

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

Version history

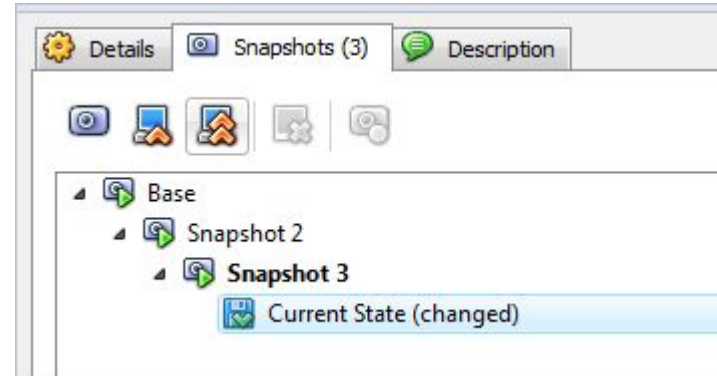
Only show named versions ☐

JUNE 2020

- June 11, 2020, 9:28 PM** 
Current version
● Joey Lim
- ▶ June 11, 2020, 8:41 PM 
● Joey Lim
- ▶ June 11, 2020, 5:45 PM
● Joey Lim

June 11, 2020, 5:29 PM
● Joey Lim

Google Docs Version History



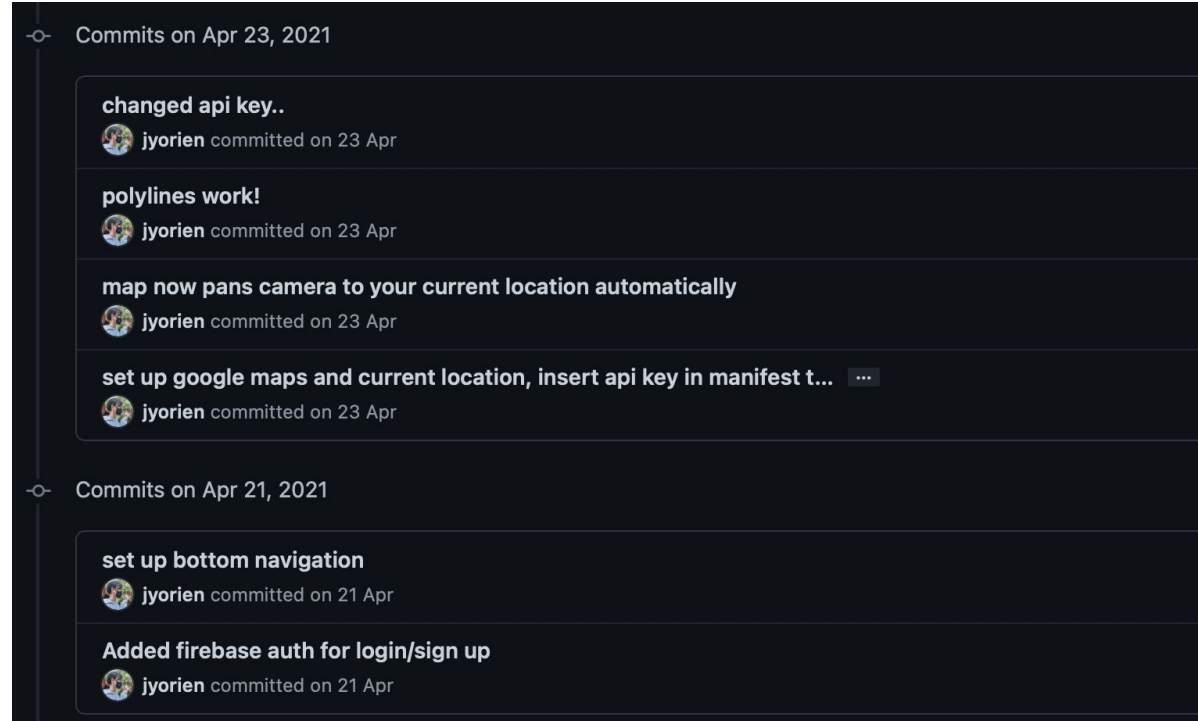
VirtualBox VM Snapshots

Time Travel?

