Introduction to Version Control with Git

Objectives

Version Control

- What is version control?
- Why use version control?

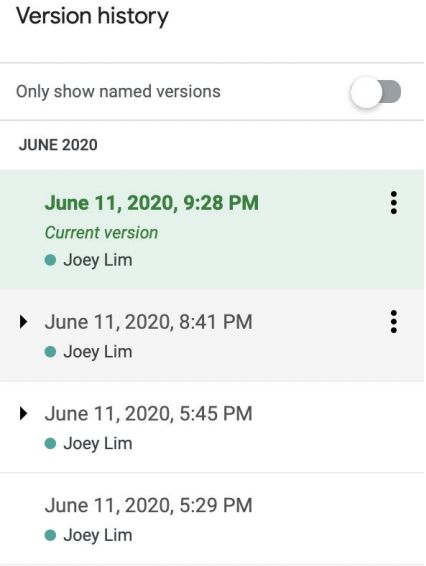
Commit & Revert

- Why make a commit?
- How to make a commit?
- Why revert a commit?
- How to revert a commit?

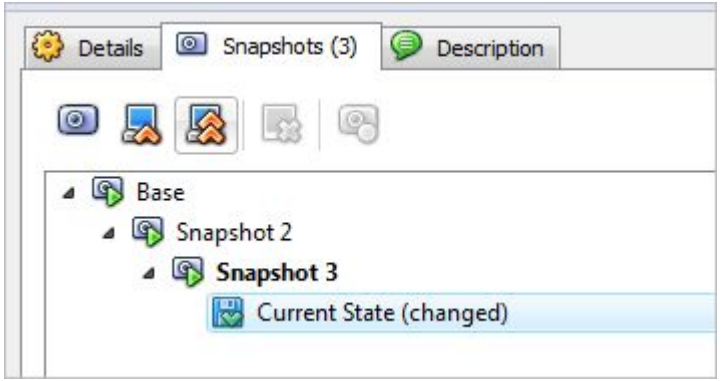
Push & Pull

- What is pushing?
- How do we push changes?
- What is pulling?
- How do we pull changes?

Version Control Is Not New



Google Docs Version History

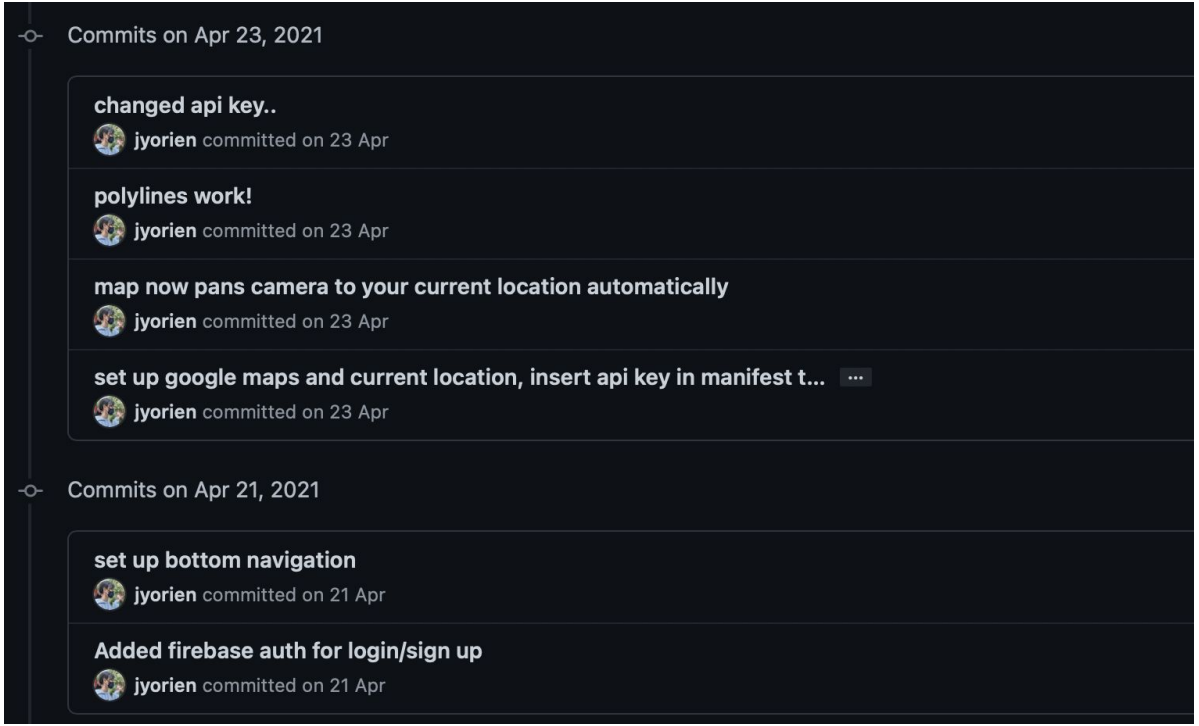


VirtualBox VM Snapshots

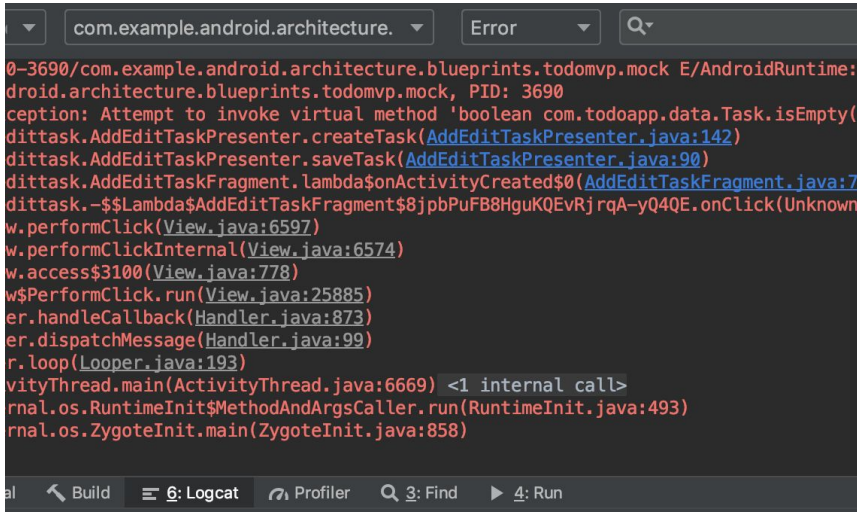
Time Travel?

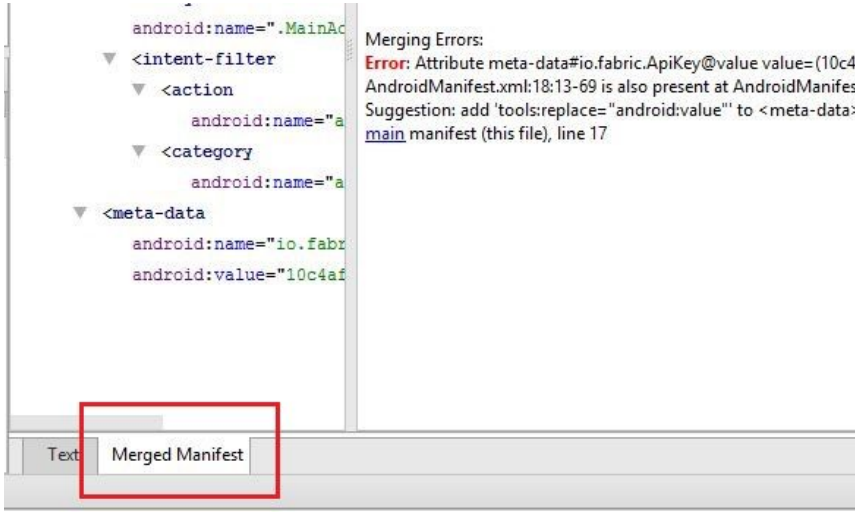
Version Control

- A change tracking system

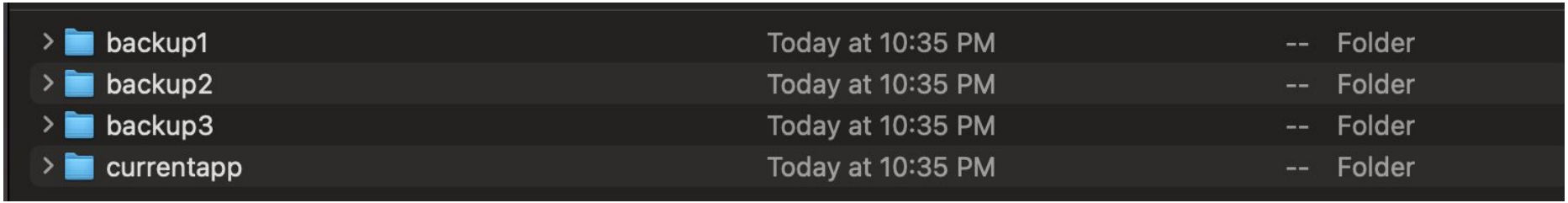


Why use Version Control?



