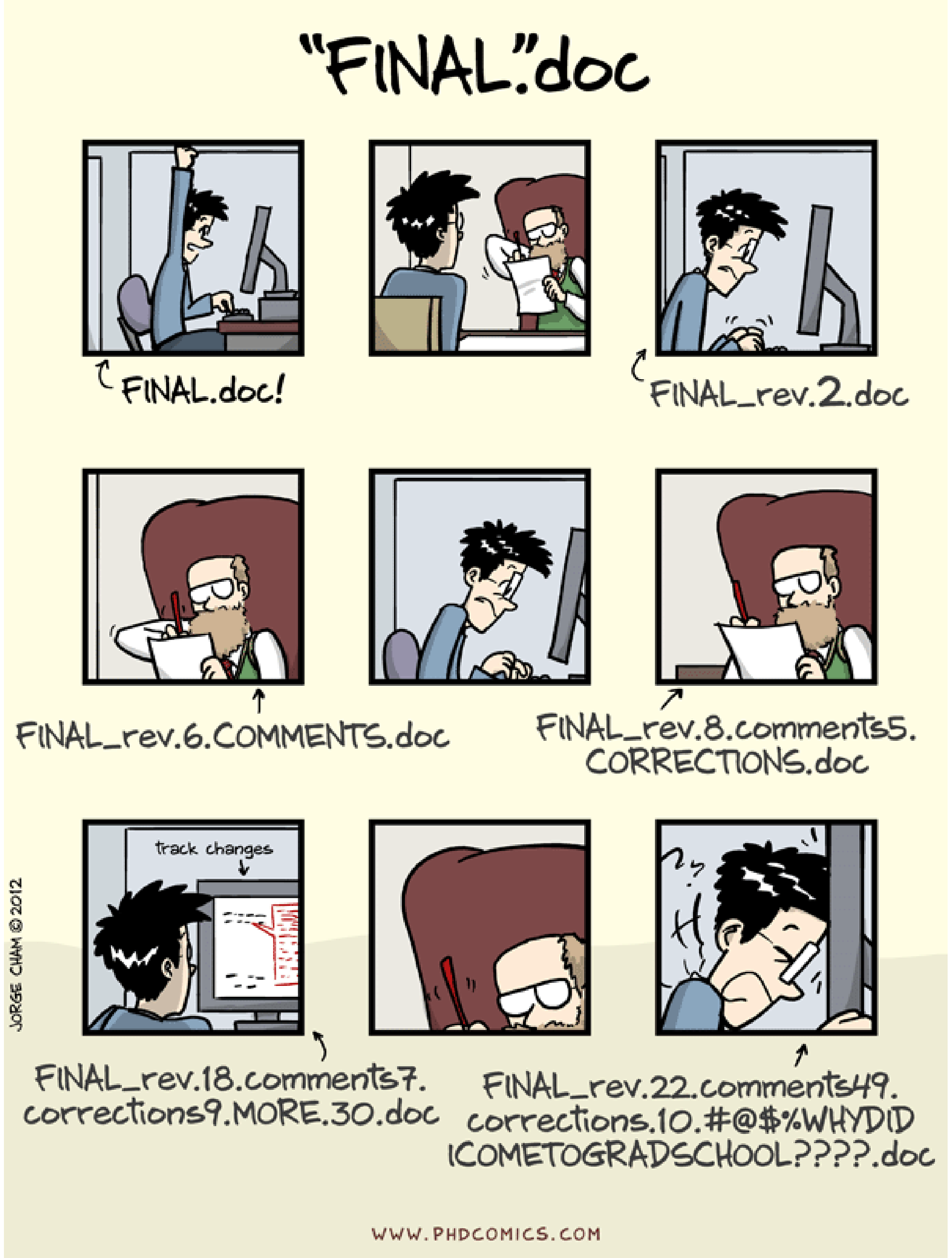
Version control and GitHub

Why use version control?


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDIDICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Version control and Git

- Main features of a version control system:
    1. Save each new set of changes sequentially
    2. Keep track of different versions of a document/project
    3. Merge changes from multiple versions

- **Git** is a specific version control **system**
    - Think “track changes” in Word + Dropbox, but much more general and powerful

- A whole new system to learn. Is it worth the effort?
    - Maybe not when working alone
    - But critical to avoid disaster when collaborating
- The gold standard in the private sector – used EVERYWHERE

GitHub

- GitHub is a **specific website** that uses Git to host projects in the cloud
- We will use GitHub at a few points in this course
  - Lecture slides
  - Assignment 2
  - Term project
- Why?
  - To start building habits of using version control
  - To get you used to the basic terminology and actions of Git and GitHub
- “Real” developers & data scientists use Git at the command line
  - I’m not going to require you to do that now
  - But I encourage you to learn it on your own

Getting set up with GitHub

1. Create a GitHub account (unless you already have one)
2. Download GitHub Desktop
3. Connect GitHub Desktop to your GitHub account

1. Create a GitHub account

- Go here and fill out the forms: https://github.com/

- No need to apply for the GitHub Student benefits (though you can if you want to)

