

Git in five minutes

NSDF-Data student guide 8 — for every computer

Version 2 · 14 October 2026

Tip

This is guide 8. It explains the few Git ideas you need for this project, in plain words. Read it once, now or later, and come back to it when a Git word or message puzzles you. You do not have to memorize anything: you will use four commands.

The idea

Git keeps every saved version of your project, like an endless “Save As” where each save has a note saying what changed. Nothing you have saved is lost, even if you change your mind later. **GitHub** is the website that keeps a shared copy online, so your lead can see your work.

The words

Word	What it means here
repository	The project folder (NSDF-Data) together with its whole saved history.
commit	One saved version, with a short message saying what changed.
branch	A separate line of saved versions. Your branch is your own copy of the work.
push	Send your commits from your computer to GitHub.
pull request	A button on github.com that means “I’m done, please have a look.” Your lead reads it and, if it is ready, adds your work to the project. Guide 4 shows it step by step.
develop and master	The shared branches that belong to the project. You never work on them.

Your branch

Everything you do happens on your own branch, named `student-yourname`. The setup guide (Step 8) made it for you. Your branch cannot hurt anyone else’s work or the public site, however

many times you commit or push.

To see which branch you are on, type:

```
git branch --show-current
```

It should print `student-yourname`. If it prints `develop` or `master`, stop and ask your lead.

The four commands

You will use these all the time. Run them in a terminal, from inside the NSDF-Data folder (the first two guides show how to get there).

```
git status
git add your-file
git commit -m "Say what you changed"
git push
```

Command	What it does
<code>git status</code>	Shows what has changed since your last commit, and which branch you are on. It only looks: it never changes anything, so use it as often as you like.
<code>git add your-file</code>	Chooses a file to include in your next commit. Use the real file name, as <code>git status</code> shows it.
<code>git commit -m "..."</code>	Saves a version of everything you added, with your message. A good message says what and why, in a few words.
<code>git push</code>	Sends your saved versions to GitHub. The first time on a new branch, Git tells you the exact command to use: copy it.

Tip

Before every commit, type `git status` and read the list. It should show only your own files. If you see data files (a folder named `idx`, or `.bin` files) or other people's files, do not add them: ask your lead.

A normal session

1. Go to the NSDF-Data folder in a terminal.
2. Type `git branch --show-current` and check that it is your branch.
3. Do your work: a notebook, a note.
4. Type `git status`, then `git add` for each file of yours, then `git commit -m "..."`, then `git push`.
5. When a note or a piece of work is ready for review, open a pull request on github.com (guide 4, Step 3).

Getting the newest project changes

The project keeps changing. To bring the newest shared changes into your branch, follow *Updating your setup* (guide 7, Step 2). It uses two lines, `git fetch origin` and `git merge --no-edit origin/develop`, and tells you what to do if Git reports a CONFLICT: stop, and send the message to your lead.

Prefer buttons to typing?

Everything above can also be done with buttons in VS Code. Guide 9 (*Git in VS Code*) shows how. It is optional: the terminal version stays the main one.

When something looks wrong

Careful

Stop, and do not try commands you found on the internet. Type `git status`, copy everything it prints, and send it to your lead together with the message that worried you. Work you have committed is very hard to lose, and nobody minds this question.

You see	What it means	What to do
nothing to commit, working tree clean	You have no unsaved changes.	Nothing: this is fine.
Untracked files:	Files Git has not been told about yet.	Add the ones that are yours. Leave data files alone.
Your branch is ahead of 'origin/...' by 2 commits	You committed but have not pushed.	<code>git push</code>
Your branch is behind	There are newer commits on GitHub.	Ask your lead, or see guide 7, Step 2.
CONFLICT	Git cannot combine two changes by itself.	Stop. Do not edit anything. Send the message to your lead.
A text editor opens (a window full of ~ characters, or text starting Merge branch)	Git wants a message for a merge.	Press Esc, type :wq, press Enter.
fatal: not a git repository	You are not inside the NSDF-Data folder.	Go to the NSDF-Data folder and try again.
Please tell me who you are, or a password or permission error when you push	Git does not know your name, or you are not signed in to GitHub.	Setup guide, Step 8. Never type your GitHub password into the terminal.

You may hear about “Git Flow”

Git Flow is the set of branches (develop, release/..., hotfix/...) the project lead uses to publish versions. It is the maintainers’ business. You only ever use your own branch and the pull request

button.

Where to learn more

These are outside websites, which you can read at your own pace. None is required.

- **Software Carpentry, “Version Control with Git”**: a gentle course for scientists. <https://swcarpentry.github.io/git-novice/>
- **GitHub: About pull requests**: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- **GitHub Skills**: short interactive courses in your browser. <https://skills.github.com/>
- **Pro Git**, the free book, for later. <https://git-scm.com/book/en/v2>
- **Dangit, Git!?!:** plain recipes for “I did something wrong”. <https://dangitgit.com/>

Session Info

Guide version 2, 14 October 2026. Written for every computer. Please tell the project lead where a step was unclear.

Where to get help

Anthony Villano, project lead: anthony.villano@ucdenver.edu

When you write, include the output of `git status` and the message that worried you.

Key links

- Repository: <https://github.com/villano-lab/NSDF-Data>
- Student guides: <https://villano-lab.github.io/NSDF-Data/>