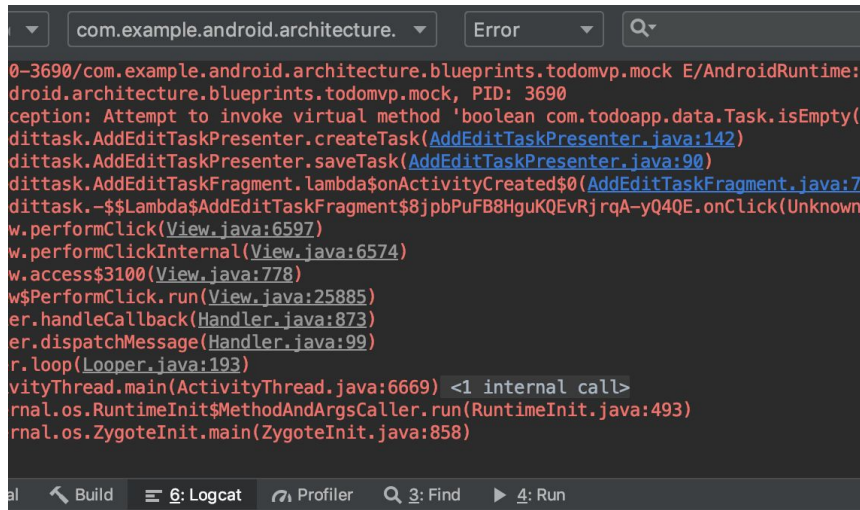


Version Control

- A change tracking system

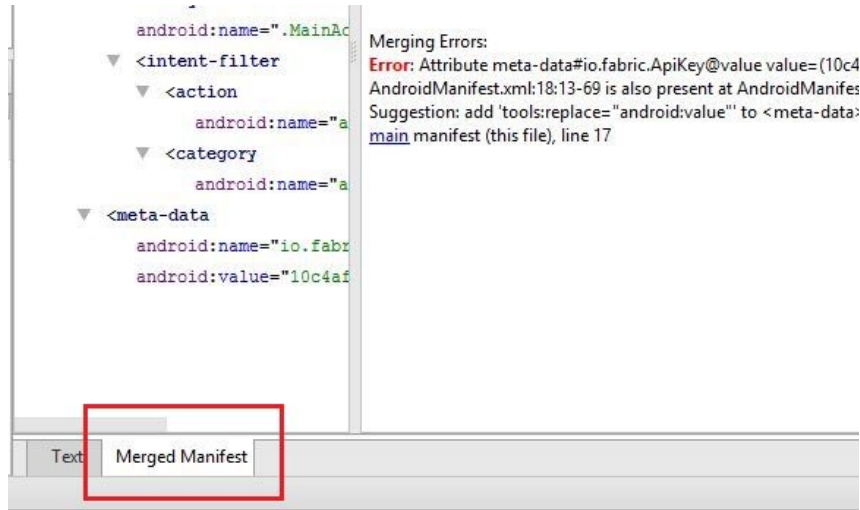


Why use Version Control?



The screenshot shows the Logcat window in Android Studio. The package filter is set to 'com.example.android.architecture'. The error level is set to 'Error'. The log shows a sequence of method calls, including 'createTask', 'saveTask', and 'onClick', followed by a runtime exception: 'E/AndroidRuntime: Attempt to invoke virtual method 'boolean com.todoapp.data.Task.isEmpty()' on a null object reference'. The exception occurs in 'AddEditTaskPresenter.java:142'.

```
0-3690/com.example.android.architecture.blueprints.todomvp.mock E/AndroidRuntime:
droid.architecture.blueprints.todomvp.mock, PID: 3690
ception: Attempt to invoke virtual method 'boolean com.todoapp.data.Task.isEmpty(
dittask.AddEditTaskPresenter.createTask(AddEditTaskPresenter.java:142)
dittask.AddEditTaskPresenter.saveTask(AddEditTaskPresenter.java:90)
dittask.AddEditTaskFragment.lambda$onActivityCreated$0(AddEditTaskFragment.java:7
dittask.-$$Lambda$AddEditTaskFragment$8jpbPuFB8HguKQEvRjrqa-yQ4QE.onClick(Unknown
w.performClick(View.java:6597)
w.performClickInternal(View.java:6574)
w.access$3100(View.java:778)
w$PerformClick.run(View.java:25885)
er.handleCallback(Handler.java:873)
er.dispatchMessage(Handler.java:99)
r.loop(Looper.java:193)
vityThread.main(ActiviyThread.java:6669) <1 internal call>
rnal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:493)
rnal.os.ZygoteInit.main(ZygoteInit.java:858)
```