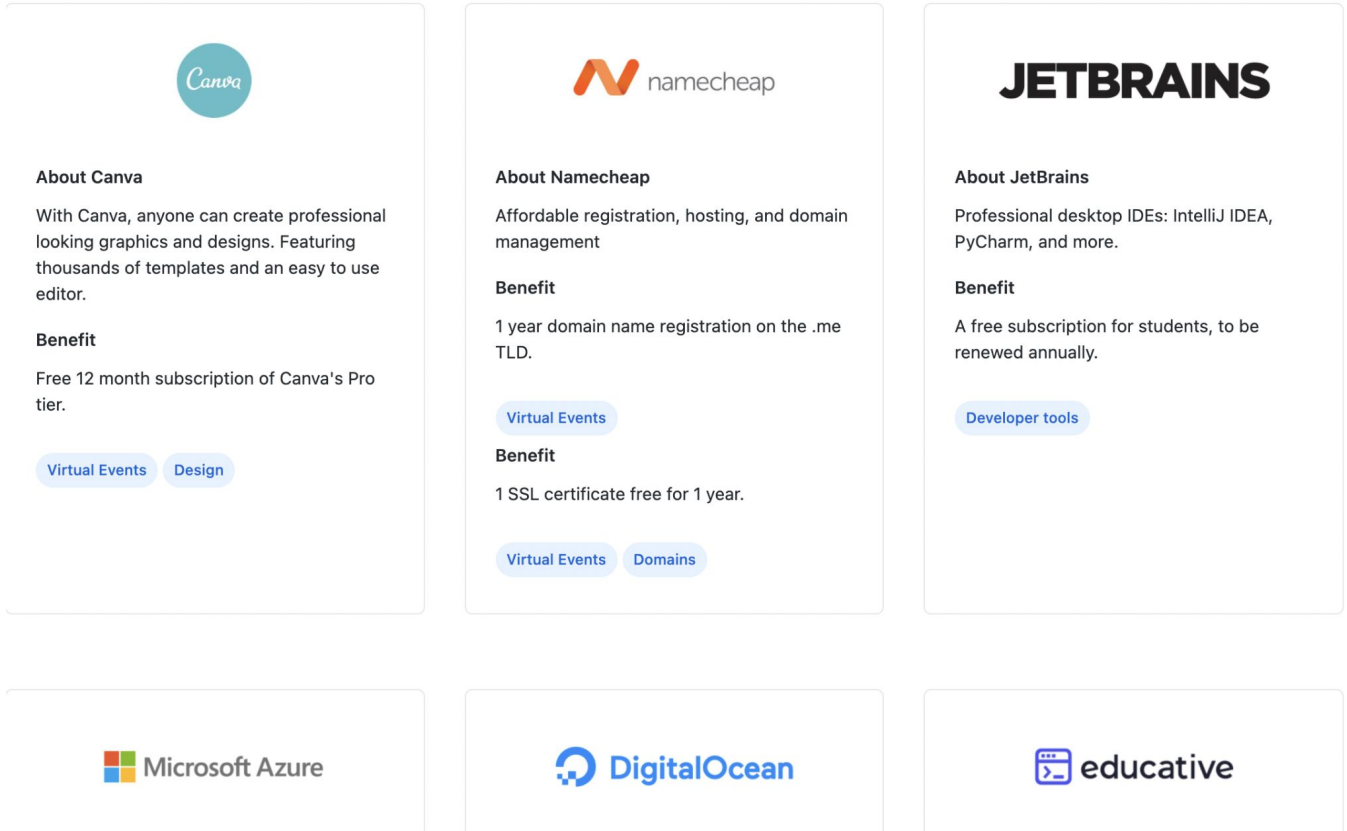


Why use Version Control?



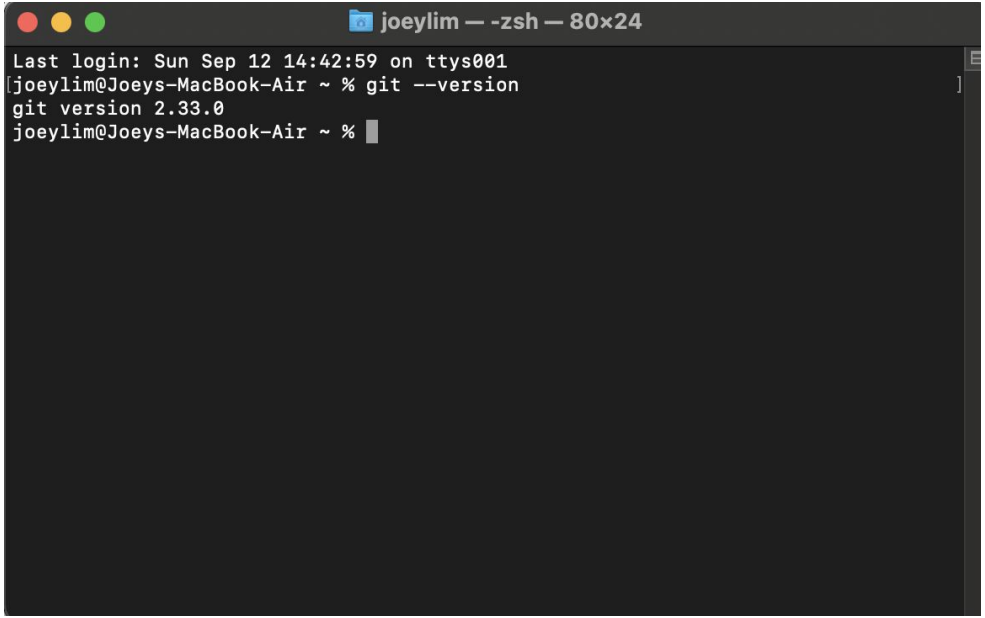
GitHub Student Developer Pack

https://education.github.com/pack



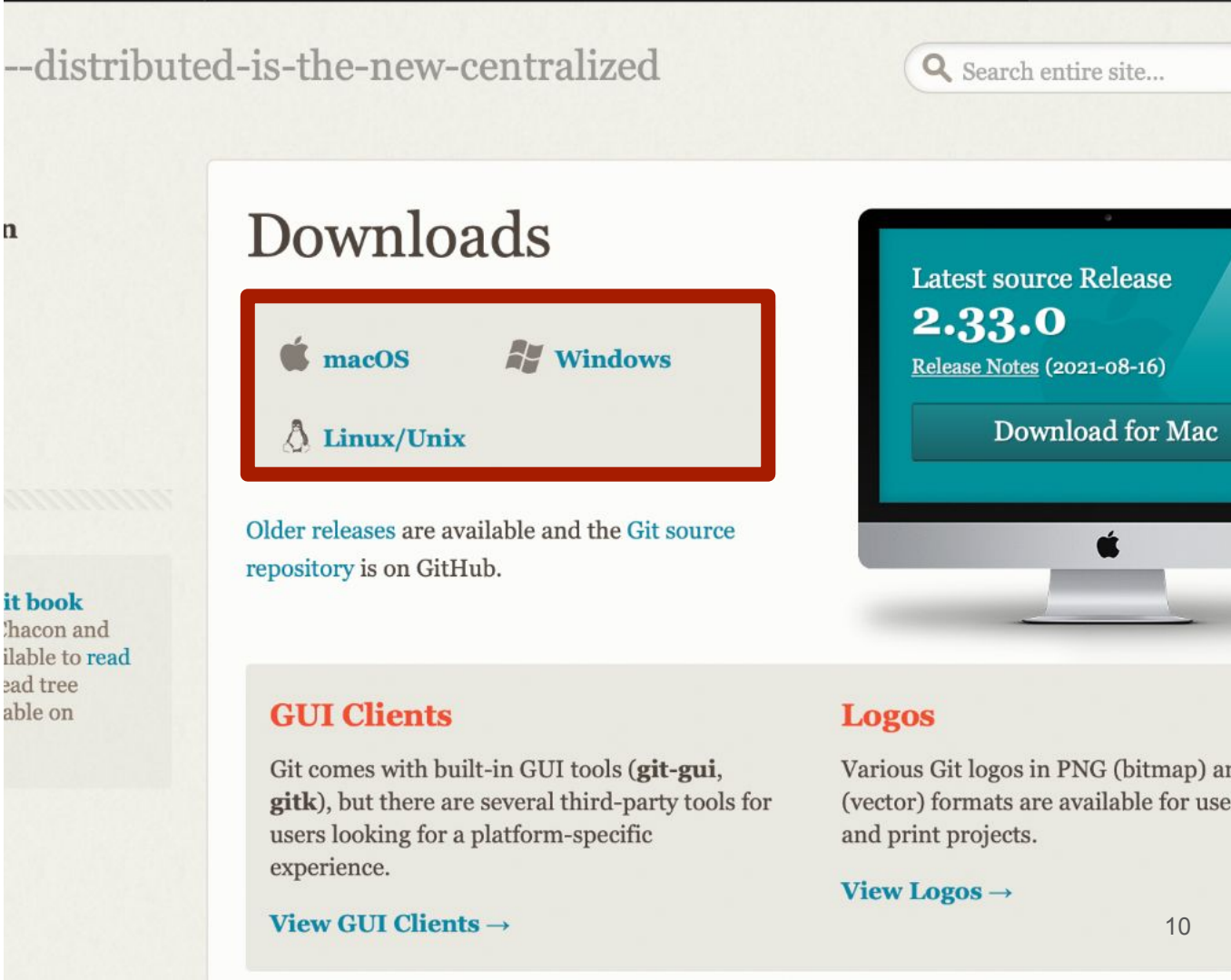
Download Git

- Windows Key and search ‘cmd’
- Type ‘git --version’ to see if git is already installed



