

DATASCI 350 - GitHub Tutorial

Danilo Freire

Contents

1	Introduction	2
2	What are Git and GitHub?	2
3	Setting up	2
3.1	Create a GitHub account	2
3.2	Apply to GitHub Education	3
3.3	Install GitHub Desktop	4
4	Getting the course materials	5
4.1	Fork the repository	5
4.2	Clone your fork	6
5	Saving your changes	9
5.1	Make changes	9
5.2	Commit	11
5.3	Push	11
6	Updating the course materials	12
7	Recommended readings and resources	13

1 Introduction

This tutorial gets you set up with GitHub: a free educational account, [GitHub Desktop](#) installed, a copy of the course materials on your computer, and the routine for keeping that copy up to date. It also covers the basics of Git along the way, including commits and pushes. The tutorial uses GitHub Desktop throughout, but any Git client will do the same job if you already have one you prefer.

2 What are Git and GitHub?

[Git](#) is a version control system. It records every change you make to a set of files, so you can see what changed, when, and why, and go back to any earlier state. [Linus Torvalds](#) wrote it in 2005 to manage the [Linux](#) kernel, and it is now the *de facto standard* for the job. Git is also famously hard to learn, which is where GitHub comes in.

[GitHub](#) hosts Git repositories on the web. It stores your project in the cloud, keeps every version, and lets several people work on the same files without overwriting each other. It adds issue tracking, project management, and [free website hosting](#) on top. [Microsoft](#), [Google](#), and [Meta](#) all keep code there¹, as does almost every open-source project. Think of it as a social network for people who write code.

We use GitHub throughout this course, so it is worth getting comfortable with it. Both public and private repositories are free for students.

3 Setting up

3.1 Create a GitHub account

Go to <https://github.com/> and sign up, using the button in the top right. Use your Emory email address, which qualifies you for the educational account described below. Choose your username carefully, as it is public and it will follow you.

¹As a curiosity, the code for the [Apollo 11 mission](#) is also available on GitHub.

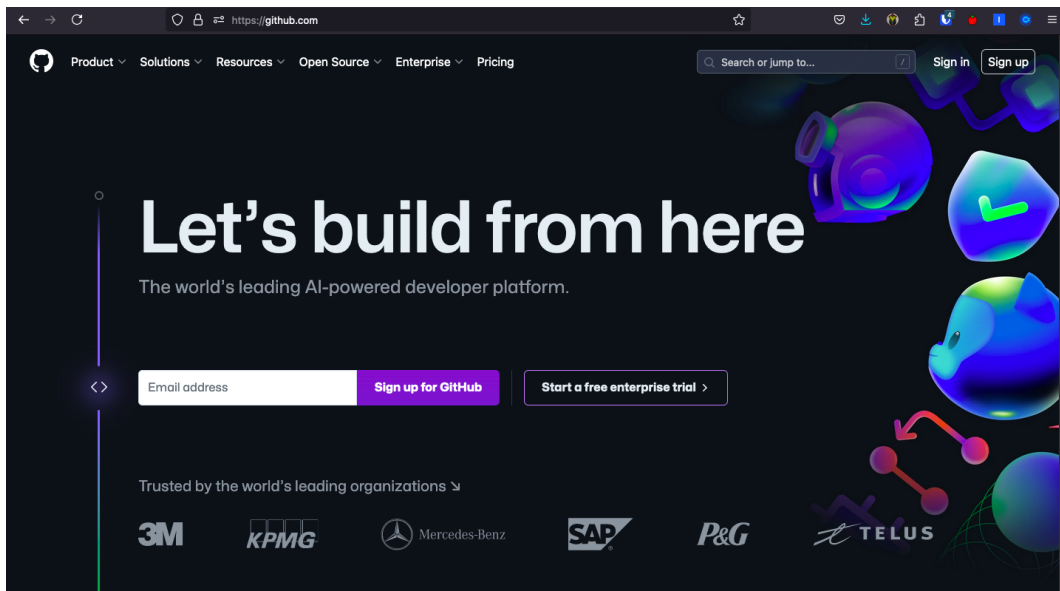


Figure 1: Signing up for GitHub.

3.2 Apply to GitHub Education

Go to <https://github.com/education> and apply for the Student Developer Pack. It gives you a set of paid tools for free, including GitHub Copilot, a domain name, and cloud hosting credit. Click “Join GitHub Education” and follow the instructions. You will need proof of your student status, such as an ID card or a transcript.