2. Download GitHub Desktop

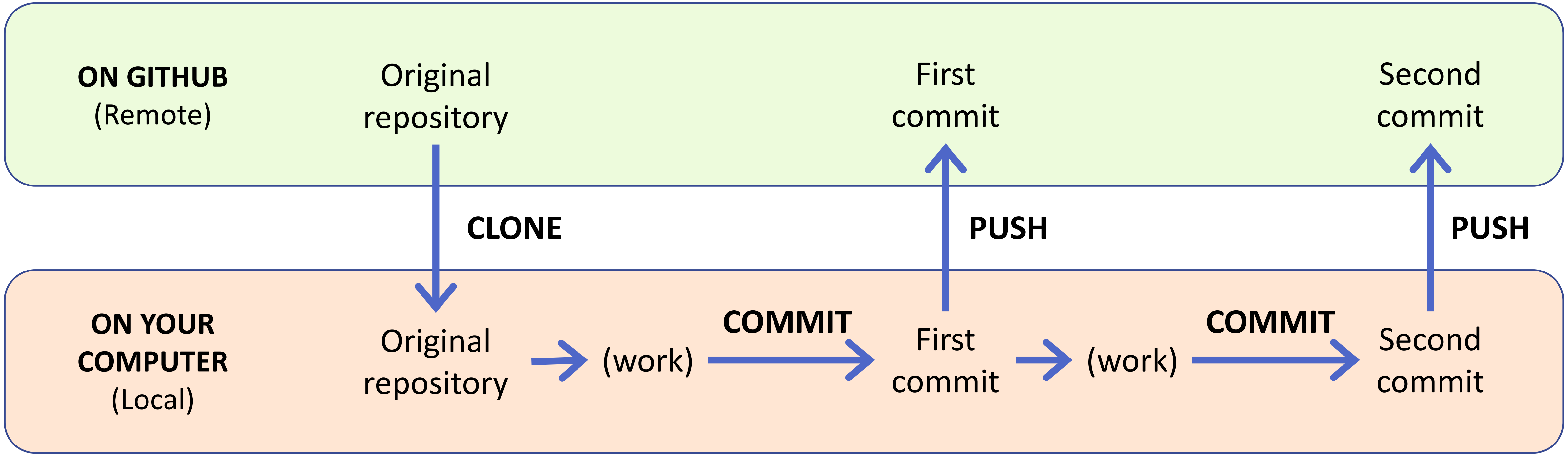
- Go here: https://desktop.github.com/

- GitHub Desktop is a standalone app for using Git and GitHub through a graphical user interface (GUI).
- Recommend but not strictly required
    - You can submit assignments directly through the GitHub website, but it will end up being harder in many ways
    - You can use Git at the command line (shell) if you already know it or want to learn

3. Connect GitHub Desktop to your GitHub account

- Open GitHub Desktop and go to File -> Options
- If you need help, try this: https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/installing-and-authenticating-to-github-desktop/setting-up-github-desktop

Basic workflow (only 1 contributor)

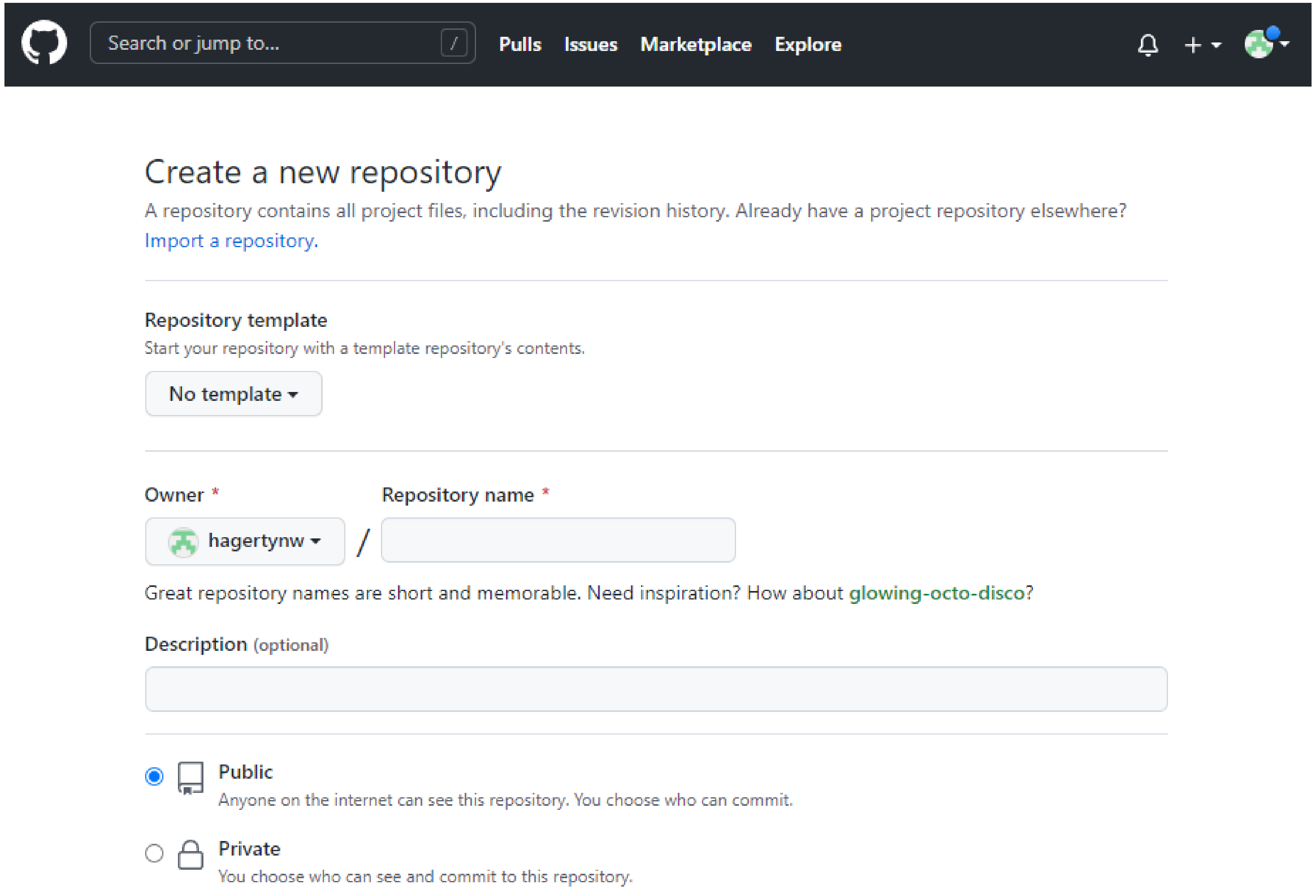


Workflow for your project

1. On GitHub.com, create a new repository
2. **Clone** this repository to your local machine
3. Do some work (edit the repository)
4. **Commit** changes (i.e., save a draft)
5. **Push** your commit to GitHub (back it up to the cloud)

1. Create a new repository

A repository (**repo**) is the full record of a project folder and all its changes ever.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template

Owner * Repository name *

hagertynw /

Great repository names are short and memorable. Need inspiration? How about glowing-octo-disco?