The screenshot shows the 'Merged Manifest' window in Android Studio. It displays the merged AndroidManifest.xml file. A red box highlights the 'Merged Manifest' tab. To the right of the manifest, there is a 'Merging Errors' section with the following text: 'Error: Attribute meta-data#io.fabric.ApiKey@value value=(10c4 AndroidManifest.xml:18:13-69 is also present at AndroidManifest.xml:18:13-69. Suggestion: add 'tools:replace="android:value"' to <meta-data: main manifest (this file), line 17'.

```
android:name=".MainAc
<intent-filter
  <action
    android:name="a
  <category
    android:name="a
<meta-data
  android:name="io.fabr
  android:value="10c4af
```

Merging Errors:
Error: Attribute meta-data#io.fabric.ApiKey@value value=(10c4
AndroidManifest.xml:18:13-69 is also present at AndroidManifest
Suggestion: add 'tools:replace="android:value"' to <meta-data:
main manifest (this file), line 17

Why use Version Control?

>  backup1	Today at 10:35 PM	-- Folder
>  backup2	Today at 10:35 PM	-- Folder
>  backup3	Today at 10:35 PM	-- Folder
>  currentapp	Today at 10:35 PM	-- Folder

GitHub Student Developer Pack

<https://education.github.com/pack>



About Canva

With Canva, anyone can create professional looking graphics and designs. Featuring thousands of templates and an easy to use editor.

Benefit

Free 12 month subscription of Canva's Pro tier.

Virtual Events

Design



About Namecheap

Affordable registration, hosting, and domain management

Benefit

1 year domain name registration on the .me TLD.

Virtual Events

Benefit

1 SSL certificate free for 1 year.

Virtual Events

Domains



About JetBrains

Professional desktop IDEs: IntelliJ IDEA, PyCharm, and more.

Benefit

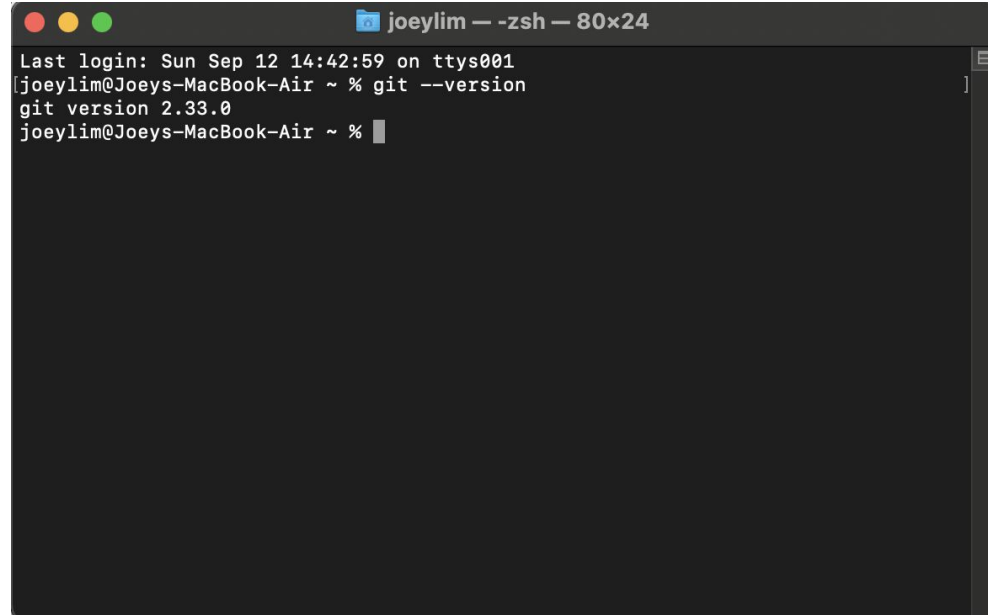
A free subscription for students, to be renewed annually.

Developer tools



Download Git

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

A terminal window titled 'joeylim — -zsh — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal shows the following text:

```
Last login: Sun Sep 12 14:42:59 on ttys001
joeylim@Joeys-MacBook-Air ~ % git --version
git version 2.33.0
joeylim@Joeys-MacBook-Air ~ %
```

Download Git

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

Git book
Pro Git and
available to read
read tree
able on

--distributed-is-the-new-centralized

Search entire site...

Downloads



Older releases are available and the [Git source repository](#) is on GitHub.



GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

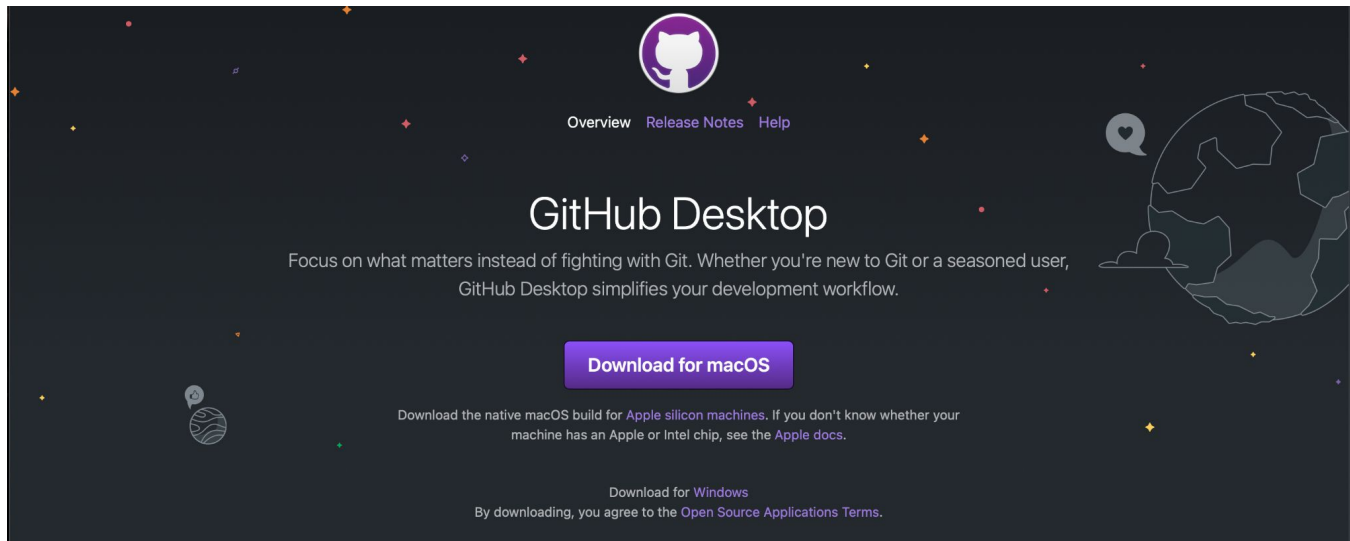
Logos

Various Git logos in PNG (bitmap) and (vector) formats are available for use in web and print projects.

[View Logos →](#)