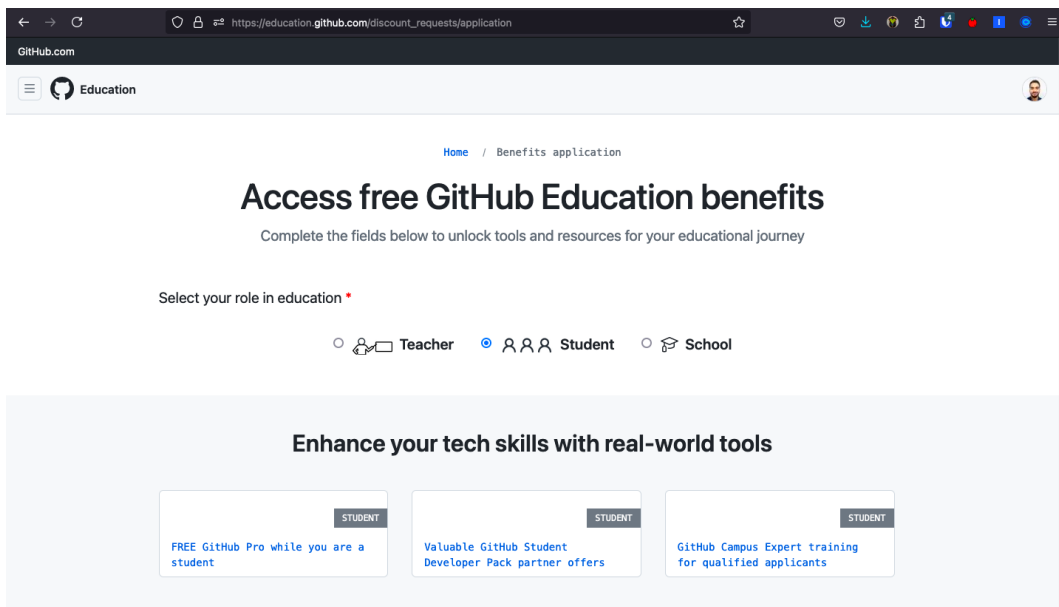


Figure 2: The GitHub Education page.

3.3 Install GitHub Desktop

Download GitHub Desktop from <https://desktop.github.com/>. It is a graphical interface to Git for Windows and macOS, and it is the tool this tutorial uses. The command line does the same things, and you are welcome to use it instead.

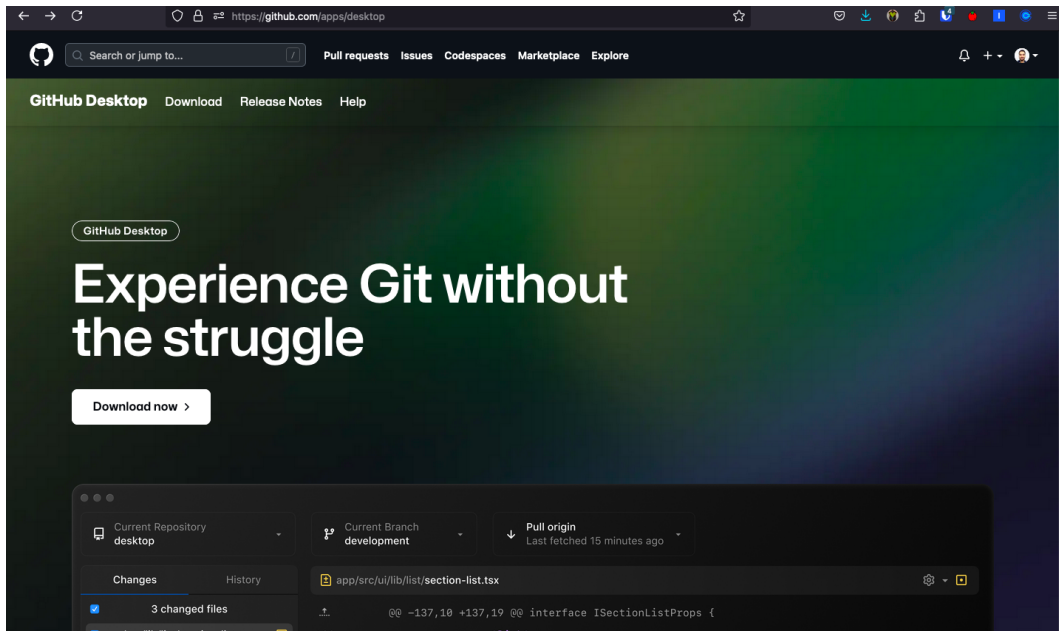


Figure 3: The GitHub Desktop website.

