository'), 'Readme', and 'Releases' (with 'No releases published' and a 'Create a new release' link). The footer includes '© 2022 GitHub, Inc.', 'Help', 'Support', 'GitHub Enterprise Server 3.2.6', 'API', 'Training', 'Blog', and 'About'.

Example: Notice how the demo_branch branch has different files in it than the master branch. The master branch is the default, and Git will assume that this is the branch you are referring to unless you specify otherwise.

demo_branch had recent pushes less than a minute ago [Compare & pull request](#)

demo_branch 3 branches 0 tags [Go to file](#) [Add file](#) [Code](#)

This branch is 1 commit ahead of main. [Contribute](#)

vellison adding notebook 7dca2e 11 seconds ago 3 commits

First Notebook.ipynb adding notebook 11 seconds ago

Add a README with an overview of your project. [Add a README](#)

About

fetch and merge your lab assignments from this repository

Releases

No releases published [Create a new release](#)

WHAT IS GITHUB?

GitHub is a web-based service that hosts _____.

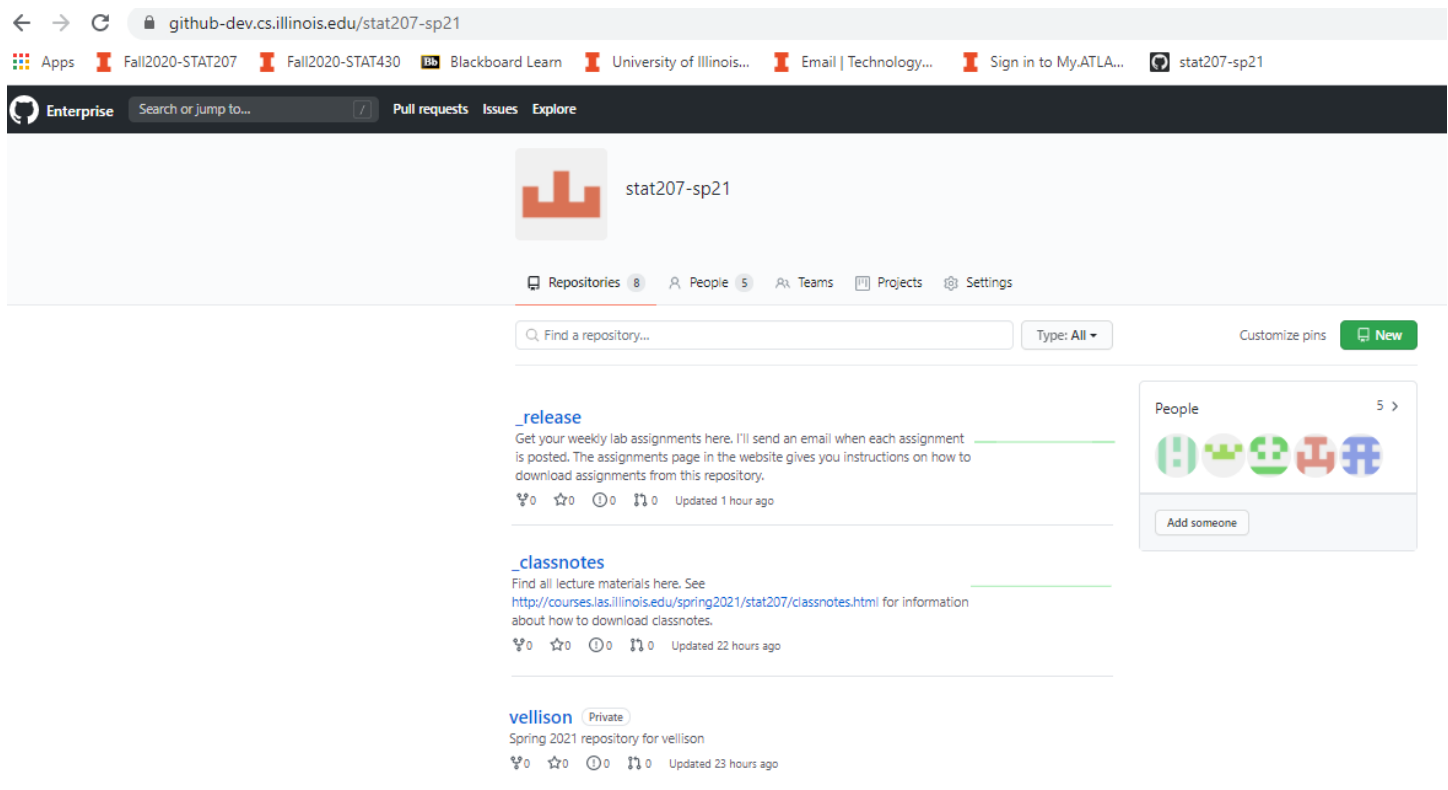
The screenshot shows the GitHub homepage for user **vmellison**. The browser address bar shows `github.com`. The navigation bar includes links for **Pull requests**, **Issues**, **Marketplace**, and **Explore**. The left sidebar shows the user's profile and a list of repositories, including `vmellison/vmellison.github.io`, `vmellison/STAT430`, `mine-cetinkaya-r... /fresh-ds-private`, `vmellison/TMPresentation`, `vmellison/SpringWorkshop`, `sta771-f18/hw-01-vmellison`, and `vmellison/rstats-ed`. The main content area features a large green banner with the text "Learn Git and GitHub without any code!" and two buttons: "Read the guide" and "Start a project". Below the banner is a notification box stating "You've been added to the Sta199-S18 organization!" with a tip to use the switch context button. On the right, there is a pink box titled "Fund open source from your organization" with an "Explore Sponsors" button.