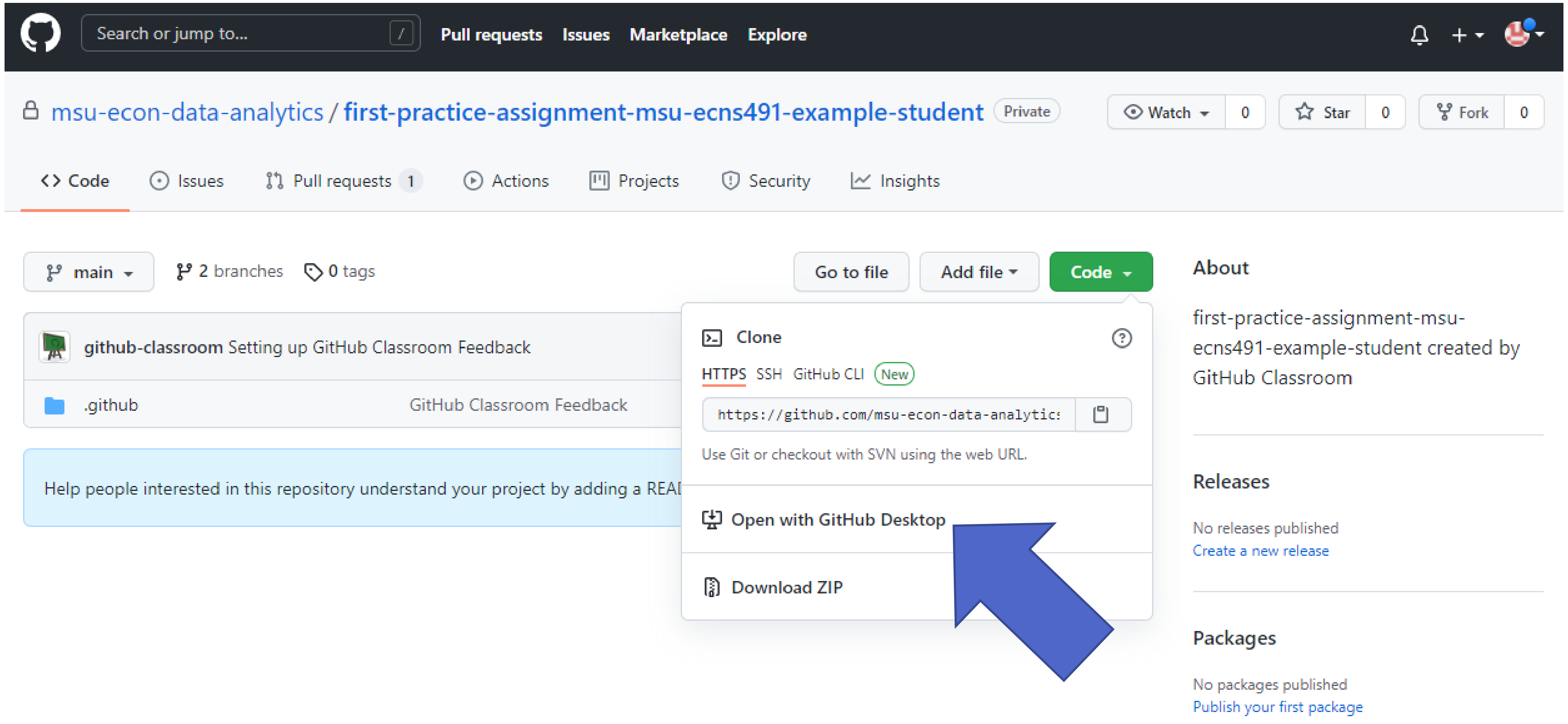
Description (optional)

[✓ Public]
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

2. **Clone** the repo to your local machine

- Clone downloads a full copy of the repo from GitHub to file storage on your computer



Search or jump to... Pull requests Issues Marketplace Explore

msu-econ-data-analytics / first-practice-assignment-msu-ecns491-example-student Private

Watch 0 Star 0 Fork 0

Code Issues Pull requests 1 Actions Projects Security Insights

main 2 branches 0 tags Go to file Add file Code

github-classroom Setting up GitHub Classroom Feedback

.github GitHub Classroom Feedback

Help people interested in this repository understand your project by adding a REAI

Clone

HTTPS SSH GitHub CLI New

```
https://github.com/msu-econ-data-analytic:
```

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

first-practice-assignment-msu-ecns491-example-student created by GitHub Classroom

Releases

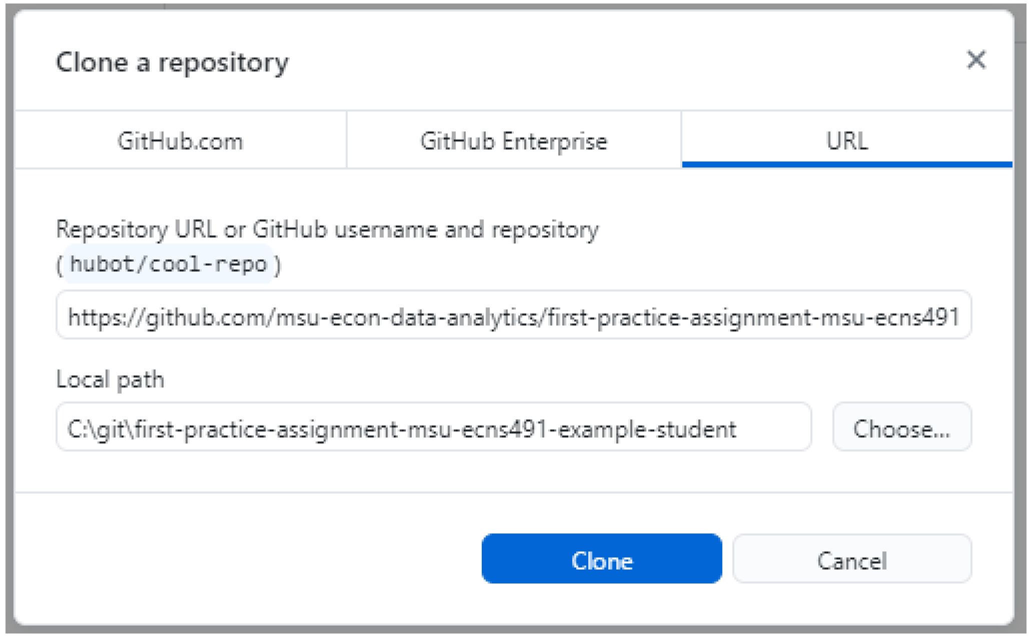
No releases published
Create a new release

Packages

No packages published
Publish your first package

2. **Clone** the repo to your local machine

- GitHub Desktop should now come up
- Choose where you want to store the repo on your computer (the default location is probably fine)