Open GitHub Desktop. Sign in with your GitHub account. You should then see this screen:

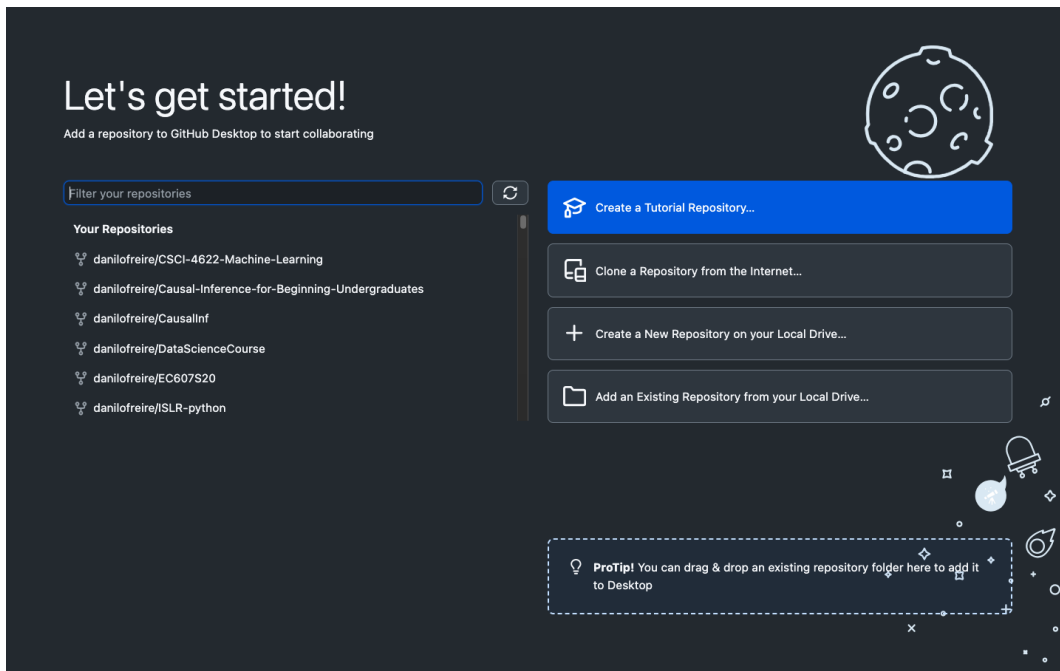


Figure 4: GitHub Desktop after signing in.

4 Getting the course materials

4.1 Fork the repository

The course materials live at <https://github.com/danilofreire/datasci350>. You will make your own copy, called a *fork*, so that you can annotate and change the files without touching mine.

Open the repository page. Click “Fork” in the top right corner.

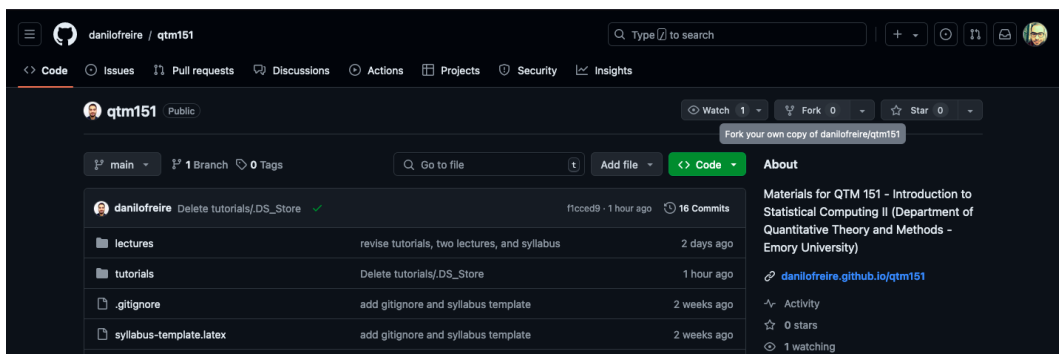


Figure 5: Forking the repository.

GitHub asks you to name your copy. Any name works, but `datasci350` keeps it consistent with the instructions in these tutorials. Click “Create fork”.