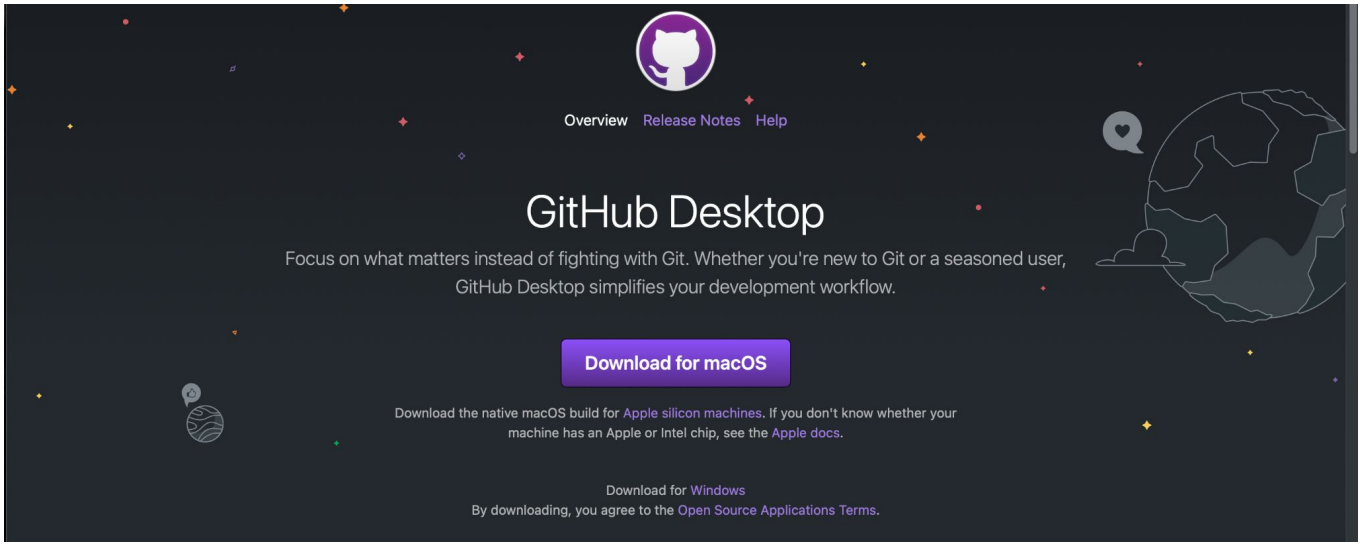
Download Git

- If not installed, go to https://git-scm.com/downloads



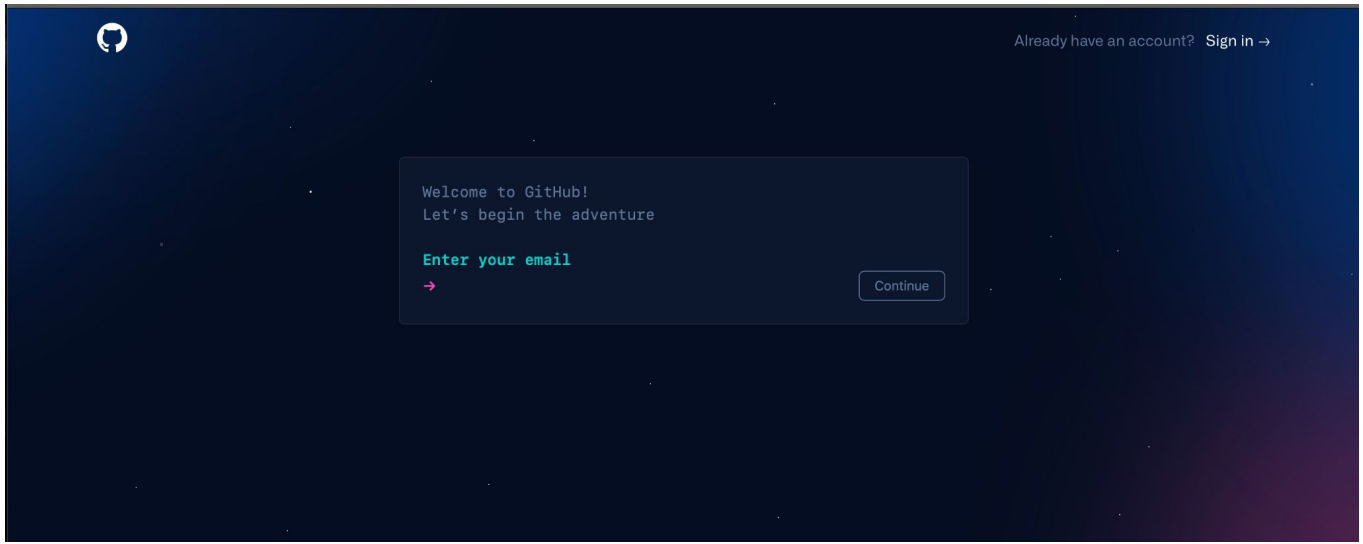
Download GitHub Desktop

- Go to https://desktop.github.com and download GitHub's desktop client



Sign Up for GitHub

- Go to https://github.com/signup and sign up for a GitHub account



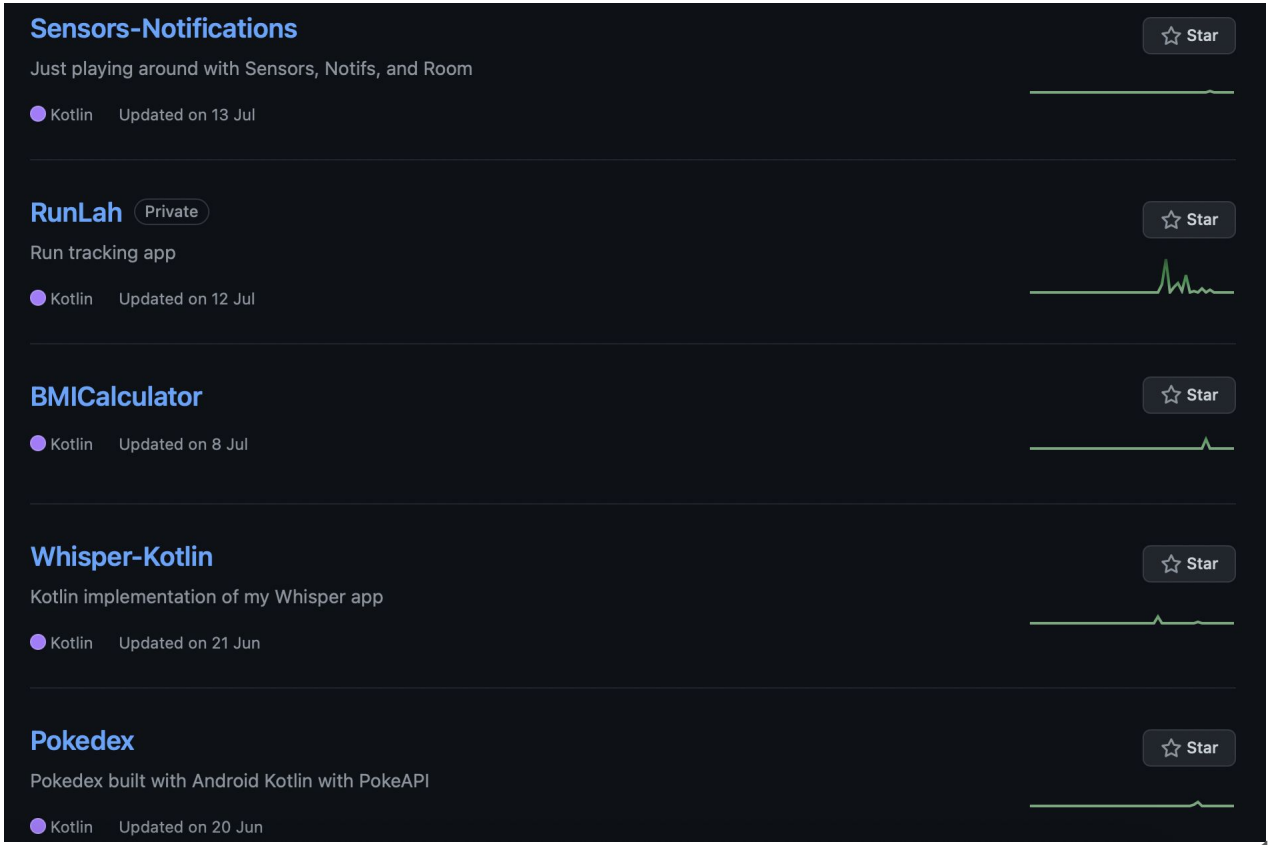
5 Minutes Break

- Download Git https://git-scm.com/downloads
- Download GitHub Desktop https://desktop.github.com
- Sign Up for GitHub account https://github.com/signup

  Note: Don’t worry if you don’t know how to set up, we will run through the installation with you!