Clone a repository

Repository URL or GitHub username and repository
(`hubot/cool-repo`)

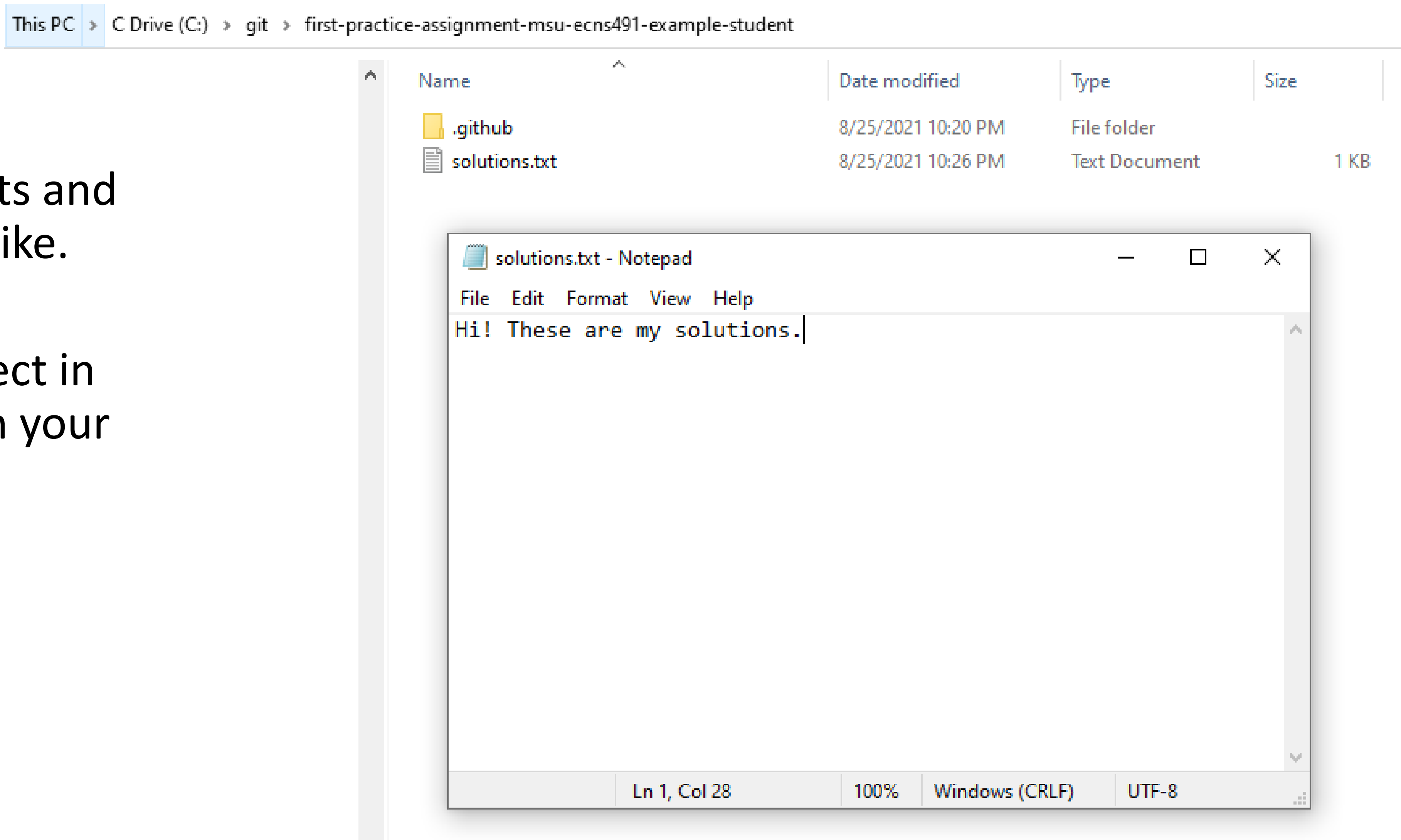
https://github.com/msu-econ-data-analytics/first-practice-assignment-msu-ecns491

Local path

C:\git\first-practice-assignment-msu-ecns491-example-student

3. Do the assignment (edit the repo)

- Create or edit scripts and documents as you like.
- Save all documents related to this project in the repo's folder on your computer.


This PC › C Drive (C:) › git › first-practice-assignment-msu-ecns491-example-student
solutions.txt - Notepad
File Edit Format View Help