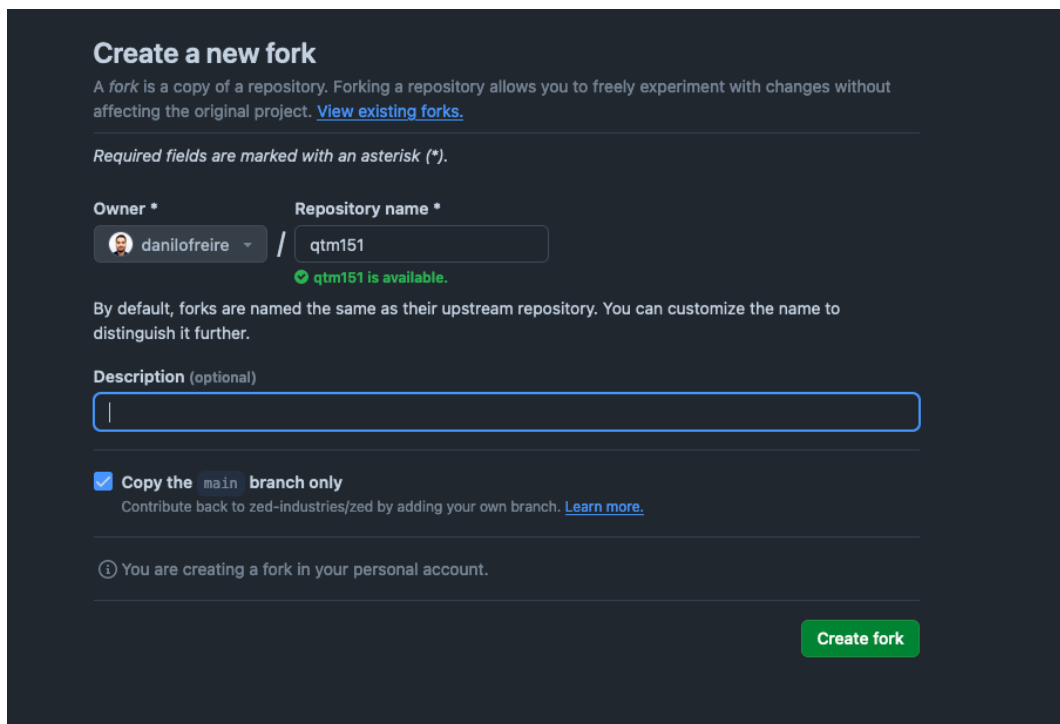
The image shows the GitHub 'Create a new fork' interface. At the top, it says 'Create a new fork' and explains that a fork is a copy of a repository. Below this, it states 'Required fields are marked with an asterisk (*)'. The form has two main sections: 'Owner *' and 'Repository name *'. The 'Owner' dropdown is set to 'danilofreire'. The 'Repository name' text box contains 'qtm151', and a green checkmark below it says 'qtm151 is available.'. Below these fields, there is a note: 'By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.' Then, there is a 'Description (optional)' text box. Below that, there is a checkbox labeled 'Copy the main branch only', which is checked. A note below the checkbox says 'Contribute back to zed-industries/zed by adding your own branch. Learn more.' At the bottom, there is a note with an information icon: 'You are creating a fork in your personal account.' and a green 'Create fork' button.

Figure 6: Naming your copy of the repository.

You do this once, at the start of the semester. From then on you pull my updates into your fork and push your own work back to it.

4.2 Clone your fork

Forking gave you a copy on GitHub's servers. Cloning brings that copy onto your computer.

Open your fork on GitHub. Check the URL: it should have *your* username in it, not mine. Click the green "Code" button. Copy the URL.

Open GitHub Desktop. Click "Clone a Repository from the Internet".