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 our STAT207 Github Enterprise organization here: <https://github-dev.cs.illinois.edu/stat207-sp22-el1>

Below is an example of what students will see (after you have clicked <https://edu.cs.illinois.edu/create-ghe-repo/stat207el1-sp22/> to create your **private netid repository** in the STAT207 Github Enterprise organization).



The screenshot shows the Github Enterprise interface for the organization 'stat207-sp21'. The browser address bar shows 'github-dev.cs.illinois.edu/stat207-sp21'. The navigation bar includes 'Enterprise', a search bar, and links for 'Pull requests', 'Issues', and 'Explore'. The organization name 'stat207-sp21' is displayed with a red bar icon. Below the name are tabs for 'Repositories' (8), 'People' (5), 'Teams', 'Projects', and 'Settings'. A search bar for repositories is present, along with a 'Type: All' dropdown and 'Customize pins' and 'New' buttons. The repository list shows three items: '_release' (updated 1 hour ago), '_classnotes' (updated 22 hours ago), and 'vellison' (private, updated 23 hours ago). A 'People' sidebar on the right shows 5 members and an 'Add someone' button.

What the TAs and I will see...

EXPLORING THE REPOSITORIES ON OUR STAT207 GITHUB ENTERPRISE ORGANIZATION

How do these three class repositories work and what will they be used for?

The screenshot shows a GitHub Enterprise interface for the repository 'stat207-sp21'. The browser address bar shows 'github-dev.cs.illinois.edu/stat207-sp21'. The top navigation bar includes 'Enterprise', a search bar, and links for 'Pull requests', 'Issues', and 'Explore'. Below the repository name, there are tabs for 'Repositories' (8), 'People' (5), 'Teams', 'Projects', and 'Settings'. A search bar for repositories is present, along with a 'Type: All' dropdown and a 'New' button. The repository list shows three items:

- _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. Updated 1 hour ago.
- _classnotes**: Find all lecture materials here. See <http://courses.las.illinois.edu/spring2021/stat207/classnotes.html> for information about how to download classnotes. Updated 22 hours ago.
- vellison** (Private): Spring 2021 repository for vellison. Updated 23 hours ago.