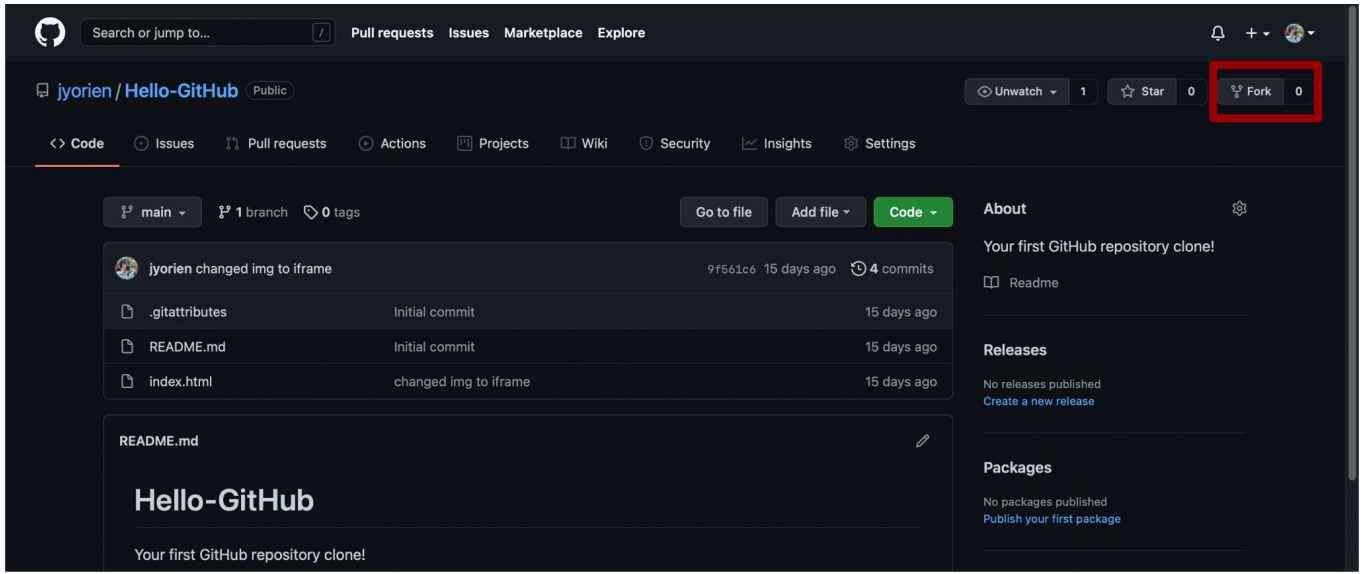
Repositories

- A repository or ‘repo’ is your project



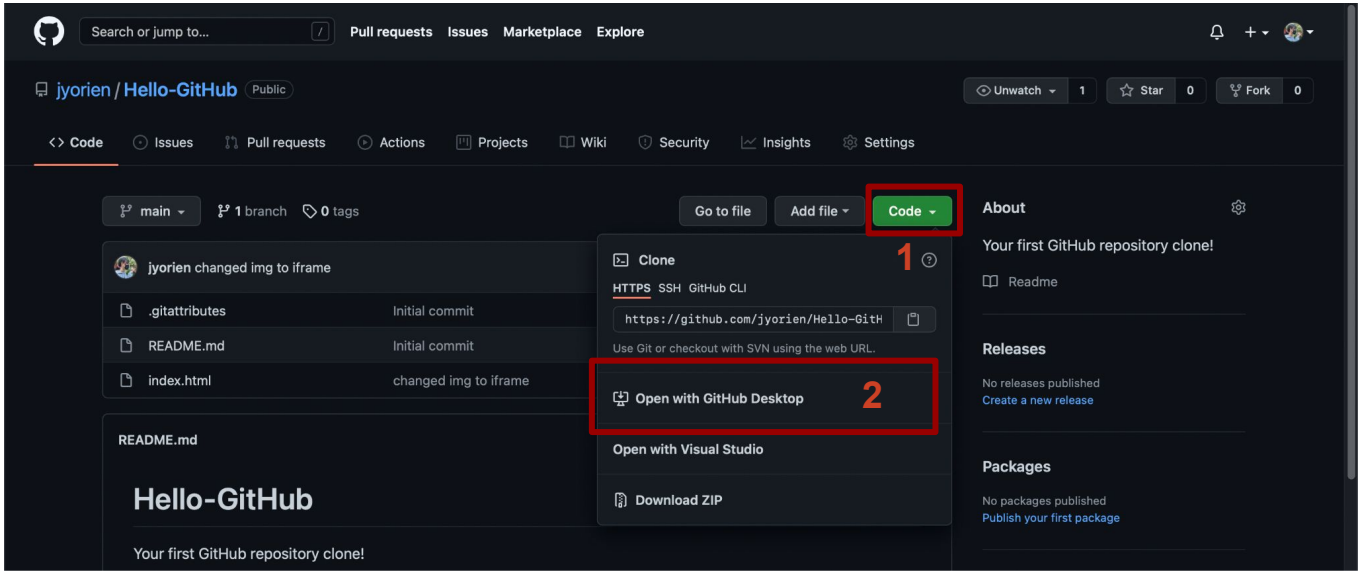
Forking

- Forking is creating a copy of the repo that belongs to your GitHub account
- Go to https://github.com/jyorien/Hello-GitHub and click ‘Fork’



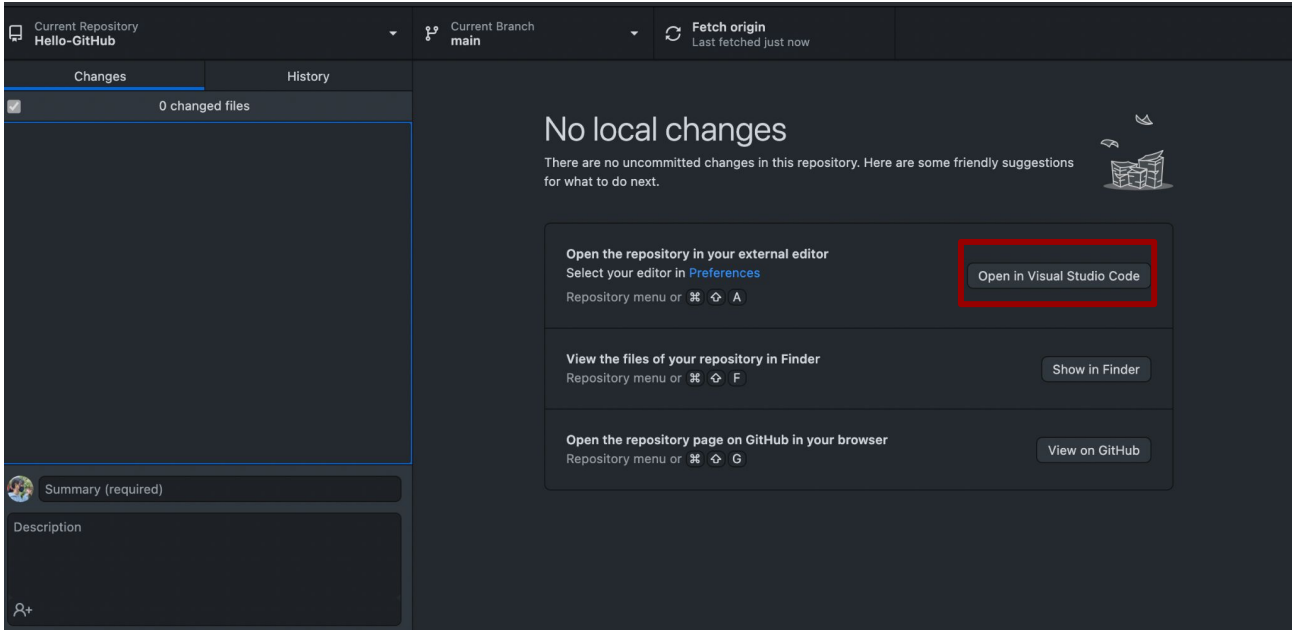
Clone

- Cloning is creating a LOCAL copy of the repo on your desktop
- Click Code -> Open with GitHub Desktop



Clone

- Congratulations! You just forked and cloned your first repo.