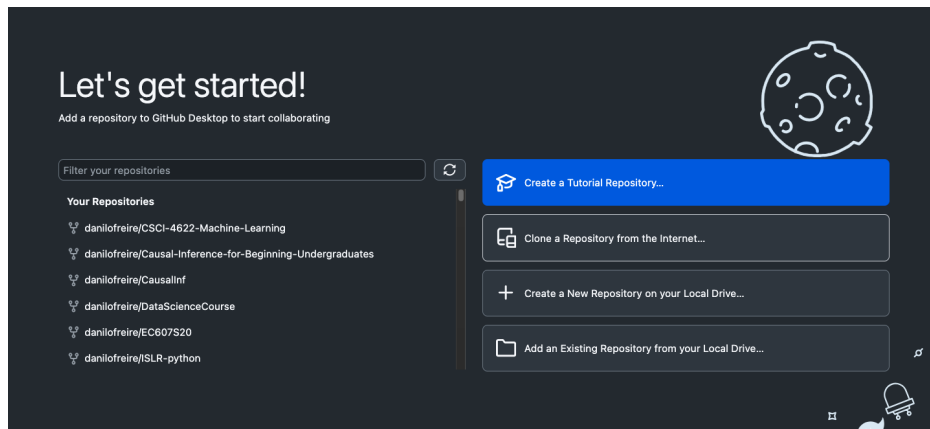


Figure 7: Click “Clone a Repository from the Internet”.

Click the “URL” tab. Paste the URL. Choose where to save the repository on your computer. Click “Clone”.

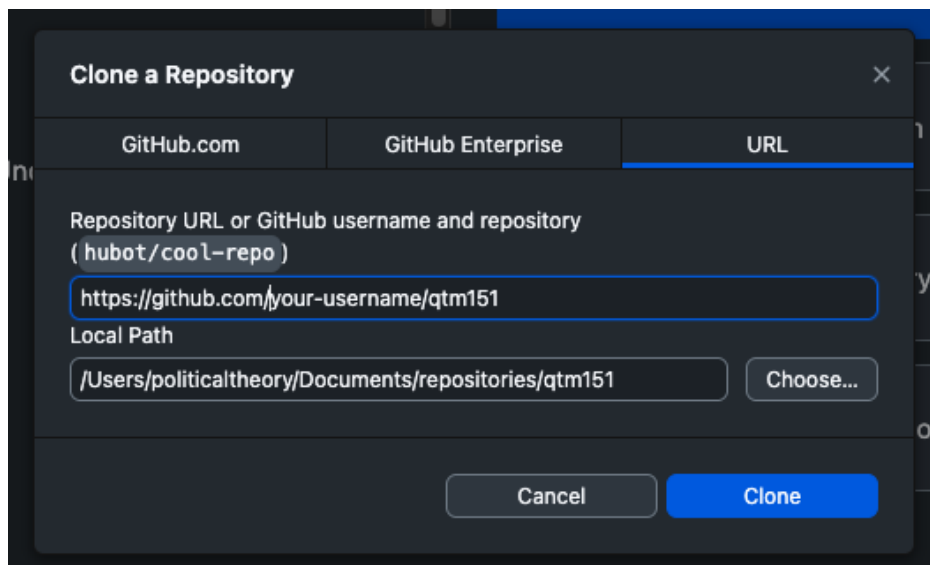


Figure 8: Paste the URL of your fork, then click “Clone”.

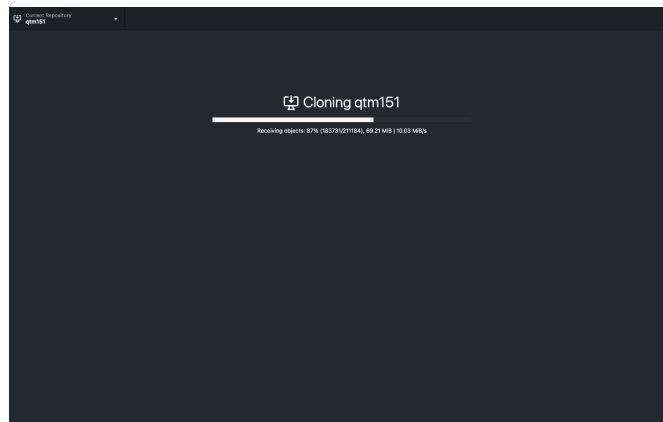


Figure 9: Cloning in progress.

GitHub Desktop then asks how you plan to use the fork. Select “To contribute to the parent project”. Click “Continue”. This is the option that lets you pull my updates later.

