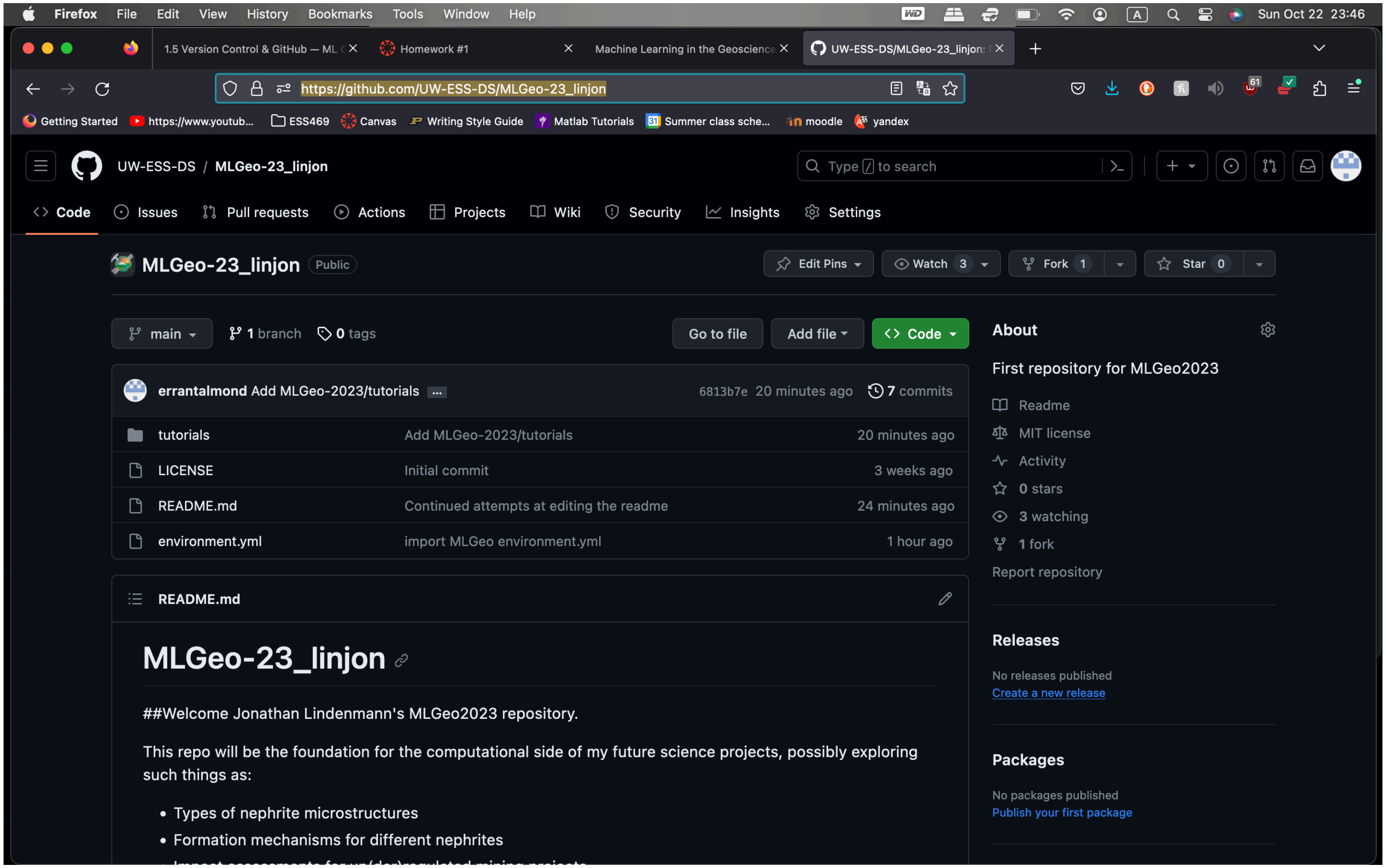
Making your own contributions to my repository

To work on my anything stored in my repository, you can get all the files to your local system with a pull request. Here are the steps:

1. Access my repository on GitHub and click on ‘fork’ near the top right of the page:



First repository for MLGeo2023

No releases published
Create a new release

No packages published
Publish your first package

errantalmond Add MLGeo-2023/tutorials ... 6813b7e 20 minutes ago 7 commits

tutorials	Add MLGeo-2023/tutorials	20 minutes ago
LICENSE	Initial commit	3 weeks ago
README.md	Continued attempts at editing the readme	24 minutes ago
environment.yml	import MLGeo environment.yml	1 hour ago