On the right side, there is a 'People' section with 5 members and an 'Add someone' button.

Note: Your private netid repository starts out with just one branch: **master** (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 **master branch** for your lab assignment submissions.

HOW TO “INITIALLY DOWNLOAD” (IE. **CLONE**) A VIRTUAL REPOSITORY TO YOUR LOCAL COMPUTER.

In general:

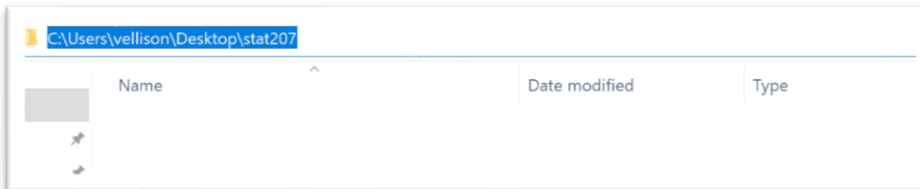
To make a copy of a repository on your local computer you can use **git clone <insert repository link>** in the command line in the folder you would like to store it in.

Example: Clone your *netid* repository

The instructions in your first lab assignment will discuss this in more detail (<http://courses.las.illinois.edu/spring2022/stat207/git.html>).

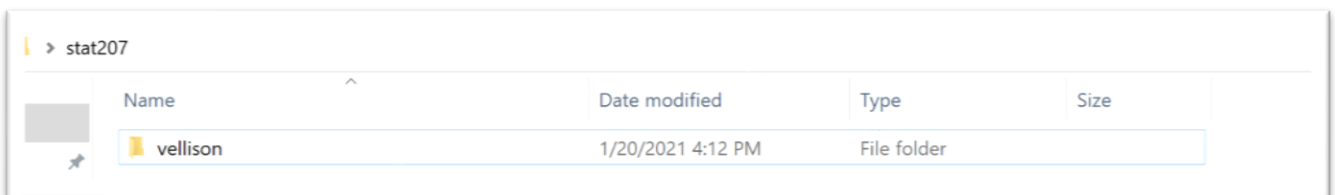
Local folder

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



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-dev.cs.illinois.edu/stat207-sp21/vellison
Cloning into 'vellison'...
warning: You appear to have cloned an empty repository.
```



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/spring2022/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-dev.cs.illinois.edu/stat207-sp21/_release.git
C:\Users\vellison\Desktop\stat207\vellison>
```

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:

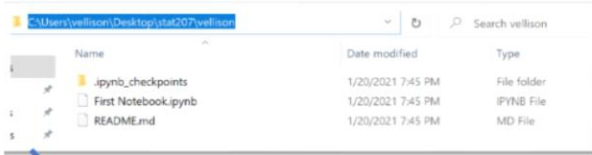
- what is in the demo_branch branch of the virtual _release repository
- what is in the master branch of your local netid repository.