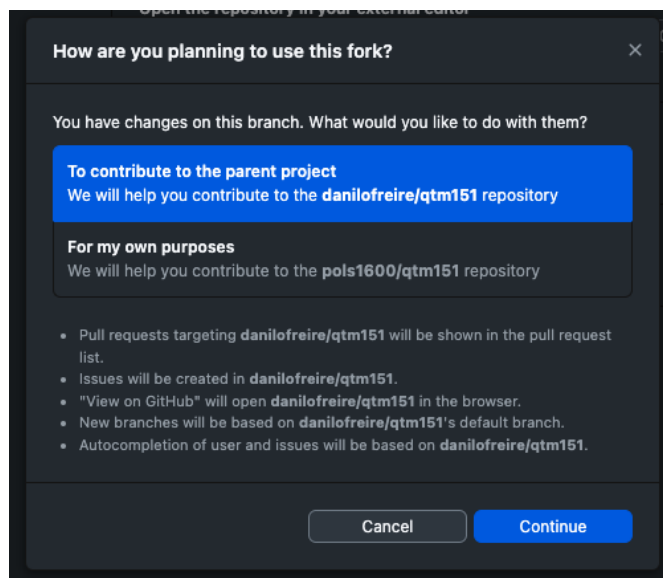


Figure 10: Select “To contribute to the parent project”.

If you pick the other option by mistake, your copy has no link back to my repository and you will never see an update. To fix it: right-click the repository in GitHub Desktop and remove it, delete the folder from your computer, then clone again.

You now have the course materials on your machine. To open them in VS Code, click “Repository” > “Open in Visual Studio Code”.

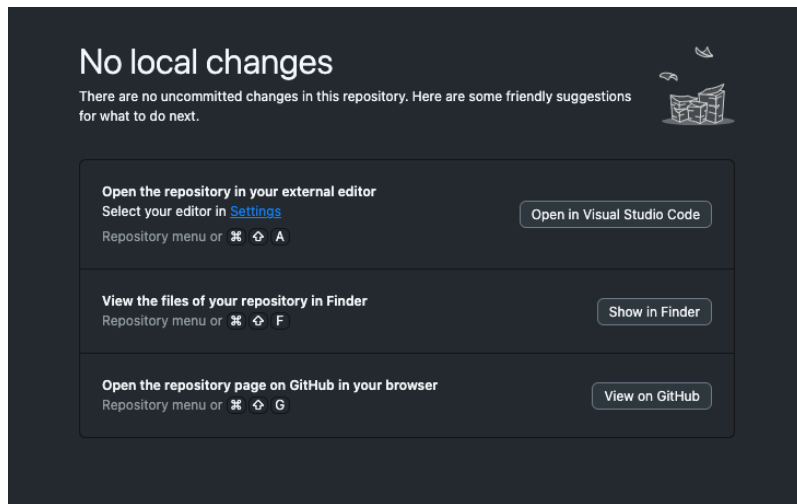


Figure 11: Opening the repository in VS Code.

5 Saving your changes

Git splits saving into two steps. A *commit* records a set of changes on your computer. A *push* sends those commits to GitHub. You can commit ten times offline and push once.

5.1 Make changes

Edit the files as you normally would. GitHub Desktop notices, and lists every changed file in the “Changes” tab.

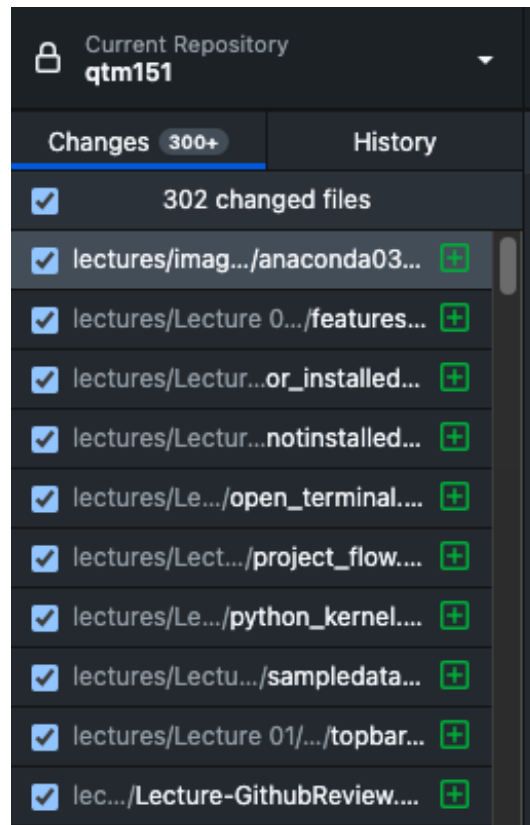


Figure 12: Changed files in the “Changes” tab.