Commit & Revert

“Commit” and “Revert”

What is “Commit” ?

Commit is **saving** or **updating** the latest version of the repository.

Why “Commit” then?

- Keeps a record of the changes
- It acts as a checkpoint so you can refer back if needed.

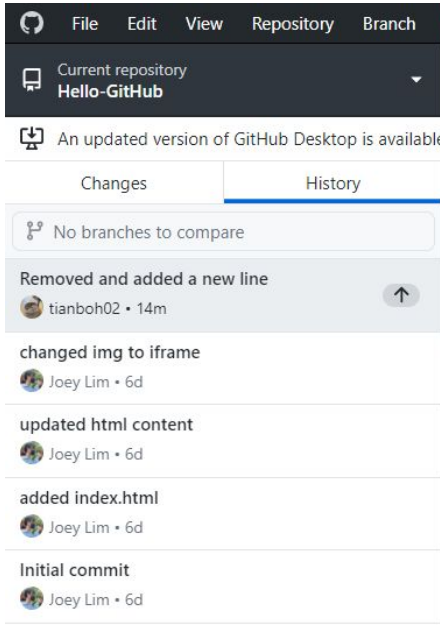
“Commit” and “Revert”

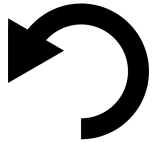
What is “Revert” ?

Revert is **undoing changes** back to a previous version of the repository.

Why “Revert” then?

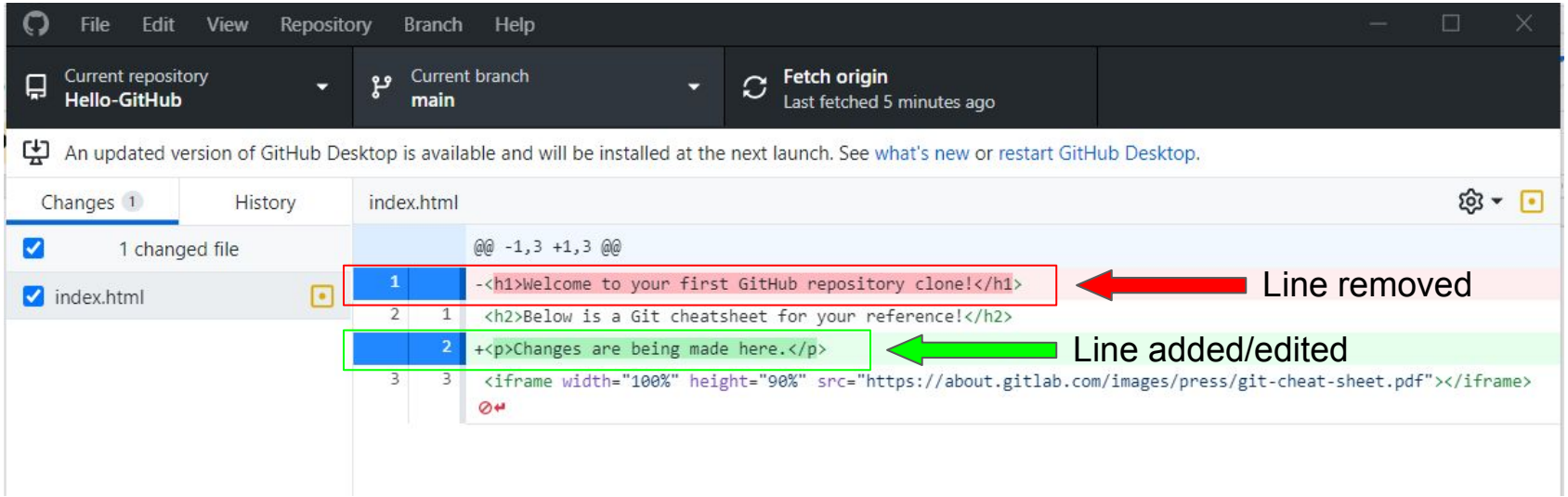
- Access or return back to the previous versions easily
- Saves you from starting your project/repository from scratch





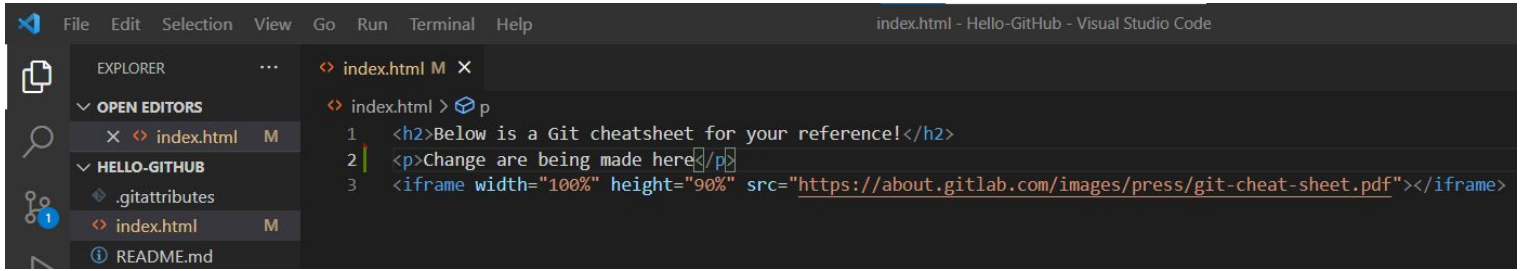
How to “commit”?

1. Open GitHub Desktop and locate your current repository.
2. If there are any new changes, it will appear as the following image below.



How to “commit”?

Before changes



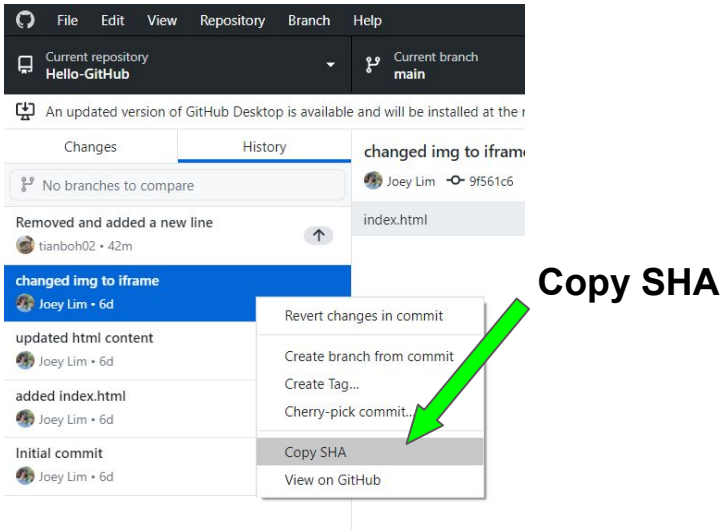
After changes

How to “commit”?

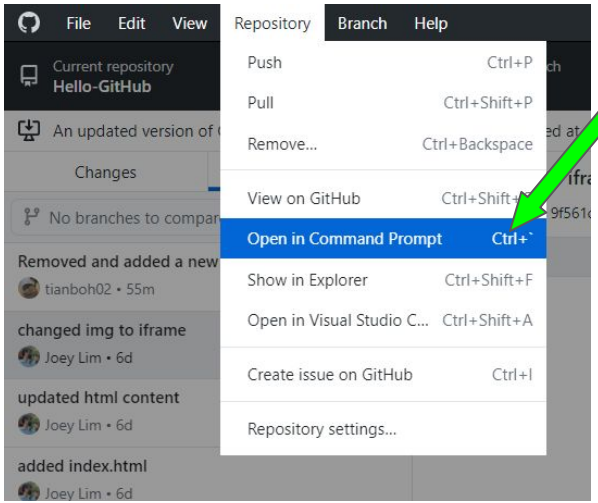
3. Add a title and description before committing (helps you to keep track)
4. Press on the button “Commit to main” to commit changes

How to “revert”?