README.md

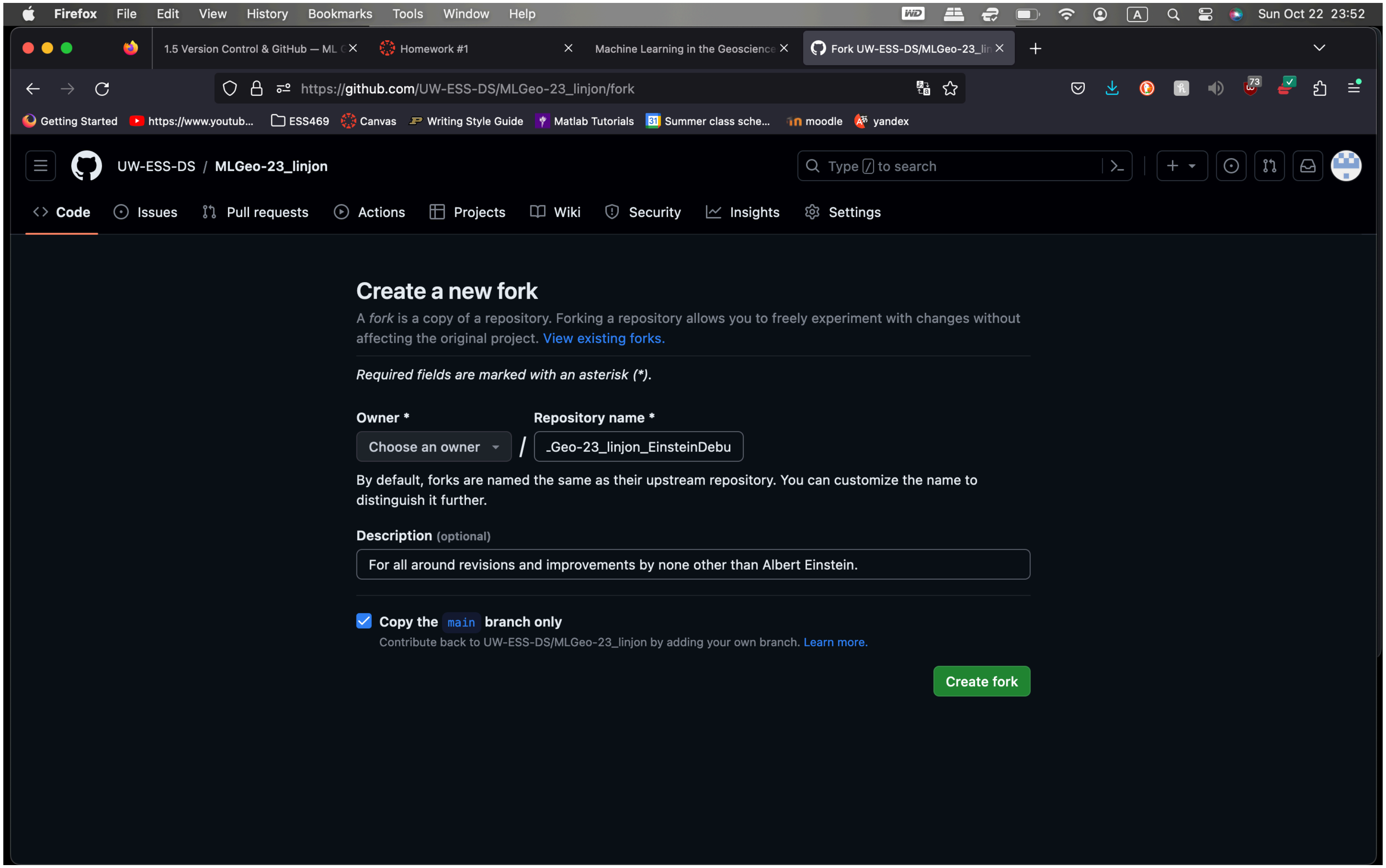
MLGeo-23_linjon

##Welcome Jonathan Lindenmann's MLGeo2023 repository.