Click a file to see what changed inside it. Green lines are additions, red lines are deletions. The “History” tab shows the same view for every commit made so far.

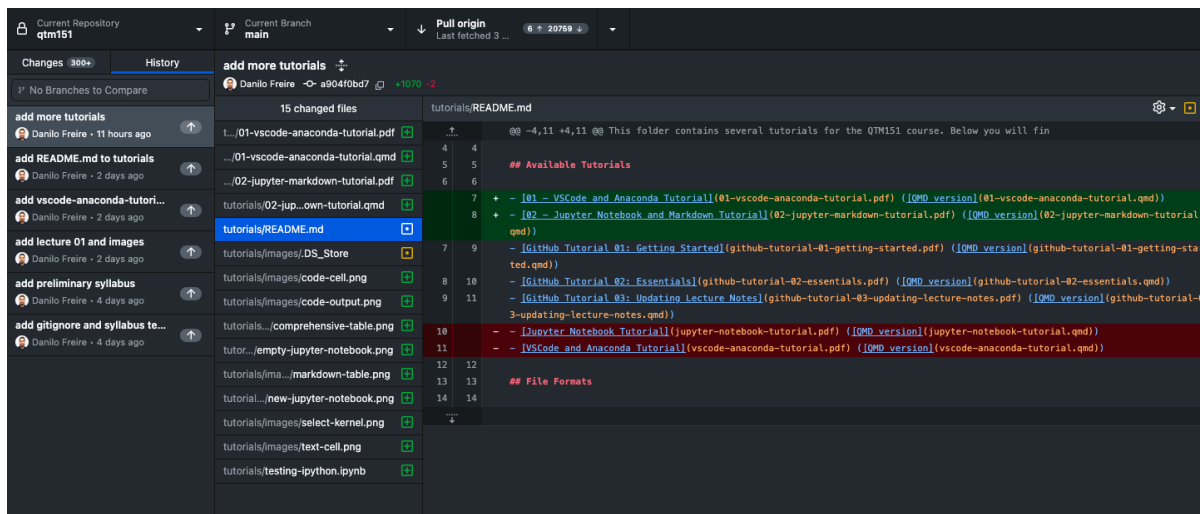


Figure 13: The history of changes to the repository.

5.2 Commit

Write a short summary of what you changed in the “Summary” box. Click “Commit to main”². That snapshot is now permanent, and you can return to it at any point.

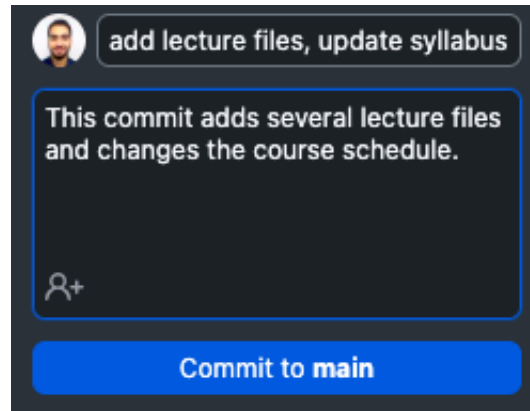


Figure 14: Committing changes.

Write summaries you will understand in three weeks. “Fixed question 4 of problem set 2” is useful; “update” is not.