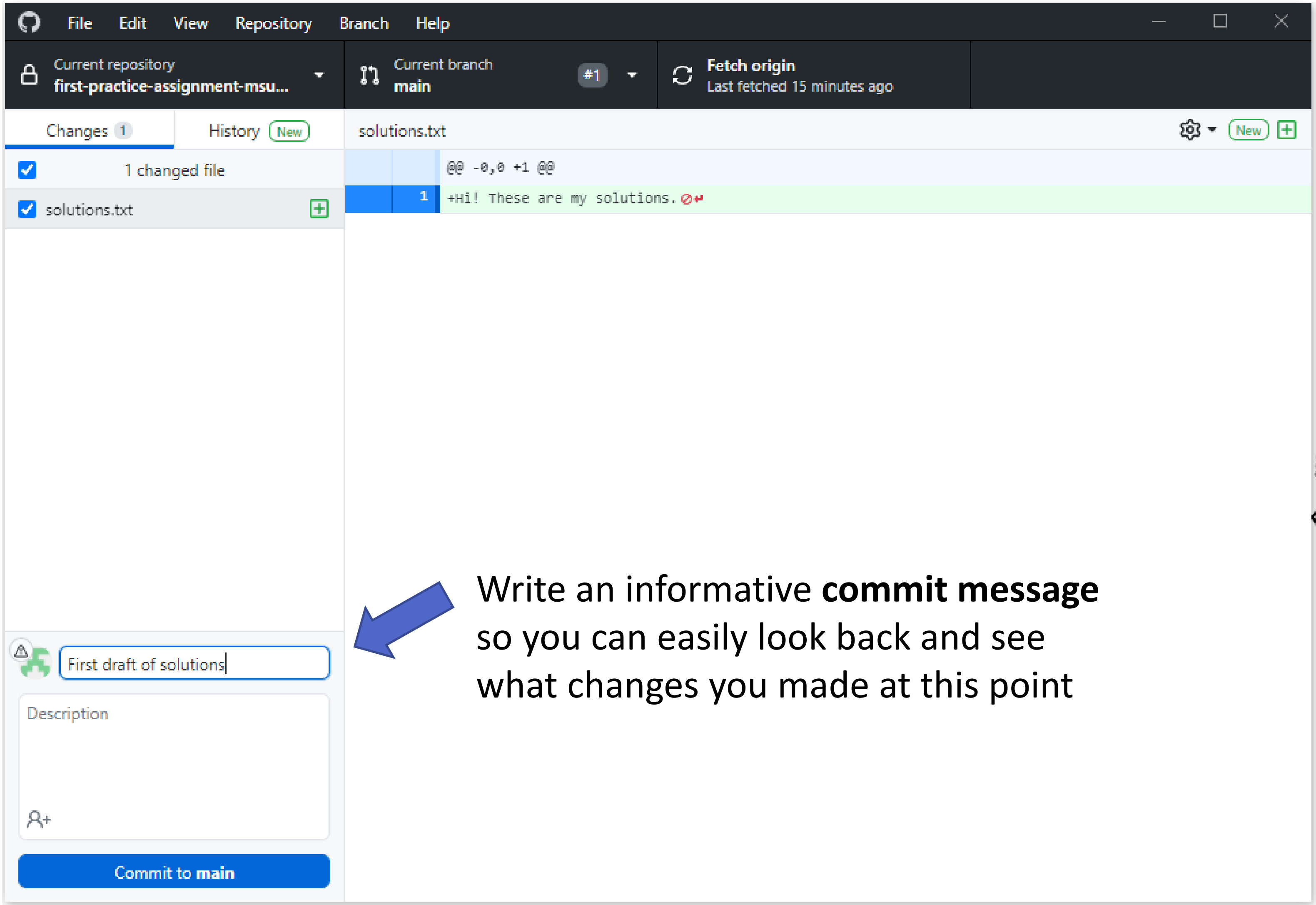
Name	Date modified	Type	Size
.github	8/25/2021 10:20 PM	File folder	
solutions.txt	8/25/2021 10:26 PM	Text Document	1 KB

```
Hi! These are my solutions.
```

Ln 1, Col 28 | 100% | Windows (CRLF) | UTF-8

4. **Commit** your changes

- Commit is like Save, but for your whole project
- It records a snapshot of your whole directory at this point
- Unlike Save (but like version history in Google Docs), you can go back to a particular commit later



```
@@ -0,0 +1 @@
1 +Hi! These are my solutions.
```

First draft of solutions

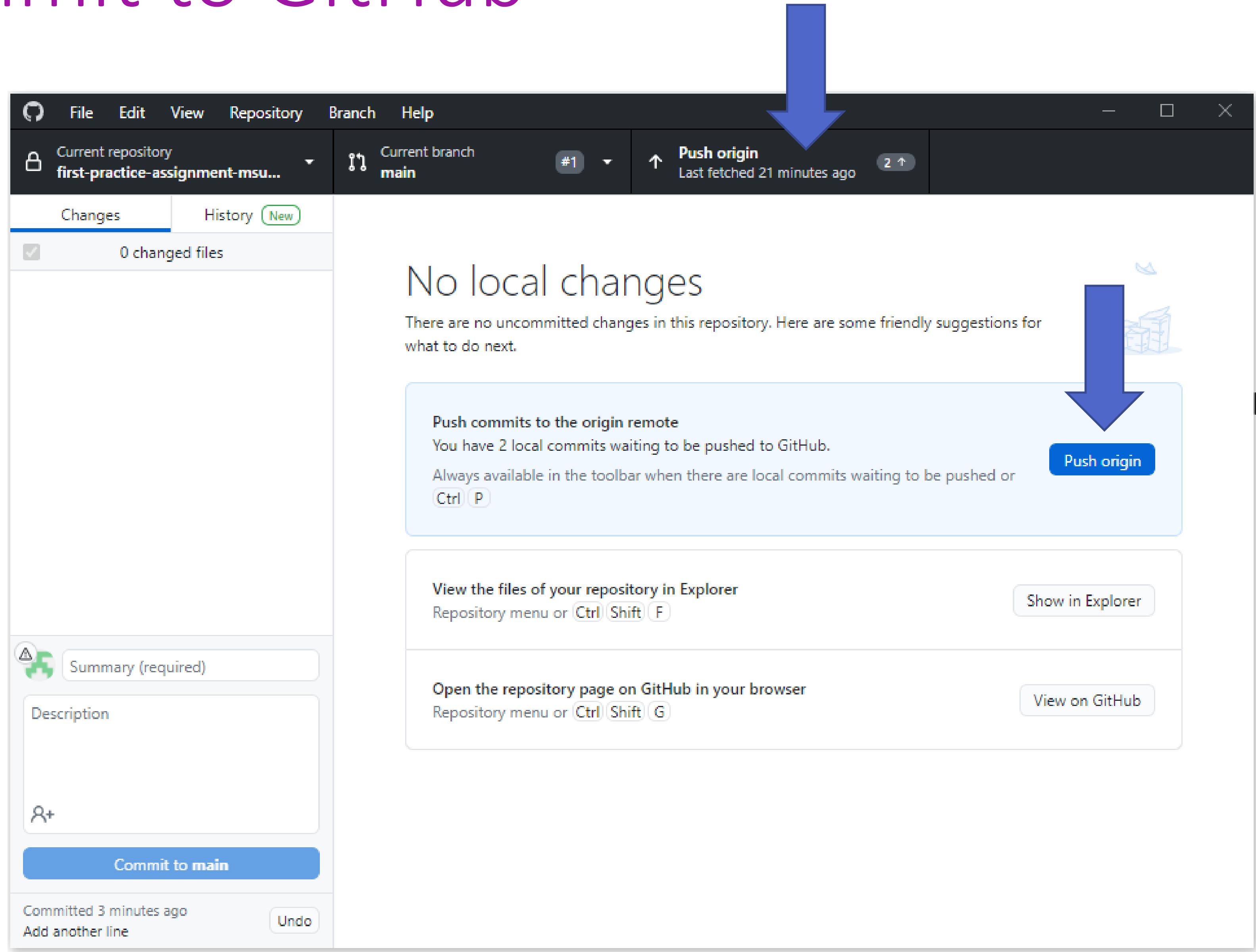
Write an informative **commit message** so you can easily look back and see what changes you made at this point