1. Right click on the point that you want to revert to and click “copy SHA”.
2. Click on “Repository” and select “Open in Command Prompt”



Copy SHA

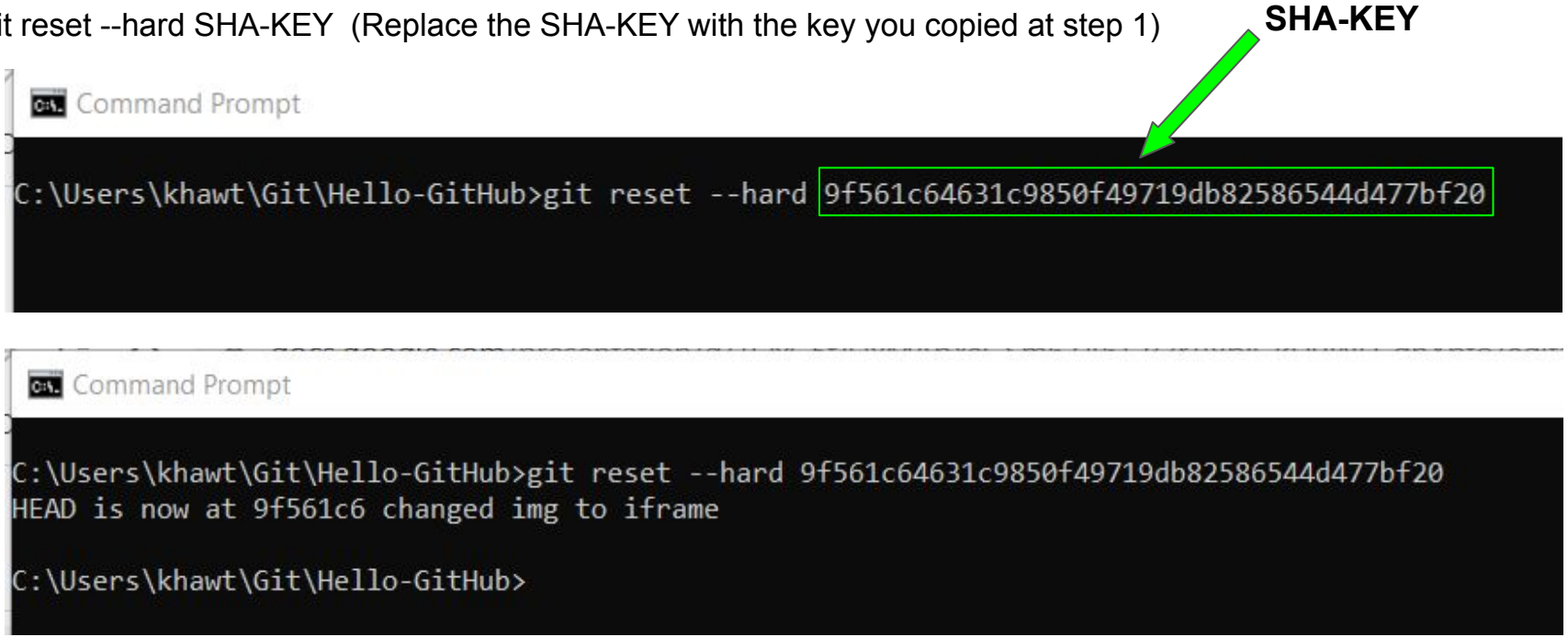


Open in Command Prompt

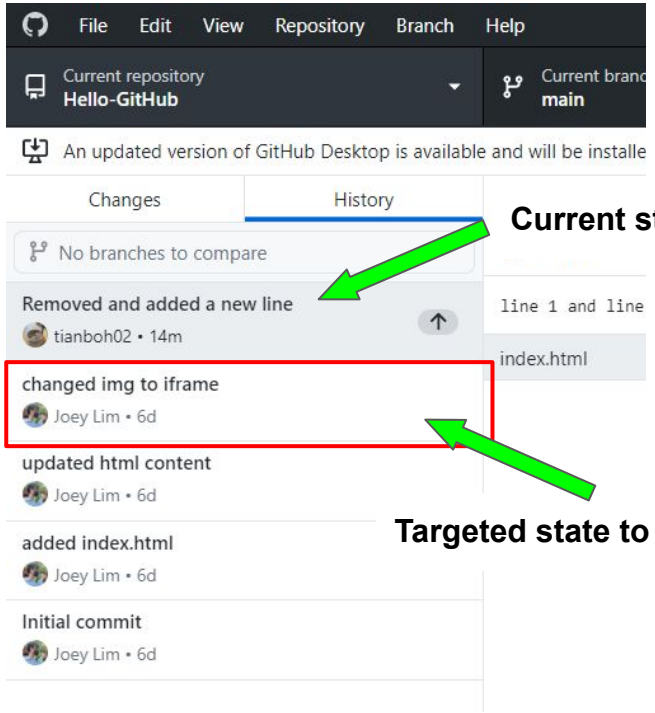
How to “revert”?

3. In the command prompt, type the following and enter:

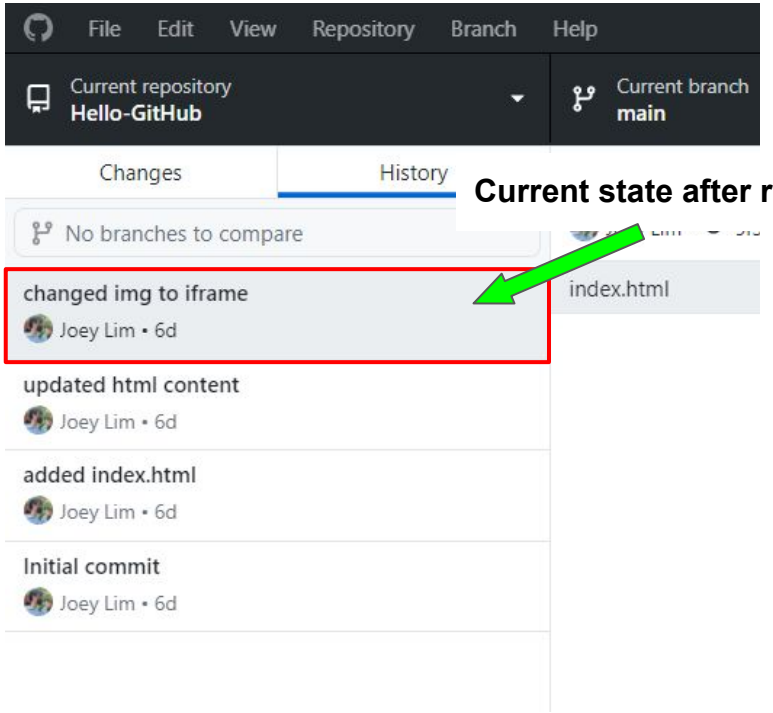
git reset --hard SHA-KEY (Replace the SHA-KEY with the key you copied at step 1)



How to “revert”?

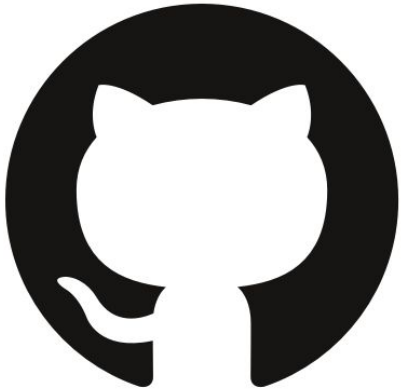


Before Revert



After Revert

Git and GitHub



Git **GitHub**

What really is Git

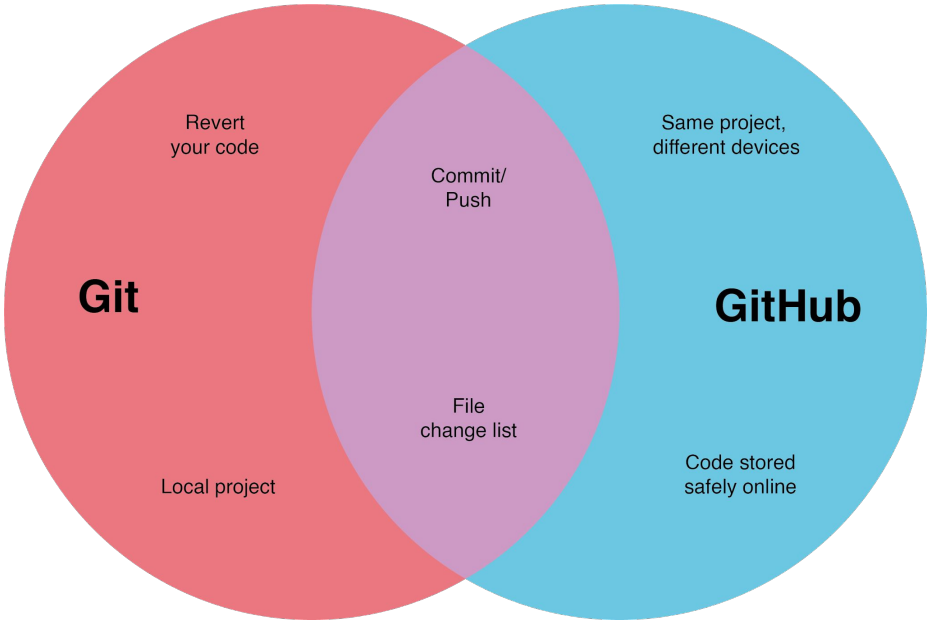
- Git is installed only on your local system
- Git is a tool which keeps a list of changes you have made to your code.
- Git helps you upload your code online to GitHub