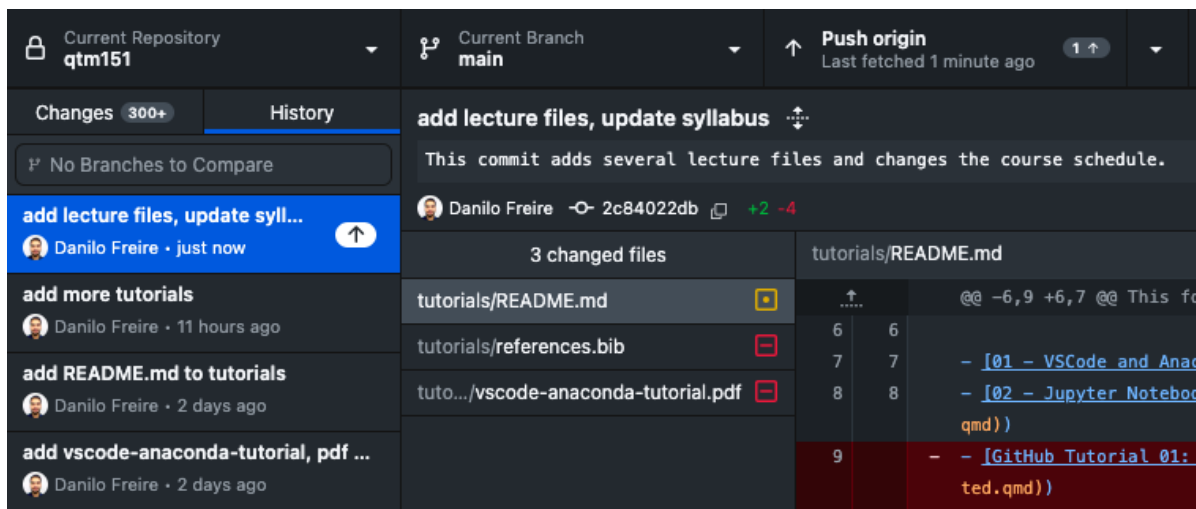


Figure 15: The commit and its message in the history.

5.3 Push

Your commits are still only on your computer. Click “Repository” > “Push” to send them to GitHub.

²The default branch in GitHub is called main. This is the branch where you will make most of your changes. You can create other branches if you want to work on different features or bug fixes.

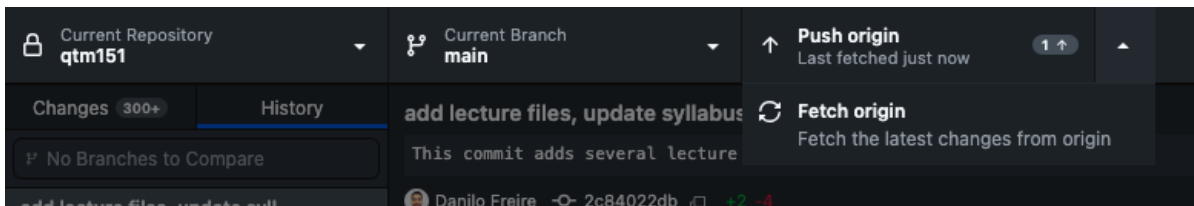


Figure 16: Pushing and fetching changes.

The neighbouring “Fetch” button does the reverse: it downloads changes from GitHub to your computer. The next section puts it to work.

6 Updating the course materials

I update the course materials regularly during the semester. Pulling those updates into your fork takes one click.

Commit everything you have been working on first. An update with uncommitted changes in the way can fail or overwrite your work.

Open the “Branch” tab in the top menu. Click “Update from upstream/main”. Here “upstream” means my repository, and “main” means its primary version.

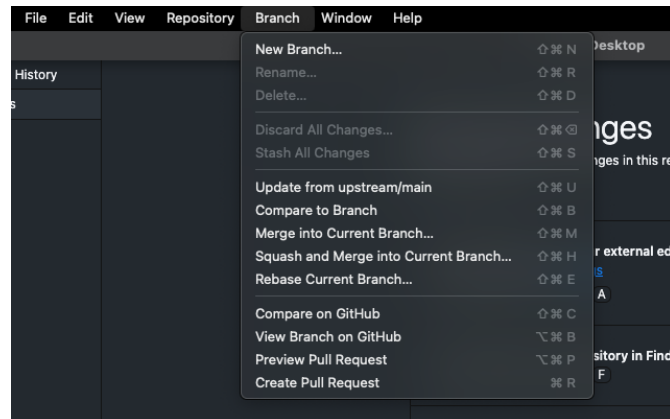


Figure 17: Updating from upstream.

Two habits make this painless:

- Update at the start of every class, so you are working from the current version.
- Save your own work under a different file name. If you write your answers into `assignment-01.qmd` and I then update that file, my version replaces yours. `assignment-01-danilo.qmd` is safe.

Your fork is public by default. To make it private, open your repository on GitHub, click “Settings”, scroll to the “Danger Zone”, click “Make private”, and confirm.

7 Recommended readings and resources

- [GitHub Skills](#): short interactive courses run inside real repositories. The best starting point.
- [GitHub Guides](#): short written guides to the basics.
- [GitHub Desktop documentation](#): the official reference for the app used here.
- [Pro Git](#): a free book covering Git in full. Read it when you outgrow the app.
- [How to Use Git & GitHub Desktop Tutorial for Beginners](#): a video walkthrough, if you prefer watching to reading.

If anything here does not work, ask me or a teaching assistant. Good luck with your coding!