This repo will be the foundation for the computational side of my future science projects, possibly exploring such things as:

- Types of nephrite microstructures
- Formation mechanisms for different nephrites
- Impact assessments for un(der)regulated mining projects

2. Add some information on the following page to identify the purpose of the fork and who is working on it. You'll also want to choose yourself as the owner:


Firefox File Edit View History Bookmarks Tools Window Help
Sun Oct 22 23:52
1.5 Version Control & GitHub — ML
Homework #1
Machine Learning in the Geoscience
Fork UW-ESS-DS/MLGeo-23_lin
https://github.com/UW-ESS-DS/MLGeo-23_linjon/fork
Getting Started
https://www.youtub...
ESS469
Canvas
Writing Style Guide
Matlab Tutorials
Summer class sche...
moodle
yandex
UW-ESS-DS / MLGeo-23_linjon
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner * | **Repository name** *

Choose an owner / _Geo-23_linjon_EinsteinDebu

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

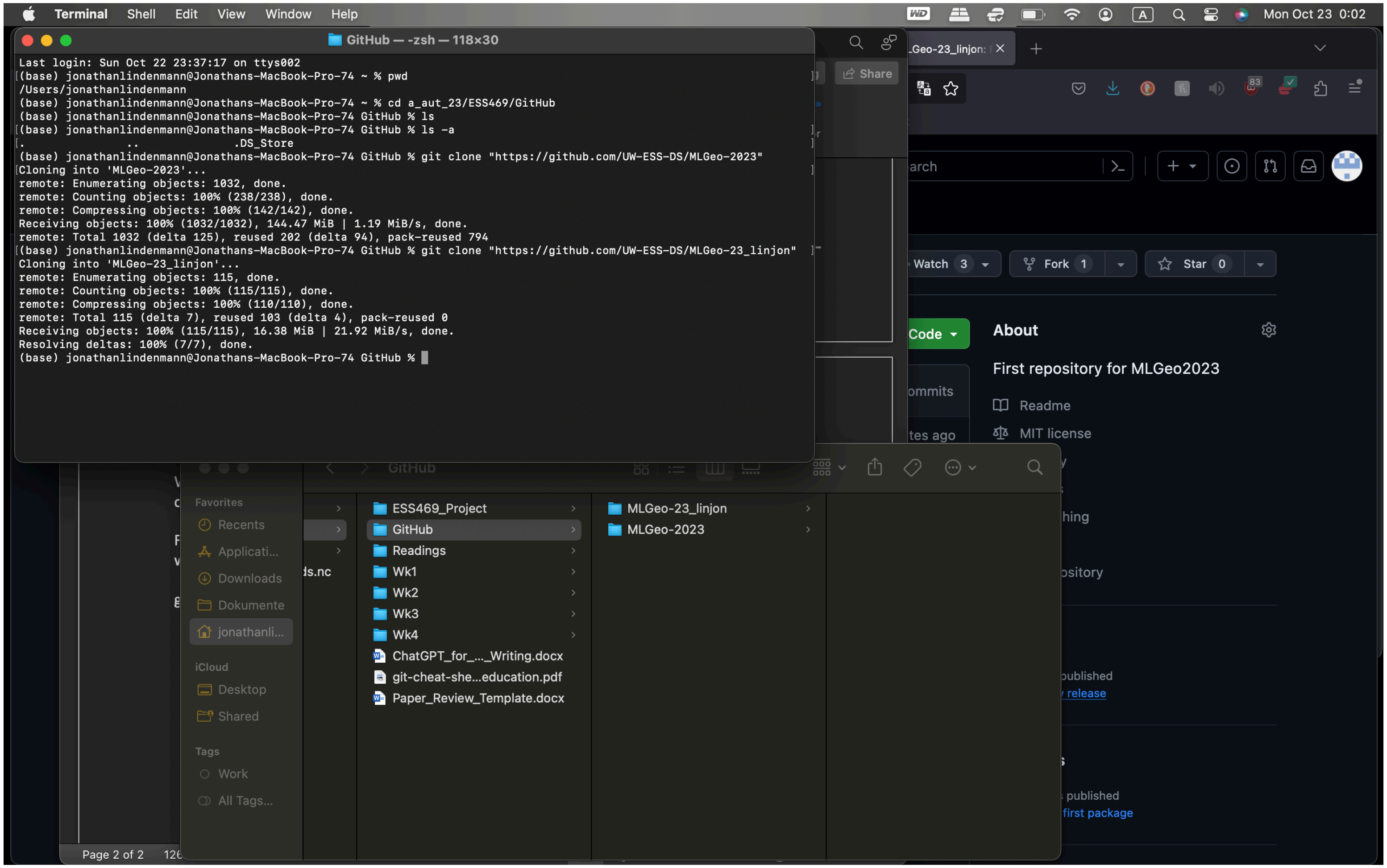
For all around revisions and improvements by none other than Albert Einstein.