4. **Commit** your changes

- Commit early and often!
  - Every time you make a major change, or take a break from working
  - If you make a big mistake, you can use GitHub Desktop to roll back to an earlier commit

5. **Push** your commit to GitHub

- Commit is only local (your changes aren't on GitHub yet)
- Now we need to **push** the commit(s) to the remote GitHub repository
- Push uploads your changes to the cloud (GitHub)



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 2 local commits waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or Ctrl P

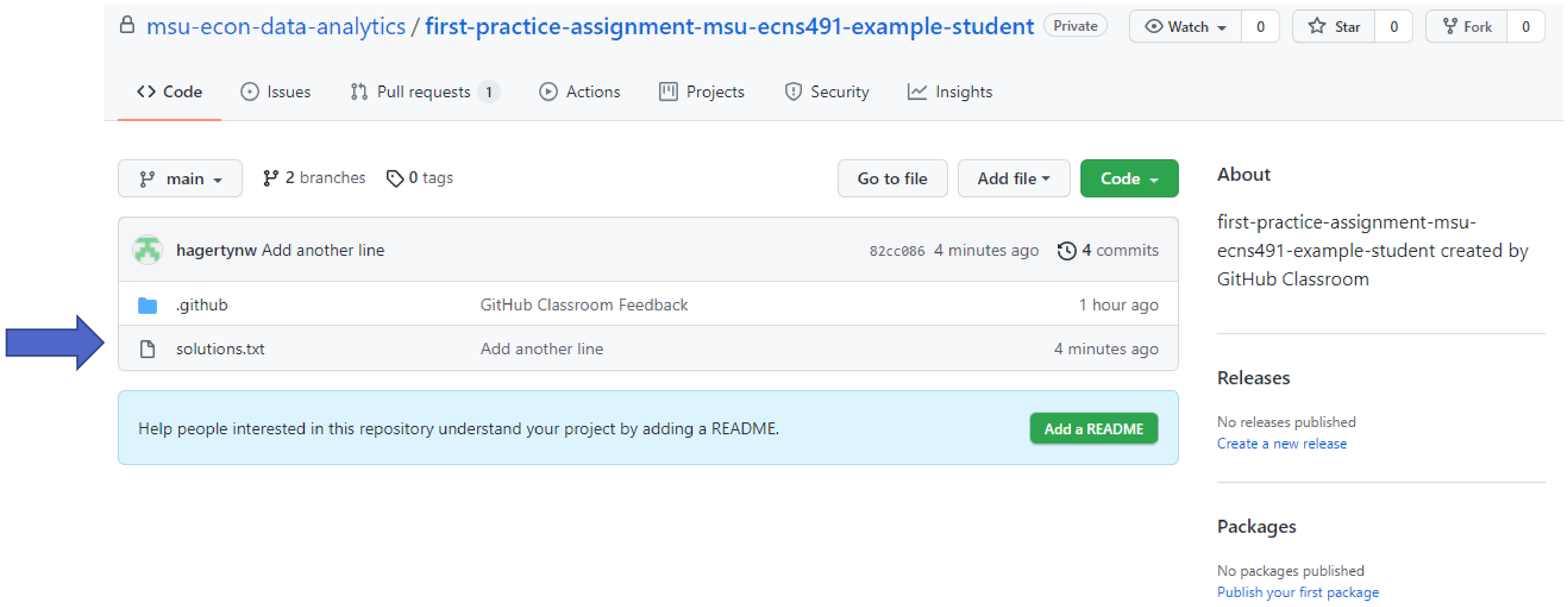
View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

Committed 3 minutes ago
Add another line

5. **Push** your commit to GitHub

- Now, back on GitHub, you can see the new files you added



msu-econ-data-analytics / first-practice-assignment-msu-ecns491-example-student Private

hagertynw Add another line 82cc086 4 minutes ago 4 commits

.github	GitHub Classroom Feedback	1 hour ago
solutions.txt	Add another line	4 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

first-practice-assignment-msu-ecns491-example-student created by GitHub Classroom