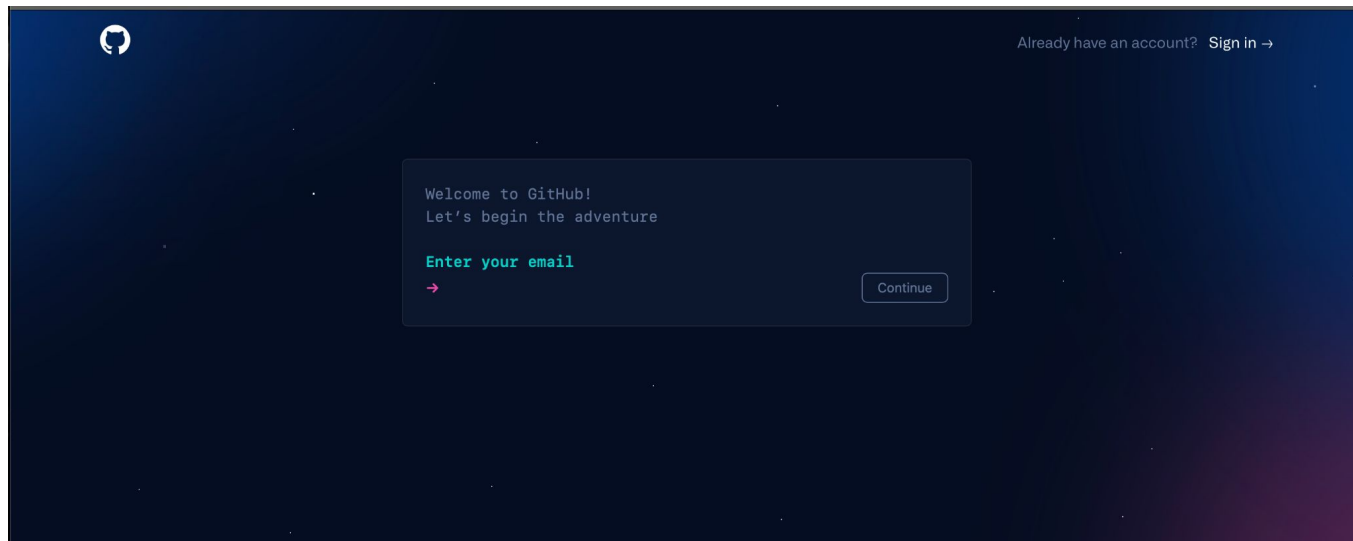
Download GitHub Desktop

- 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



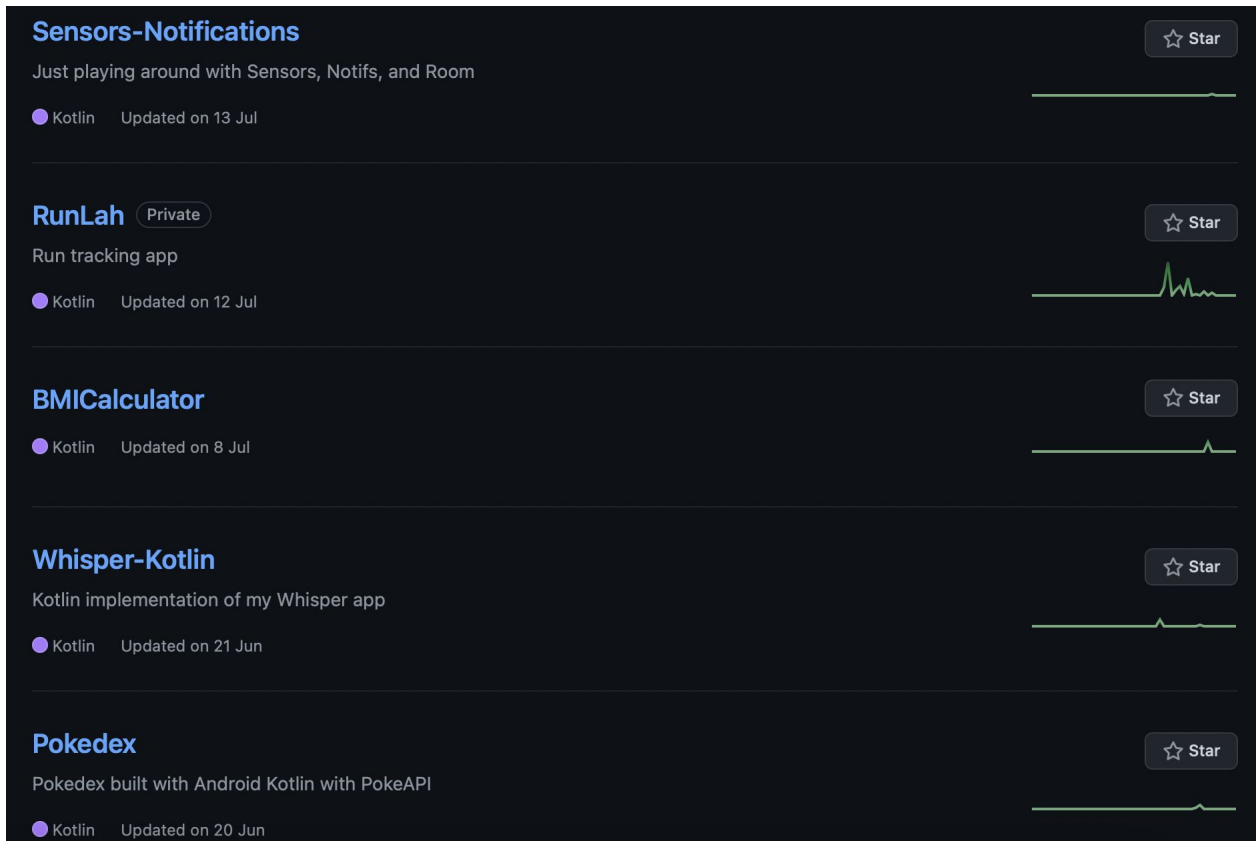
5 Minutes Break

- 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!