☑ **Copy the `main` branch only**

Contribute back to UW-ESS-DS/MLGeo-23_linjon by adding your own branch. Learn more.

3. With the new fork created, you can now use GitHub Desktop or a command line interface to clone your new fork onto your local system and begin working on it.
   For Mac OS, simply enter the following into the terminal after navigating to the directory you want to clone to:

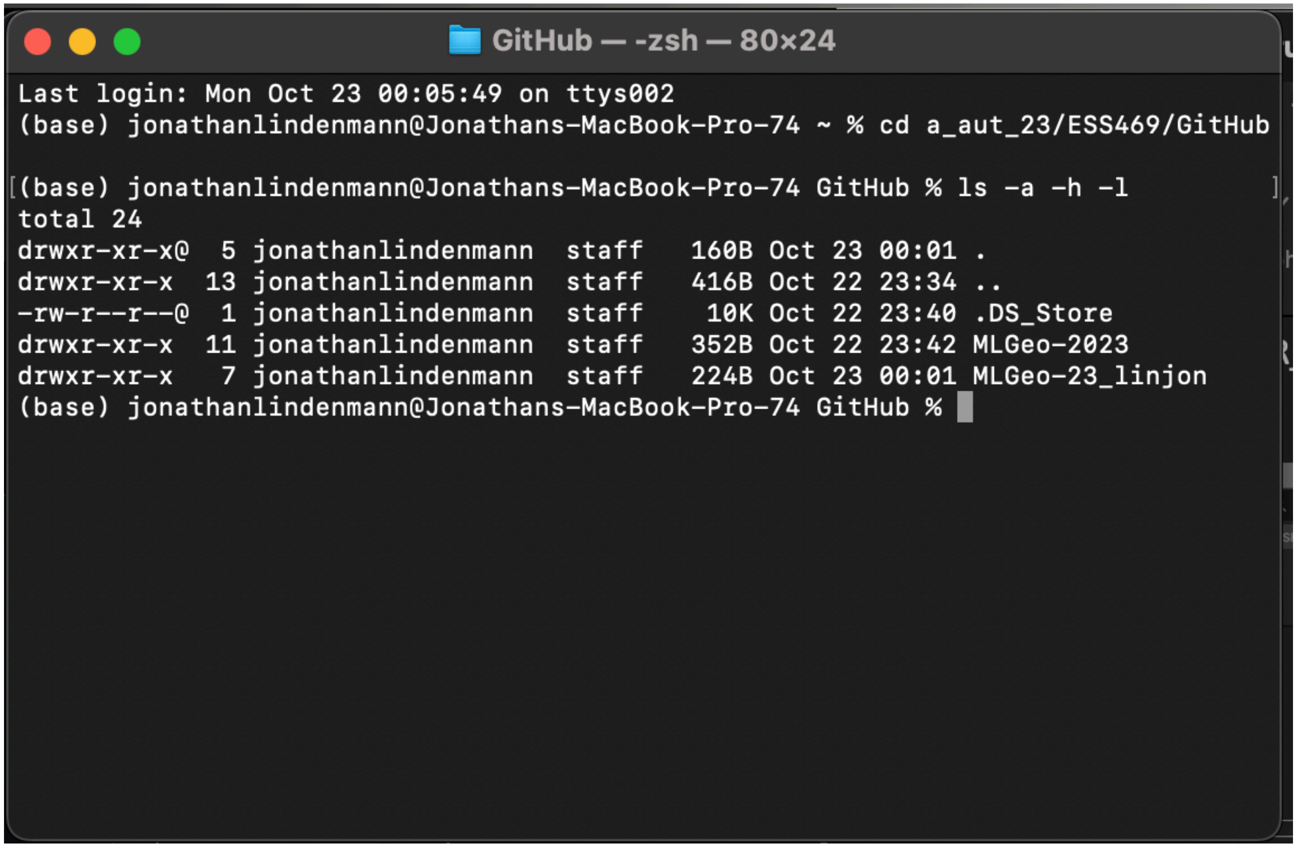
```
git clone “https://github.com/[YOUR_GITHUB_PAGE]/[YOUR_NEW_NAME_FOR_MY_REPO]”
```



```
Last login: Sun Oct 22 23:37:17 on ttys002
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % pwd
/Users/jonathanlindenmann
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % cd a_aut_23/ESS469/GitHub
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub % ls
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub % ls -a
.                ..                .DS_Store
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub % git clone "https://github.com/UW-ESS-DS/MLGeo-2023"
Cloning into 'MLGeo-2023'...
remote: Enumerating objects: 1032, done.
remote: Counting objects: 100% (238/238), done.
remote: Compressing objects: 100% (142/142), done.
Receiving objects: 100% (1032/1032), 144.47 MiB | 1.19 MiB/s, done.
remote: Total 1032 (delta 125), reused 202 (delta 94), pack-reused 794
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub % git clone "https://github.com/UW-ESS-DS/MLGeo-23_linjon"
Cloning into 'MLGeo-23_linjon'...
remote: Enumerating objects: 115, done.
remote: Counting objects: 100% (115/115), done.
remote: Compressing objects: 100% (110/110), done.
remote: Total 115 (delta 7), reused 103 (delta 4), pack-reused 0
Receiving objects: 100% (115/115), 16.38 MiB | 21.92 MiB/s, done.
Resolving deltas: 100% (7/7), done.
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub %
```