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



```
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** of the **_release repository** and whatever you currently have in your **local copy** of your **netid repository** (most likely **master branch**)



Name	Date modified	Type
.jupyter_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

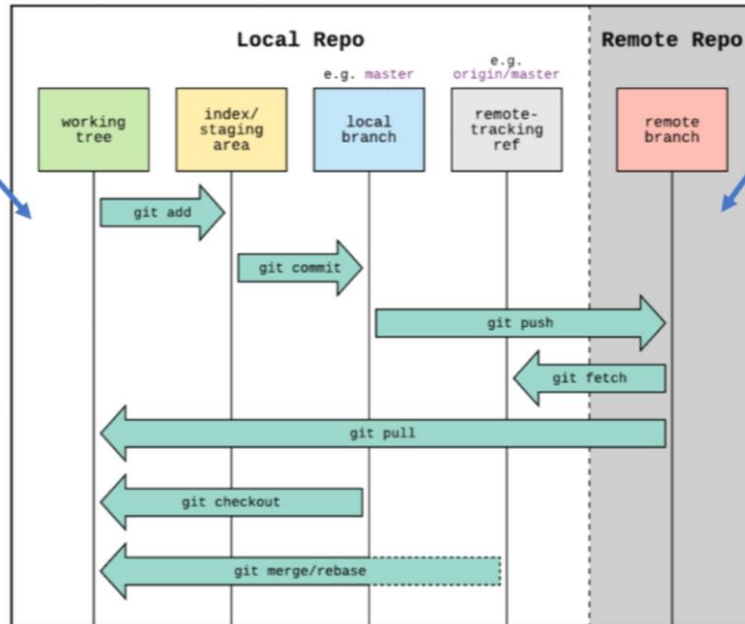
Repositories People Teams Projects Settings

Find a repository... Type All

_release

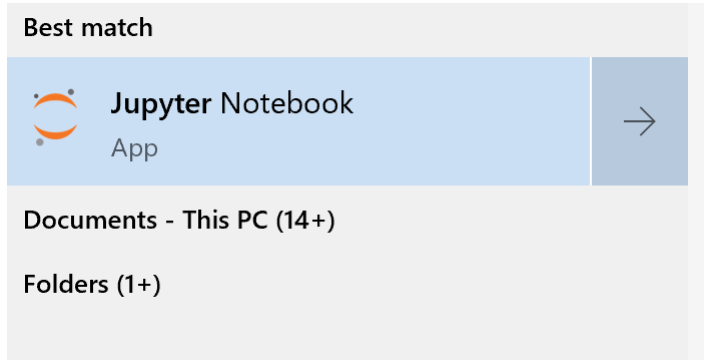
Get your weekly lab assignments here. I'll send an email when each assignment is posted. The assignments page in the website gives you instructions on how to download assignments from this repository.

Updated 2 hours ago

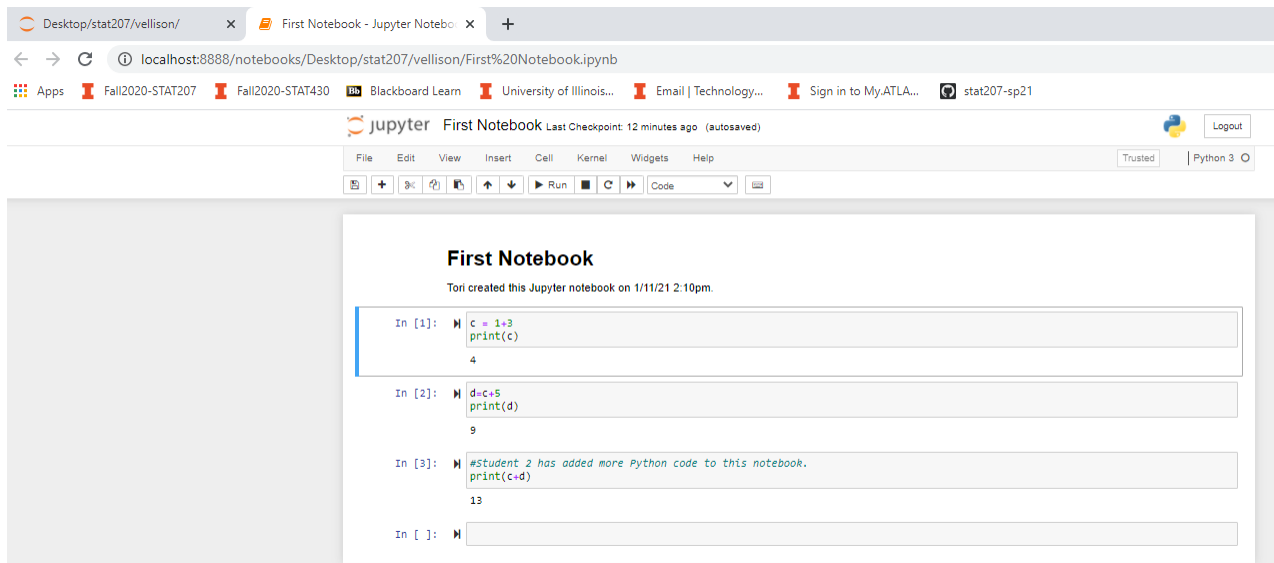
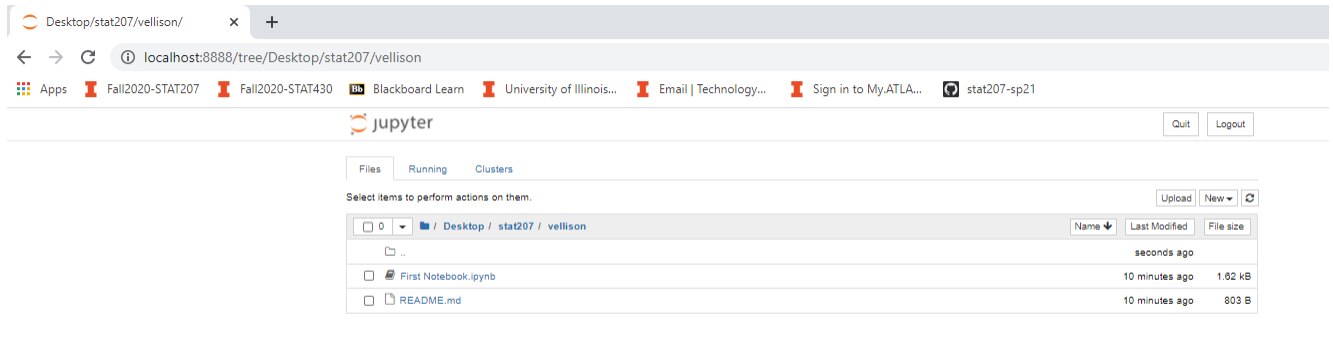


WHAT IS A JUPYTER NOTEBOOK AND HOW DO YOU USE AND EDIT ONE?

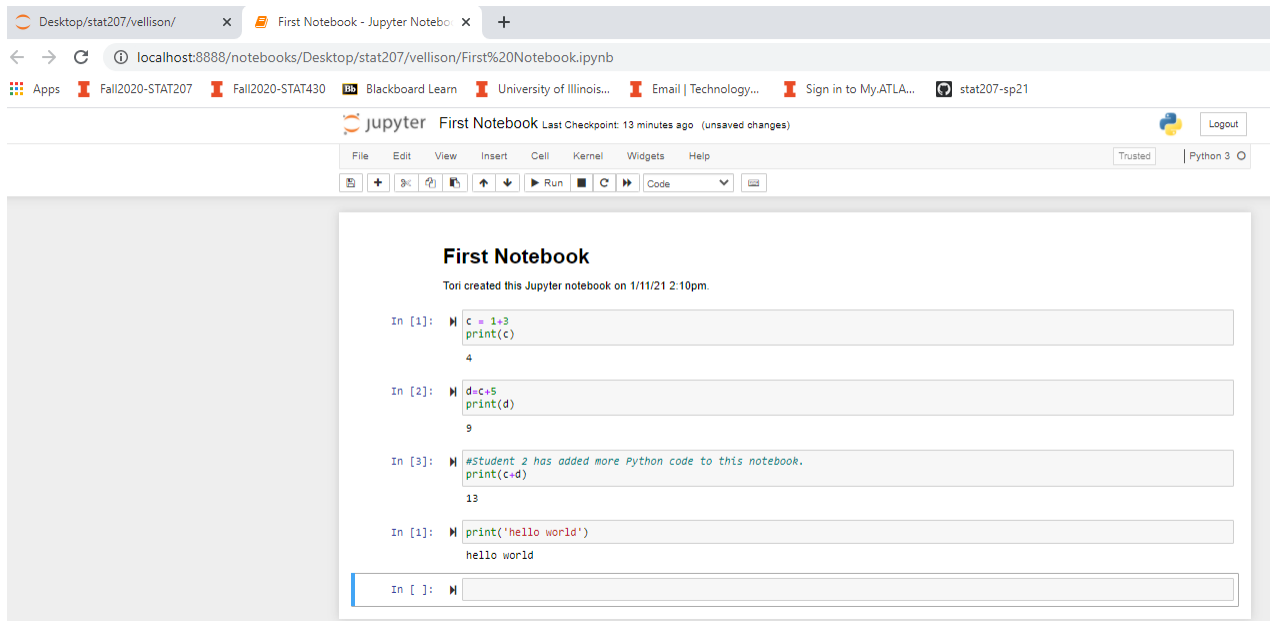