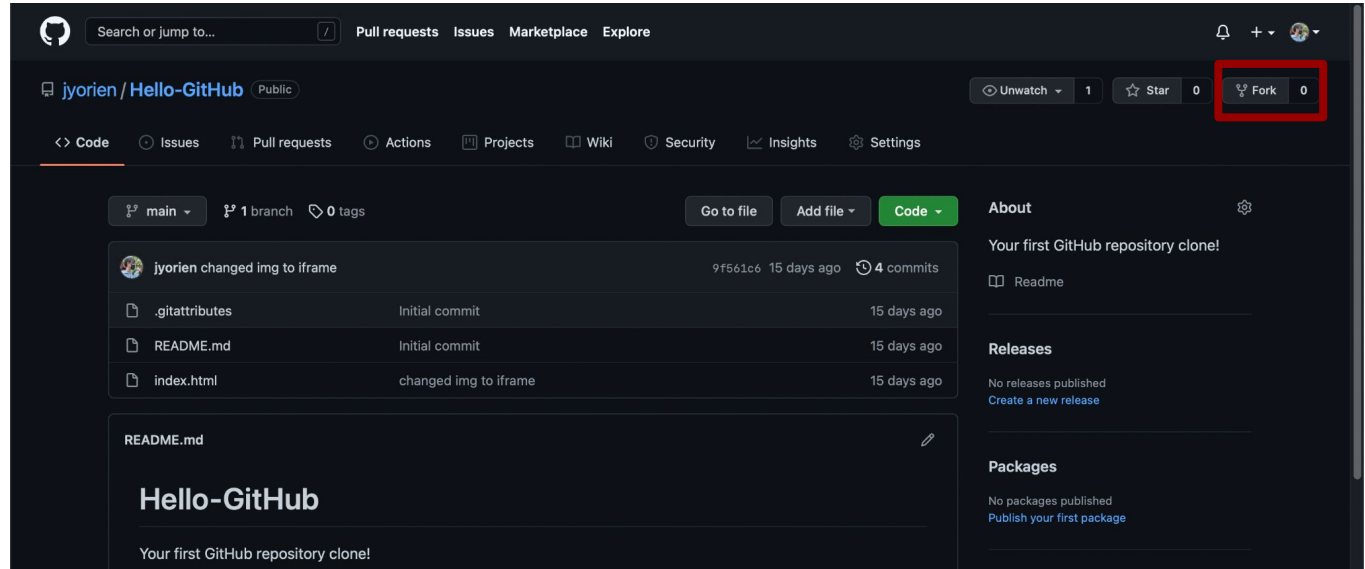
Repositories

- A repository or 'repo' is your project



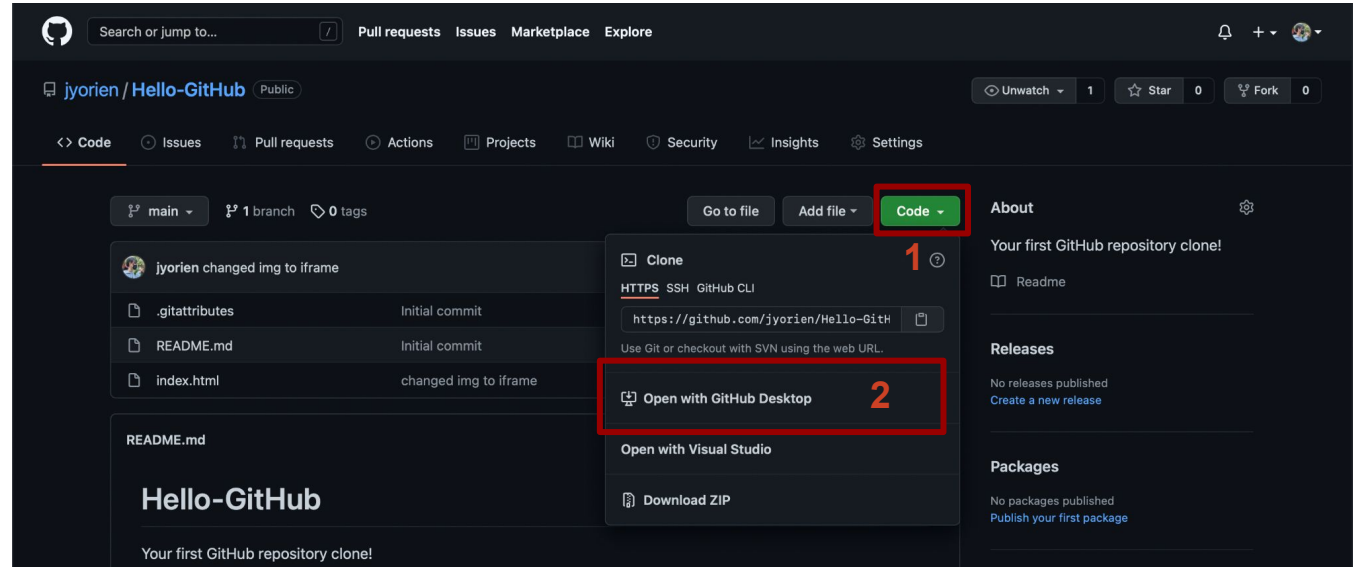
Forking

- 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'