What really is GitHub

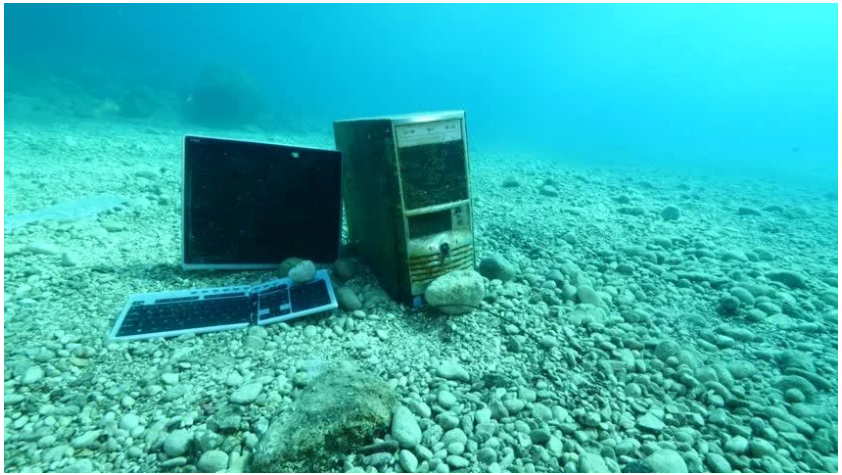
- GitHub is a free online service for storing your repositories
- Keeps your code safely stored online preventing loss of code
- Is a bit similar to Google Drive or OneDrive

Git without GitHub

- Git without GitHub is just a tool to show you what files in your repository have changed.
- Code is not backed up
- Only one device can edit the code



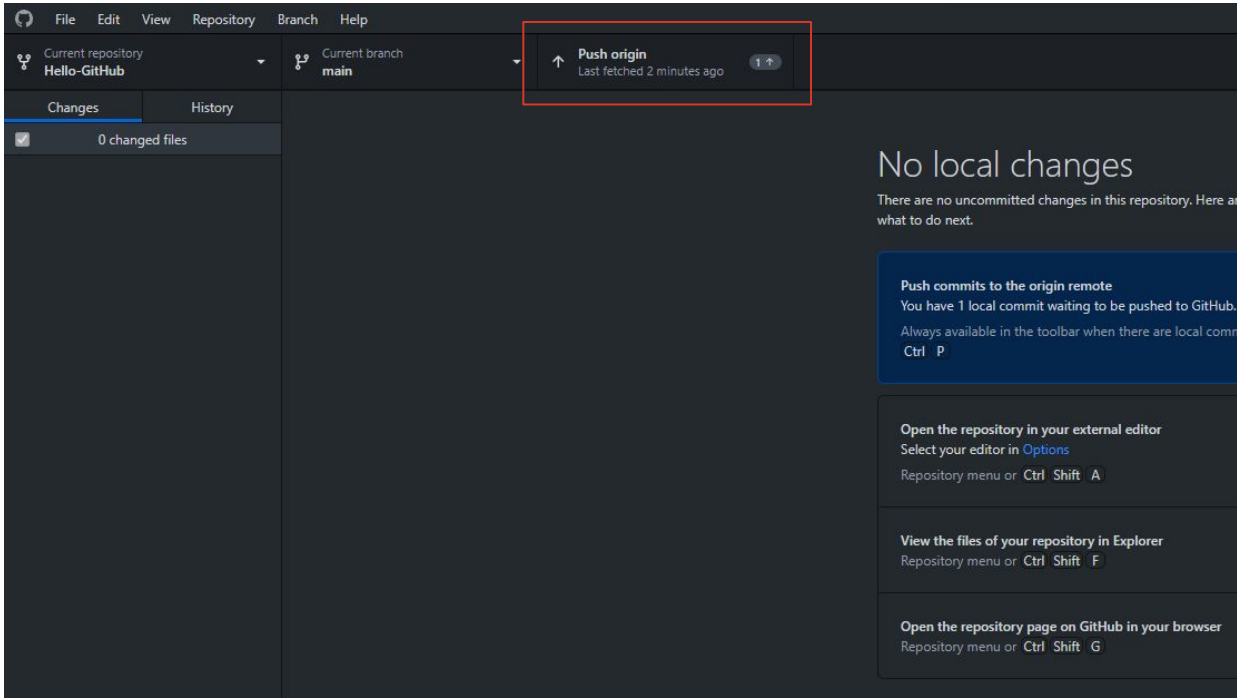
With GitHub, this won’t be a disaster



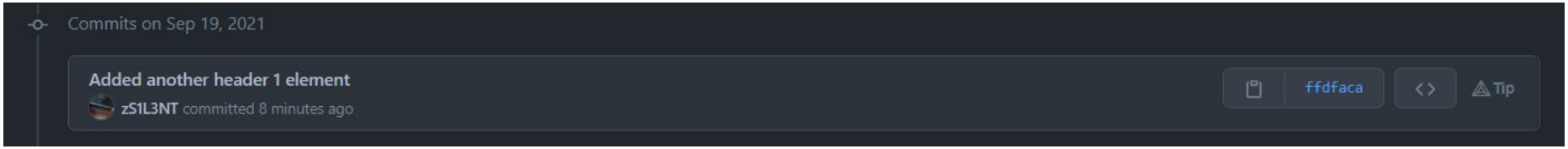
What does “Push to GitHub” mean?

A “Push to GitHub” means that you upload a commit along with the previous commits from your computer to the repository on GitHub.

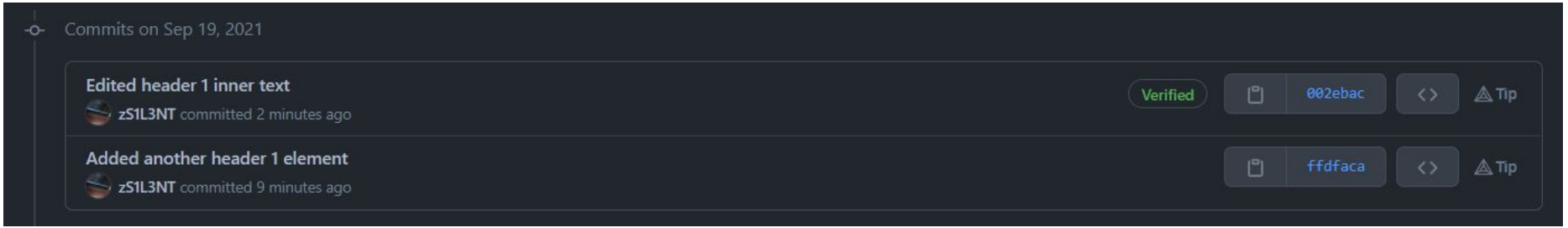
Once you make a commit to git, go to GitHub Desktop and click “Push origin”