1. First, check out the instructions in your first lab <http://courses.las.illinois.edu/spring2022/stat207/labs/01-intro.html> for how to download **minicondas** software which contains the **Jupyter notebook** application.
2. Start your **Jupyter notebook** application



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.

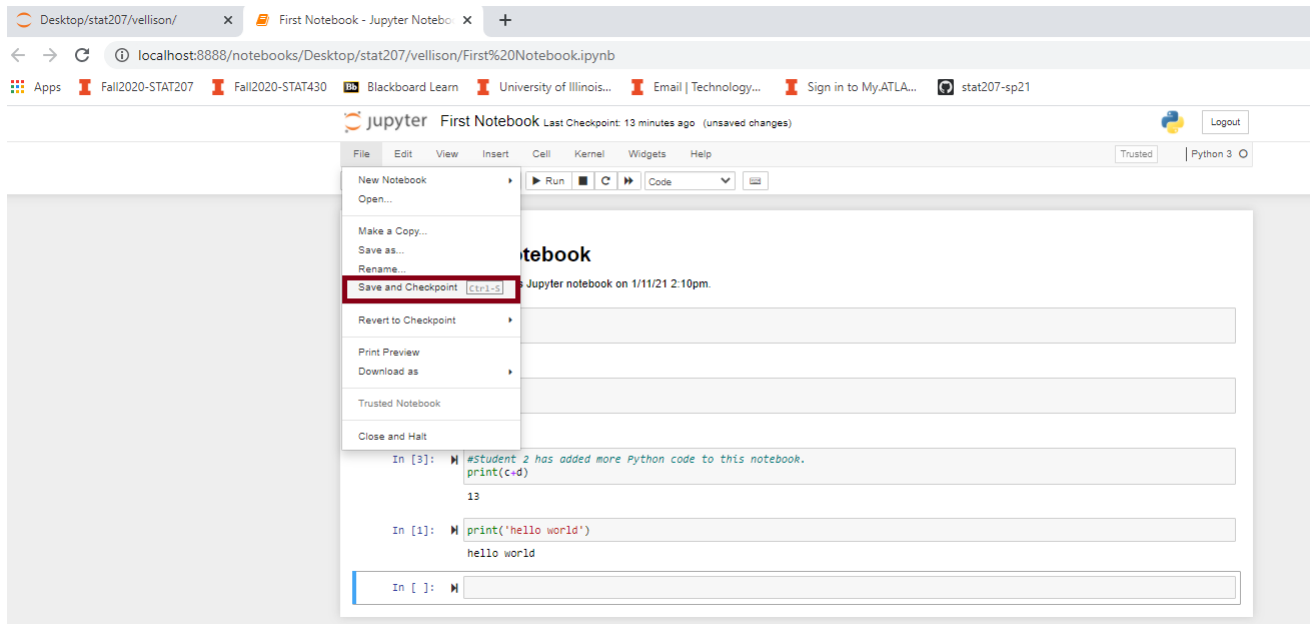


The screenshot shows a web browser window with the URL `localhost:8888/notebooks/Desktop/stat207/vellison/First%20Notebook.ipynb`. The Jupyter interface displays the title "First Notebook" and a message: "Tori created this Jupyter notebook on 1/11/21 2:10pm." Below this, there are four code cells:

- In [1]: `c = 1+3`
`print(c)`
4
- In [2]: `d=c+5`
`print(d)`
9
- In [3]: `#Student 2 has added more Python code to this notebook.`
`print(c+d)`
13
- In [1]: `print('hello world')`
hello world

The bottom cell is currently empty and labeled "In []:".

5. Save any changes made to the notebook.



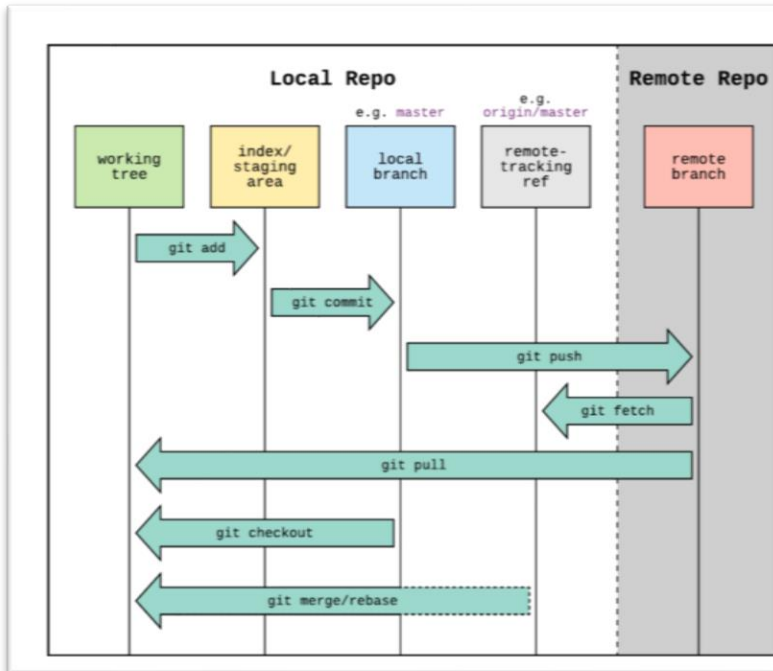
The screenshot shows the same Jupyter Notebook interface as above, but with the "File" menu open. The "Save and Checkpoint" option is highlighted with a red box. The menu items are:

- 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

The notebook content is partially visible behind the menu, showing the same code cells as in the previous screenshot.

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/spring2022/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.

The screenshot shows the GitHub repository page for 'stat207-sp21/vellison'. The main heading is 'Learn Git and GitHub without any code!' with a subtext 'Using the Hello World guide, you'll start a branch, write comments, and open a pull request.' and a 'Read the guide' button. Below this, the repository name 'stat207-sp21/vellison' is shown with a 'Private' label. The file browser shows a list of files: 'vellison any message you want about what changes you made' (185x142, 4 minutes ago, 10 commits), '.ipynb_checkpoints', 'First Notebook.ipynb' (any message you want about what changes you made, 4 minutes ago), and 'README.md' (Update README.md, yesterday). The 'README.md' file is selected and its content is displayed. The content includes a section titled '_release' and instructions for weekly lab assignments, including a link to a course page and a new branch named '_release'.

The screenshot shows the GitHub repository page for 'stat207-sp21/vellison' with the file 'First Notebook.ipynb' selected. The main heading is 'Learn Git and GitHub without any code!' with a subtext 'Using the Hello World guide, you'll start a branch, write comments, and open a pull request.' and a 'Read the guide' button. Below this, the repository name '1 / vellison' is shown with a 'Private' label. The file browser shows the file 'vellison / First Notebook.ipynb' (Latest commit 185x142 38 minutes ago, History). The file content is displayed, showing a Jupyter notebook with the following code cells:

```
In [1]: c = 1+3
print(c)
4

In [2]: d=c+5
print(d)
9

In [3]: #Student 2 has added more Python code to this notebook.
print(c+d)
13

In [1]: print("hello world")
hello world

In [ ]:
```