In the image above you can see the inputs and outputs in terminal which cloned the MLGeo-23_linjon MAIN repo and the MLGeo-2023 class repo from the UW-ESS-DS GitHub organization into the …/ESS469/GitHub folder shown in the Finder.

You can see below that the GitHub folder now contains the directories (folders) of the two repositories, as well as the .git repos, represented by the '.' and '..':



```
Last login: Mon Oct 23 00:05:49 on ttys002
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % cd a_aut_23/ESS469/GitHub

(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub % ls -a -h -l
total 24
drwxr-xr-x@  5 jonathanlindenmann  staff   160B Oct 23 00:01 .
drwxr-xr-x  13 jonathanlindenmann  staff   416B Oct 22 23:34 ..
-rw-r--r--@  1 jonathanlindenmann  staff    10K Oct 22 23:40 .DS_Store
drwxr-xr-x  11 jonathanlindenmann  staff   352B Oct 22 23:42 MLGeo-2023
drwxr-xr-x   7 jonathanlindenmann  staff   224B Oct 23 00:01 MLGeo-23_linjon
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 GitHub %
```

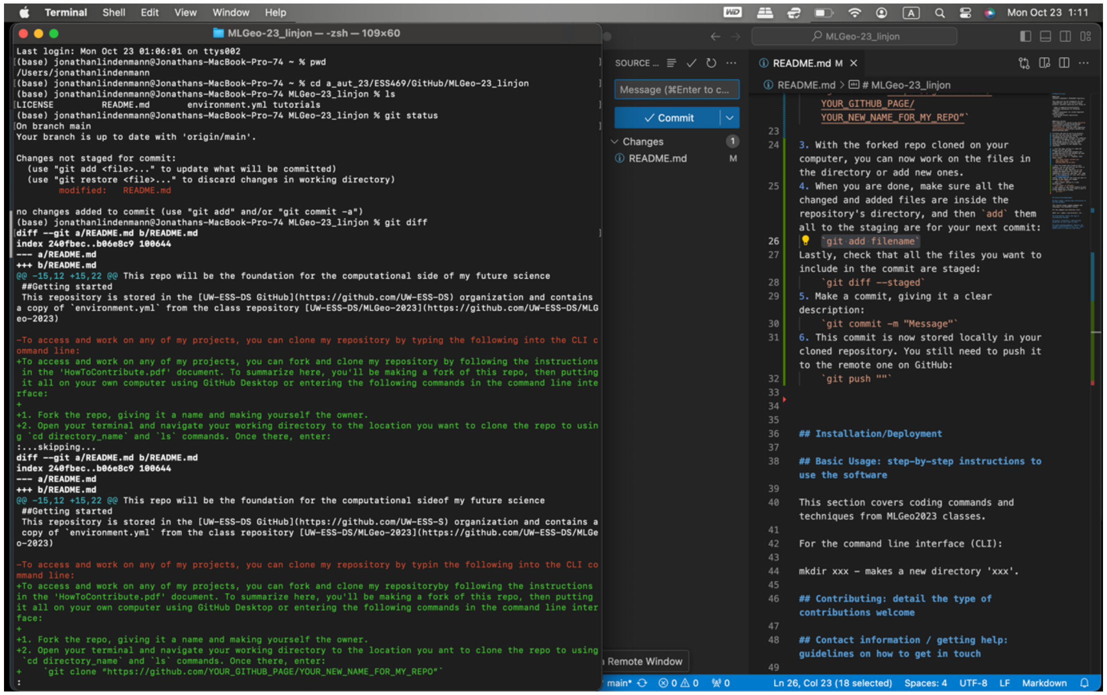
4. Now, you can work on the files in the repository or add new ones. When you are finished, stage and commit them to your clone of your fork, and finally, you can submit a pull request to have your fork merged back into the main repository.
   For example, if I want to work on my readme.md for the main repository, I could work on it in my cloned for on my computer, and when I am finished, I can add it to the staging area for changes to commit to the cloned repository, before then pushing that commit to the remote repository on GitHub. Let's say I've worked on my readme.md a bit in VSCode and saved the file. I can now stage the file:

   Check status and see what the changes are:

   ```
   git status
   ```

   ```
   git diff
   ```

   The image below shows that these commands return the readme.md file as modified and show the changes I made.



```
Last login: Mon Oct 23 01:06:01 on ttys002
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % pwd
/Users/jonathanlindenmann
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % cd a_aut_23/ESS469/GitHub/MLGeo-23_linjon
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % ls
LICENSE         README.md       environment.yml tutorials
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git diff
diff --git a/README.md b/README.md
index 240fbec..b06e8c9 100644
--- a/README.md
+++ b/README.md
@@ -15,12 +15,22 @@ This repo will be the foundation for the computational side of my future science
 ##Getting started
 This repository is stored in the [UW-ESS-DS GitHub](https://github.com/UW-ESS-DS) organization and contains
a copy of `environment.yml` from the class repository [UW-ESS-DS/MLGeo-2023](https://github.com/UW-ESS-DS/MLG
eo-2023)

-To access and work on any of my projects, you can clone my repository by typing the following into the CLI c
ommand line:
+To access and work on any of my projects, you can fork and clone my repository by following the instructions
 in the 'HowToContribute.pdf' document. To summarize here, you'll be making a fork of this repo, then putting
 it all on your own computer using GitHub Desktop or entering the following commands in the command line inte
rface:
+
+1. Fork the repo, giving it a name and making yourself the owner.
+2. Open your terminal and navigate your working directory to the location you want to clone the repo to usin
g `cd directory_name` and `ls` commands. Once there, enter:
:...skipping...
diff --git a/README.md b/README.md
index 240fbec..b06e8c9 100644
--- a/README.md
+++ b/README.md
@@ -15,12 +15,22 @@ This repo will be the foundation for the computational sideof my future science
 ##Getting started
 This repository is stored in the [UW-ESS-DS GitHub](https://github.com/UW-ESS-S) organization and contains a
 copy of `environment.yml` from the class repository [UW-ESS-DS/MLGeo-2023](https://github.com/UW-ESS-DS/MLGe
o-2023)

-To access and work on any of my projects, you can clone my repository by typin the following into the CLI co
mmand line:
+To access and work on any of my projects, you can fork and clone my repositoryby following the instructions
in the 'HowToContribute.pdf' document. To summarize here, you'll be making a fork of this repo, then putting
it all on your own computer using GitHub Desktop or entering the following commands in the command line inter
face:
+
+1. Fork the repo, giving it a name and making yourself the owner.
+2. Open your terminal and navigate your working directory to the location you ant to clone the repo to using
 `cd directory_name` and `ls` commands. Once there, enter:
+    `git clone "https://github.com/YOUR_GITHUB_PAGE/YOUR_NEW_NAME_FOR_MY_REPO"`
:
```

```
    YOUR_GITHUB_PAGE/
    YOUR_NEW_NAME_FOR_MY_REPO"`

3. With the forked repo cloned on your computer, you can now work on the files in the directory or add new ones.
4. When you are done, make sure all the changed and added files are inside the repository's directory, and then `add` them all to the staging are for your next commit:
    `git add filename`
Lastly, check that all the files you want to include in the commit are staged:
    `git diff --staged`
5. Make a commit, giving it a clear description:
    `git commit -m "Message"`
6. This commit is now stored locally in your cloned repository. You still need to push it to the remote one on GitHub:
    `git push ""`



## Installation/Deployment

## Basic Usage: step-by-step instructions to use the software

This section covers coding commands and techniques from MLGeo2023 classes.

For the command line interface (CLI):

mkdir xxx - makes a new directory 'xxx'.

## Contributing: detail the type of contributions welcome

## Contact information / getting help: guidelines on how to get in touch

```

Next, we want to stage the changes, and to see what is staged. The only file with changes is the readme file, so:

```
git add README.md
```