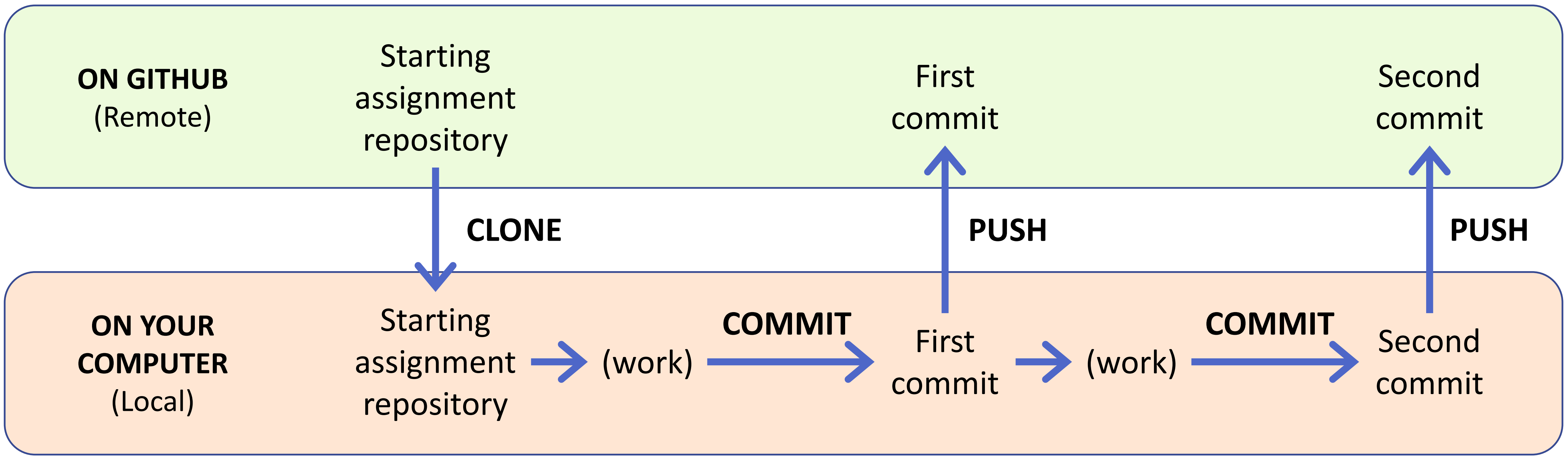
Releases

No releases published
Create a new release

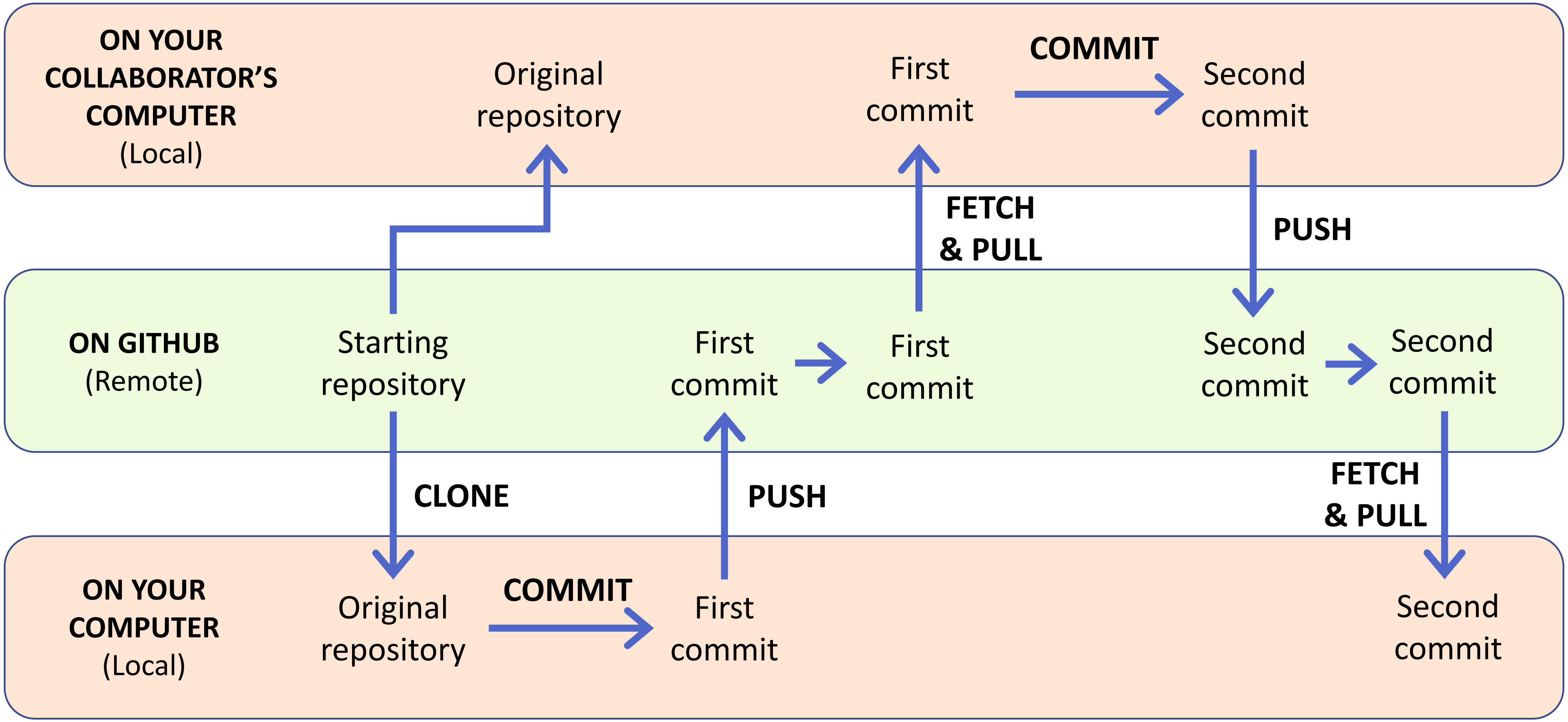
Packages

No packages published
Publish your first package

Workflow for each assignment

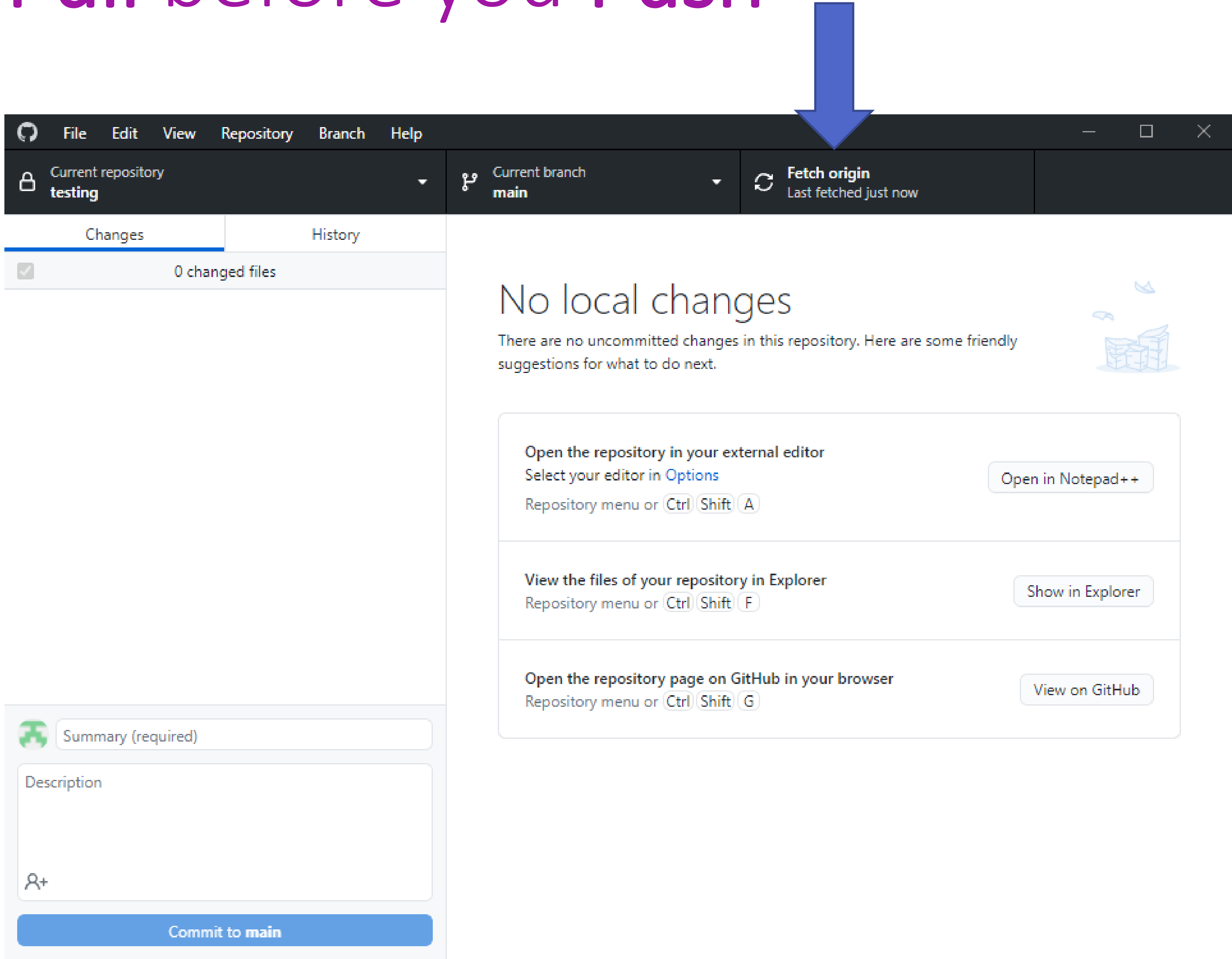


Example collaborative workflow



Always **Fetch** and **Pull** before you **Push**

- Your collaborator might have made changes since you last worked on it
- **Fetch** to check for changes



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

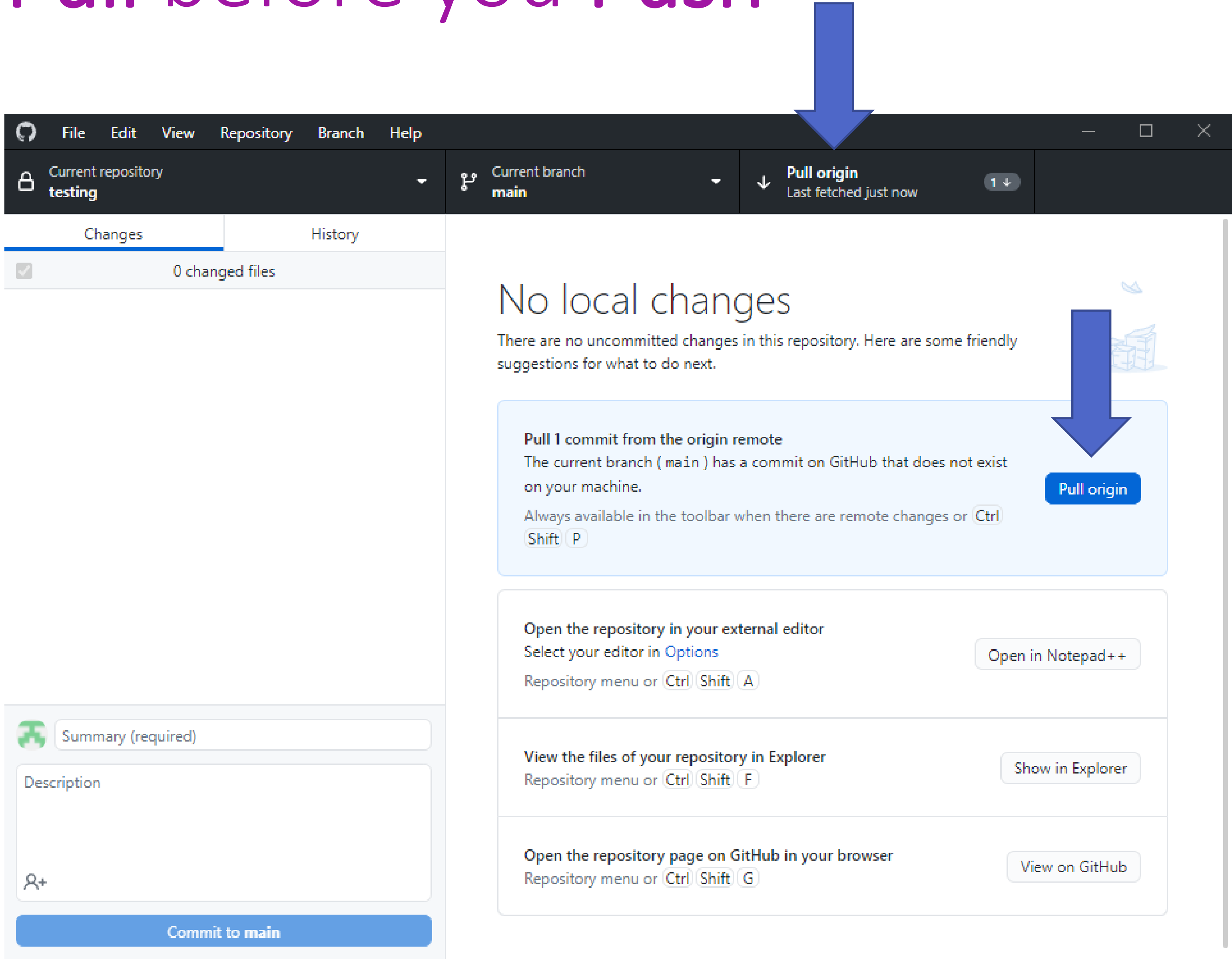
Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A

View the files of your repository in Explorer
Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

Always **Fetch** and **Pull** before you **Push**

- Your collaborator might have made changes since you last worked on it
- **Fetch** to check for changes
- **Push** to merge their changes with yours
- Resolve any merge conflicts
- Now you can **push**!



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull 1 commit from the origin remote

The current branch (`main`) has a commit on GitHub that does not exist on your machine.

Always available in the toolbar when there are remote changes or Ctrl Shift P

Open the repository in your external editor

Select your editor in Options

Repository menu or Ctrl Shift A

View the files of your repository in Explorer

Repository menu or Ctrl Shift F

Open the repository page on GitHub in your browser

Repository menu or Ctrl Shift G

Many more features & workflow options

(All optional, but very useful for collaborating)

- Forking and pull requests: https://guides.github.com/activities/forking/
- Branches and merges: https://guides.github.com/activities/hello-world/
- For much more, see the other “Git and GitHub” resources on the course resource list: https://github.com/msu-econ-data-analytics/course-materials#git-and-github