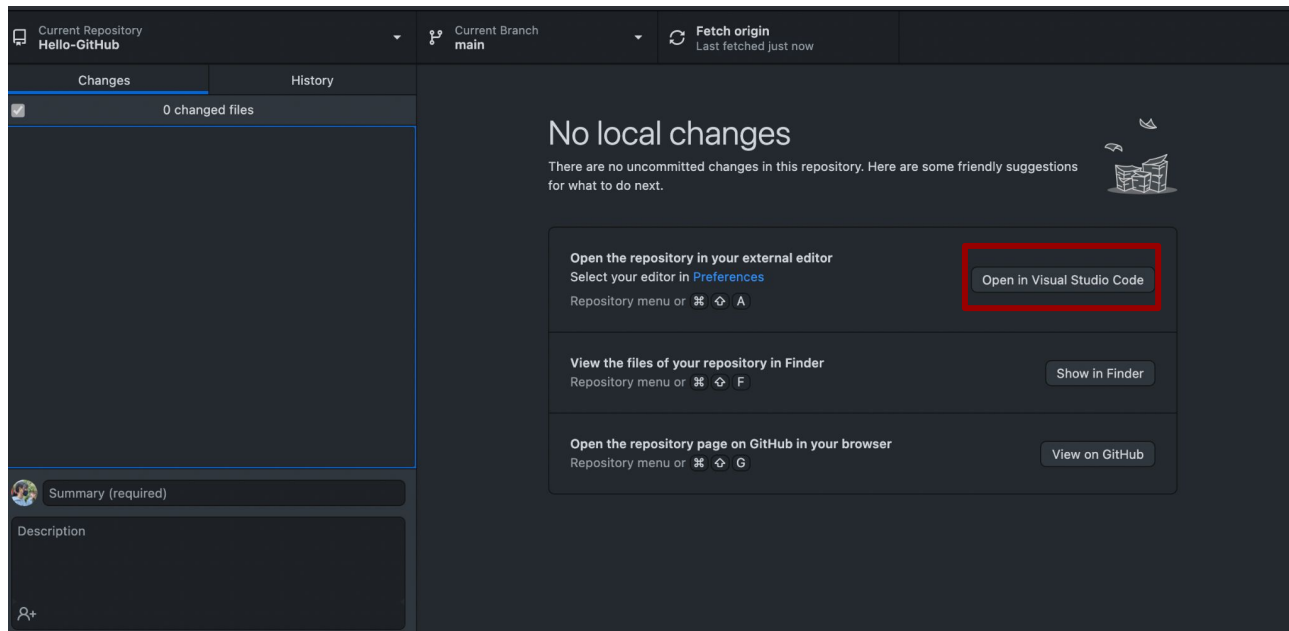
Clone

- 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.

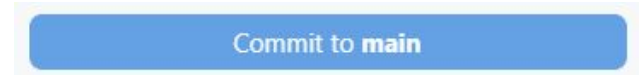


Commit & Revert

“Commit” and “Revert”

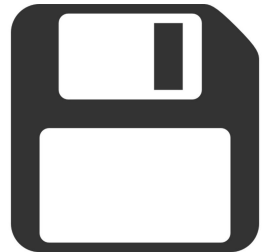
What is “Commit” ?

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.



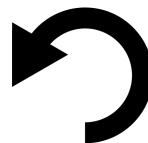
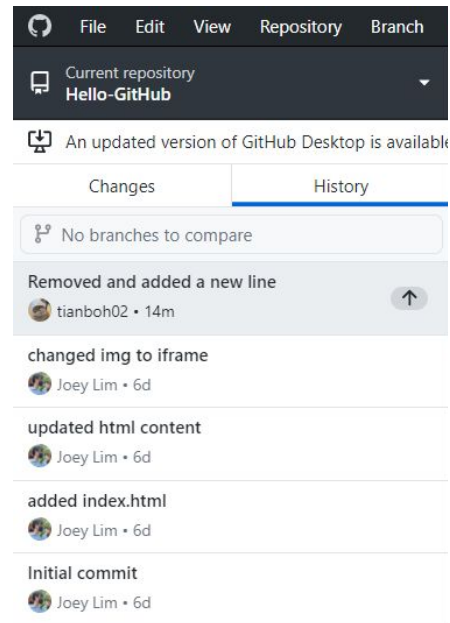
“Commit” and “Revert”

What is “Revert” ?