Before and after the “Push” on GitHub



Before push

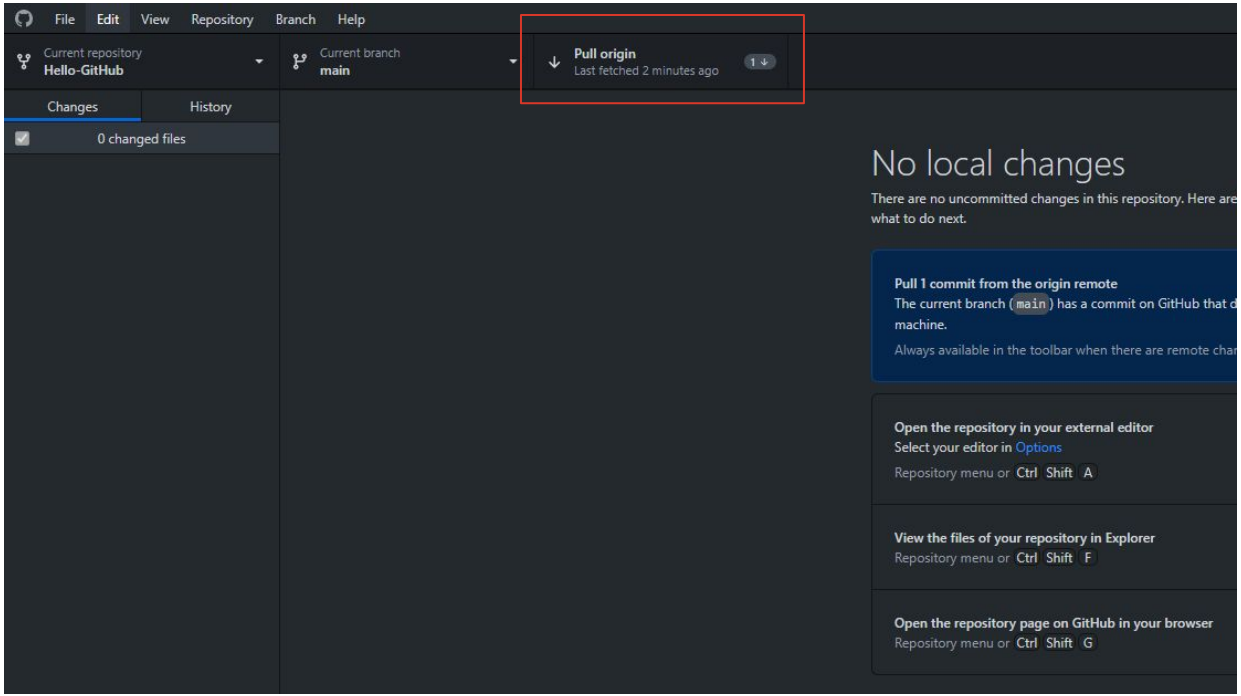


After push

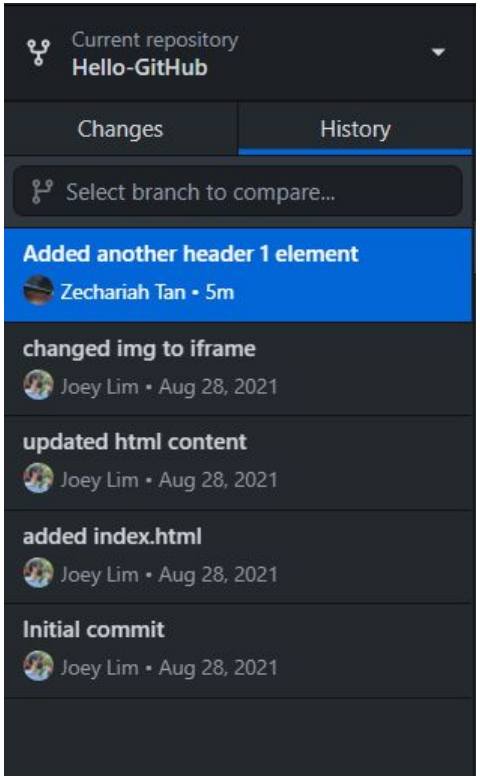
What does “Pull from GitHub” mean?

A “Pull from GitHub” means that you compare your local repository with the one on GitHub, then download the commits that you don’t have so you have the latest code

If you pushed code from your laptop and want the code on your desktop, go to GitHub Desktop and click “Pull origin” on your desktop

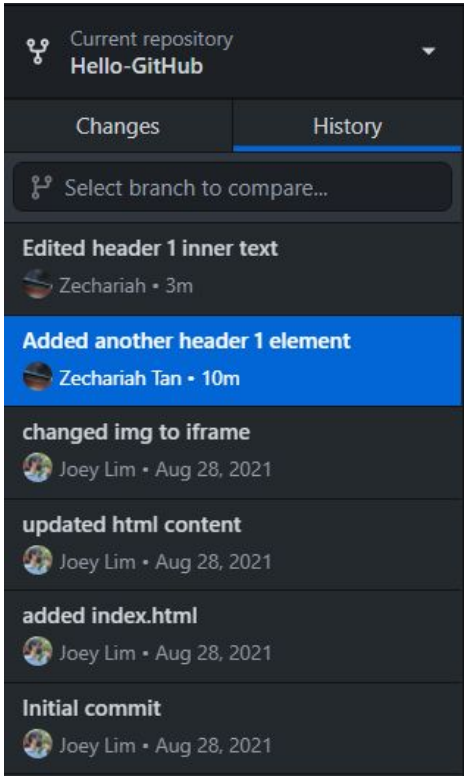


Before and after the “Pull” on GitHub Desktop

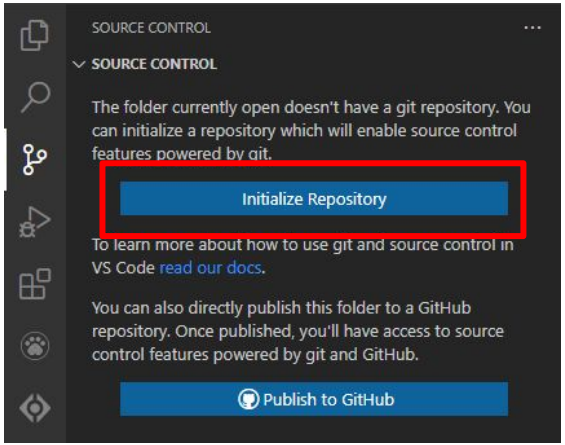


Before pull

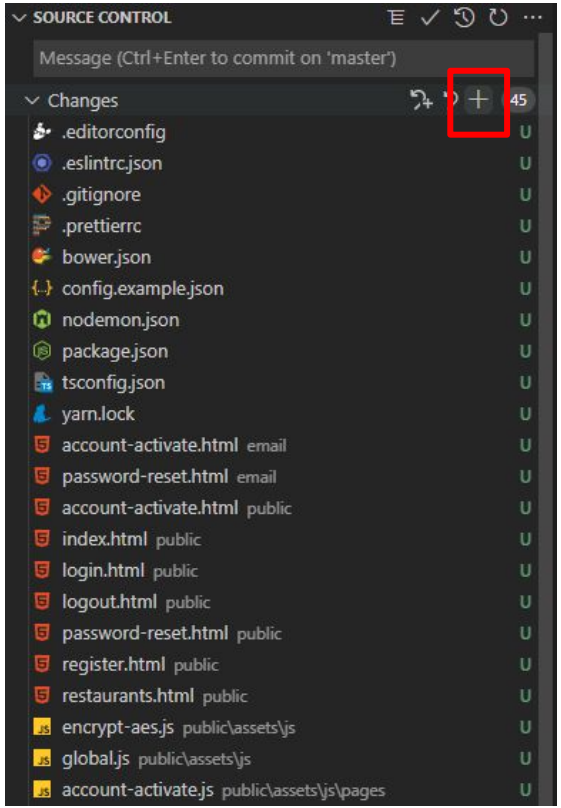
After pull



Adding Git to an existing project

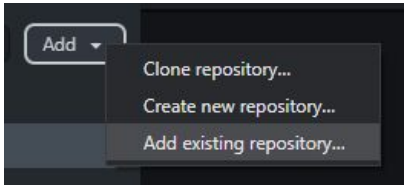


Initialise Git in your project

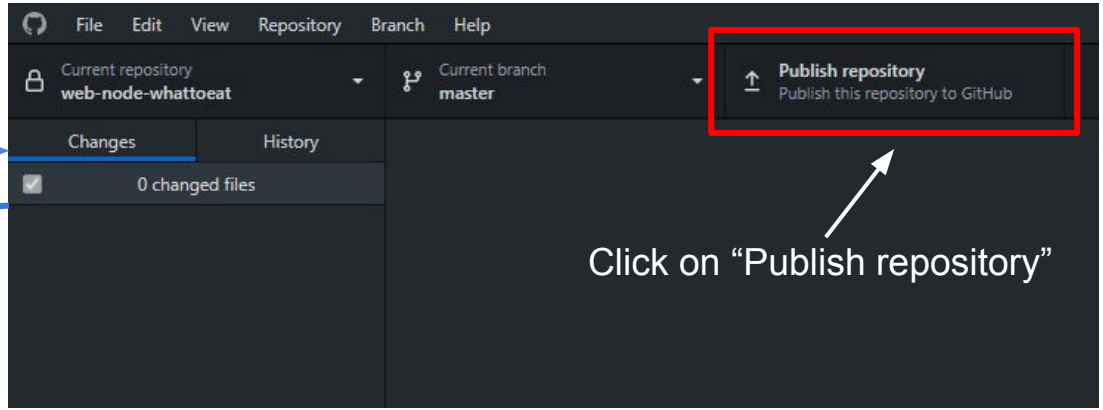


Add all these files to Git so Git will start watching them for changes. If you don't do this, Git will not care about your files

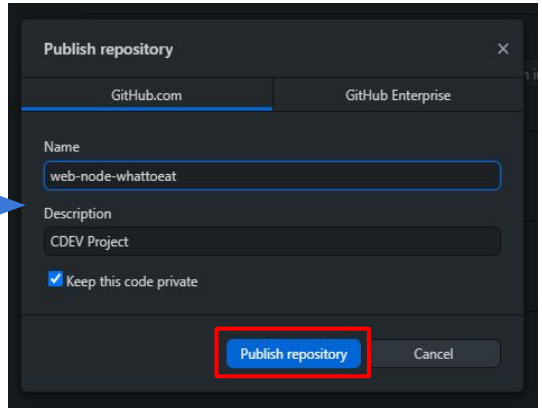
Creating the repository on GitHub



Make sure GitHub Desktop is aware this project exists



Click on “Publish repository”



Set the name, which shouldn’t match the name of any project on your GitHub

Check “Keep this code private” if you don’t want others to copy your code

Recap

Commit	Revert	Push	Clone
Saving the current state of the repository	Going back to a previous state of the repository	Uploading changes to the repository to GitHub	Copying a GitHub repository to local machine

What’s next?

- Many other things you can do

  https://about.gitlab.com/images/press/git-cheat-sheet.pdf

- Integrate Git into this semester’s project!

THANK YOU

Follow Dip IT at @itsigtp on Instagram for updates :)