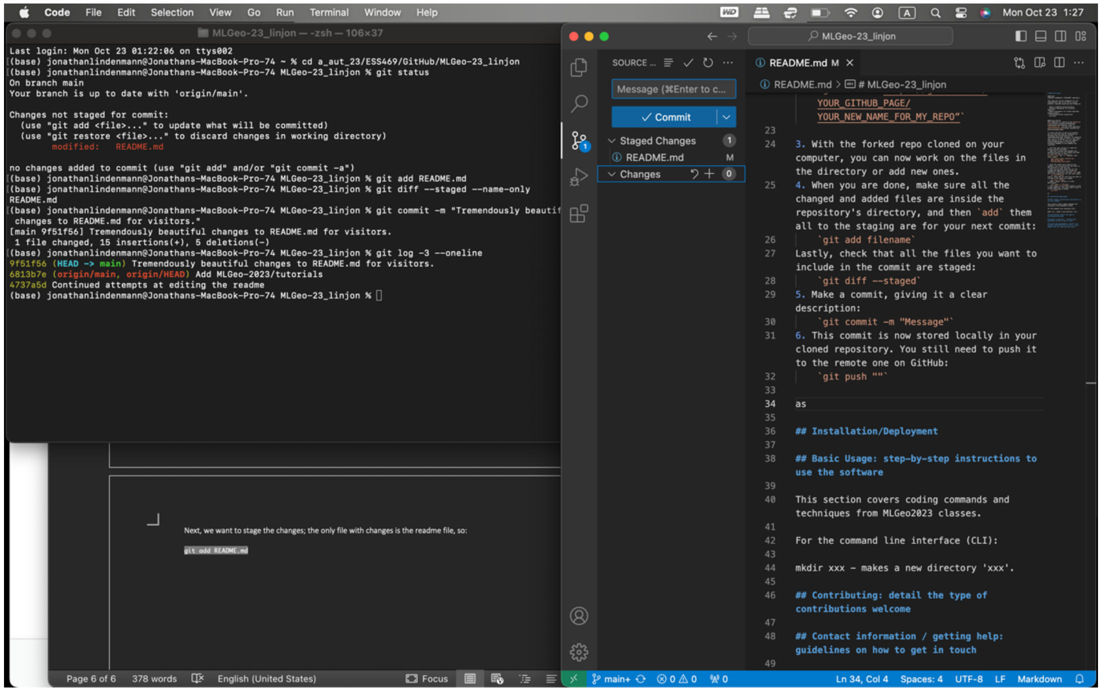
```
git diff --staged –name-only
```

Now we can commit the changes and check if it worked with `git log` :

```
git commit -m “Tremendously beautiful changes to README.md for visitors.”
```

```
git log -3 --oneline
```

The image below shows all these steps in the terminal on the left, with the changes open in the editor on the right. The VSCode UI for the same commit process is visible there:



```
Last login: Mon Oct 23 01:22:06 on ttys002
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % cd a_aut_23/ESS469/GitHub/MLGeo-23_linjon
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git add README.md
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git diff --staged --name-only
README.md
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git commit -m "Tremendously beautif
 changes to README.md for visitors."
[main 9f51f56] Tremendously beautiful changes to README.md for visitors.
 1 file changed, 15 insertions(+), 5 deletions(-)
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git log -3 --oneline
9f51f56 (HEAD -> main) Tremendously beautiful changes to README.md for visitors.
6813b7e (origin/main, origin/HEAD) Add MLGeo-2023/tutorials
4737a5d Continued attempts at editing the readme
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % 
```

```
    YOUR_GITHUB_PAGE/
    YOUR_NEW_NAME_FOR_MY_REPO"`

3. With the forked repo cloned on your
computer, you can now work on the files in
the directory or add new ones.
4. When you are done, make sure all the
changed and added files are inside the
repository's directory, and then `add` them
all to the staging are for your next commit:
    `git add filename`
Lastly, check that all the files you want to
include in the commit are staged:
    `git diff --staged`
5. Make a commit, giving it a clear
description:
    `git commit -m "Message"`
6. This commit is now stored locally in your
cloned repository. You still need to push it
to the remote one on GitHub:
    `git push ""`

as

## Installation/Deployment

## Basic Usage: step-by-step instructions to
use the software

This section covers coding commands and
techniques from MLGeo2023 classes.

For the command line interface (CLI):

mkdir xxx - makes a new directory 'xxx'.

## Contributing: detail the type of
contributions welcome

## Contact information / getting help:
guidelines on how to get in touch

```

Next, we want to stage the changes; the only file with changes is the readme file, so:

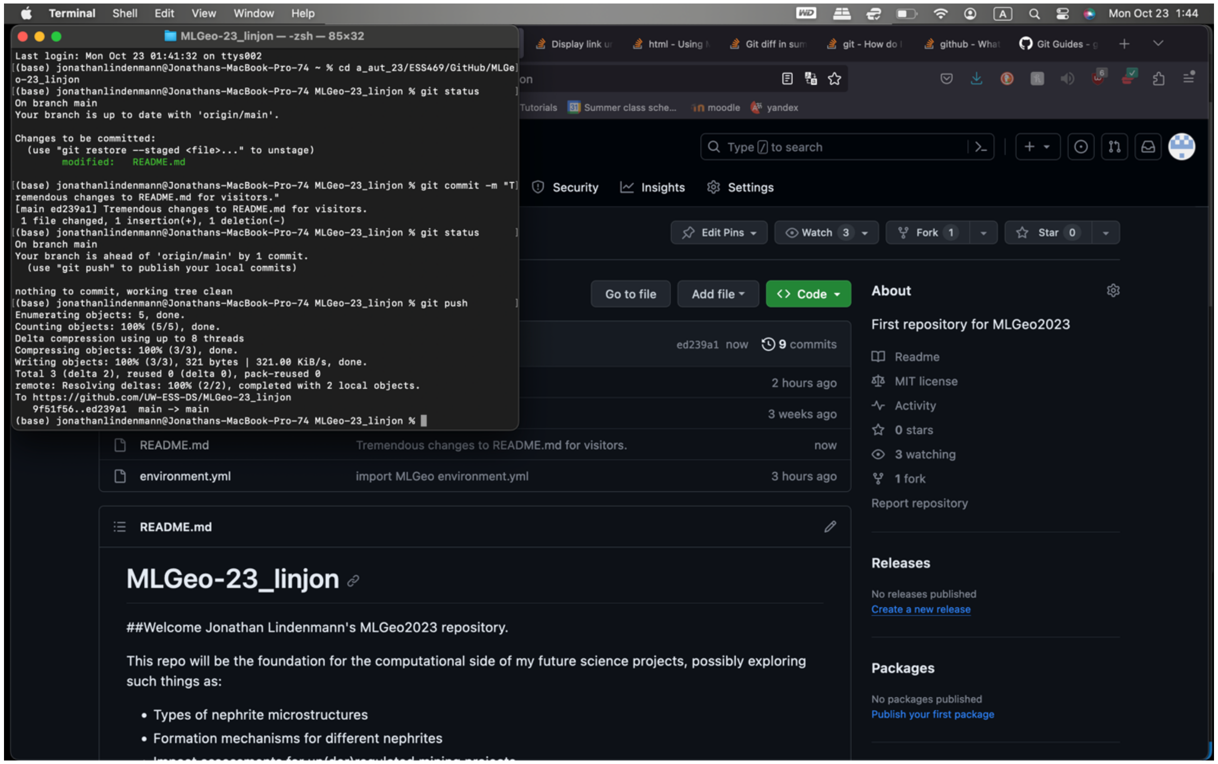
```
git add README.md
```

5. The final step is to push the commit to your remote forked repo on GitHub (you can then go on to create a pull request on GitHub to have the code merged with the main repo). However, it's best to first pull the repo, just in case you are missing some fresh changes on your local version:

   ```
   git pull
   ```

   ```
   git push
   ```

Now you've pushed the commits onto Github. And you're all done! See the image below or the final steps in Terminal and the changes reflected on the online Github repository.



```
Last login: Mon Oct 23 01:41:32 on ttys002
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 ~ % cd a_aut_23/ESS469/GitHub/MLGe
o-23_linjon
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git commit -m "T
remendous changes to README.md for visitors."
[main ed239a1] Tremendous changes to README.md for visitors.
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 321 bytes | 321.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/UW-ESS-DS/MLGeo-23_linjon
   9f51f56..ed239a1  main -> main
(base) jonathanlindenmann@Jonathans-MacBook-Pro-74 MLGeo-23_linjon % 
```

First repository for MLGeo2023

README.md — Tremendous changes to README.md for visitors. — now

environment.yml — import MLGeo environment.yml — 3 hours ago

README.md

MLGeo-23_linjon

##Welcome Jonathan Lindenmann's MLGeo2023 repository.

This repo will be the foundation for the computational side of my future science projects, possibly exploring such things as:

- Types of nephrite microstructures
- Formation mechanisms for different nephrites