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

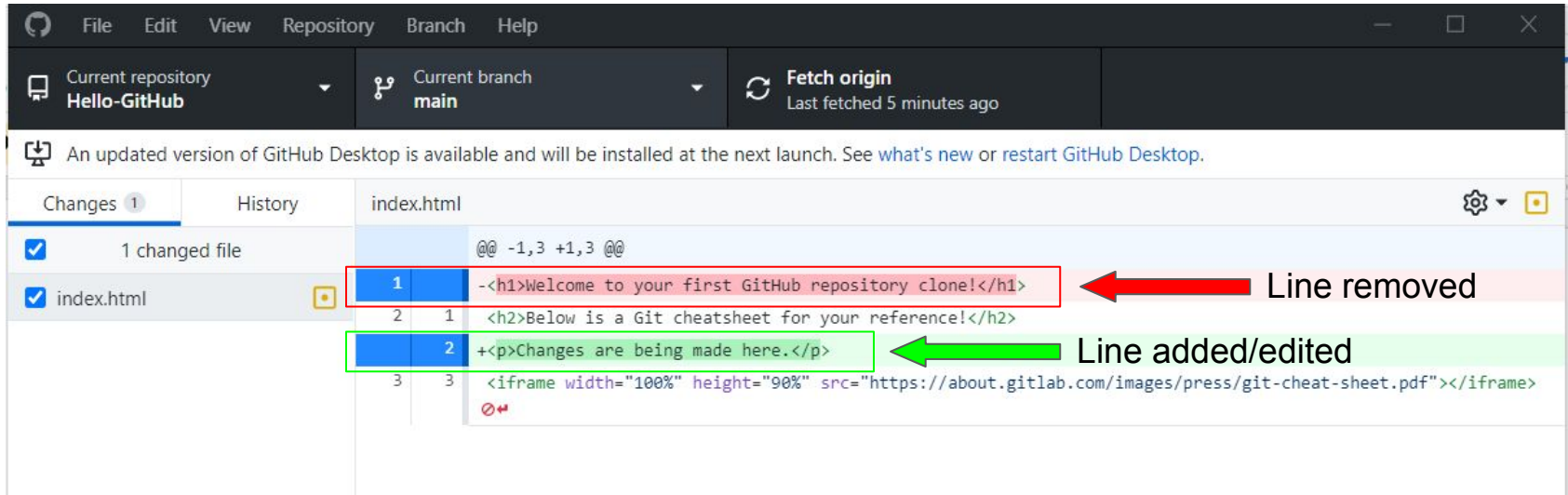
Why “Revert” then?

- 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.



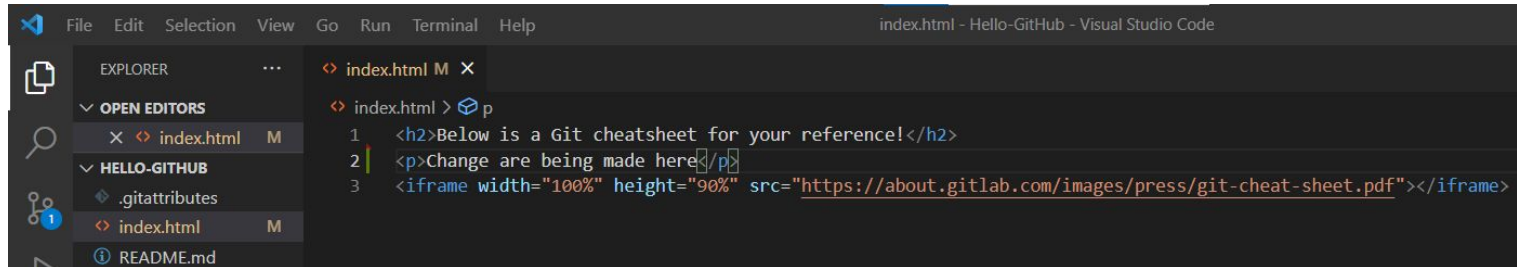
How to “commit”?



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer shows a project named 'HELLO-GITHUB' with files '.gitattributes', 'index.html', and 'README.md'. The 'index.html' file is open in the editor. The code in the editor is as follows:

```
index.html > h1
1 <h1>Welcome to your first GitHub repository clone!</h1>
2 <h2>Below is a Git cheatsheet for your reference!</h2>
3 <iframe width="100%" height="90%" src="https://about.gitlab.com/images/press/git-cheat-sheet.pdf"></iframe>
```

Before changes



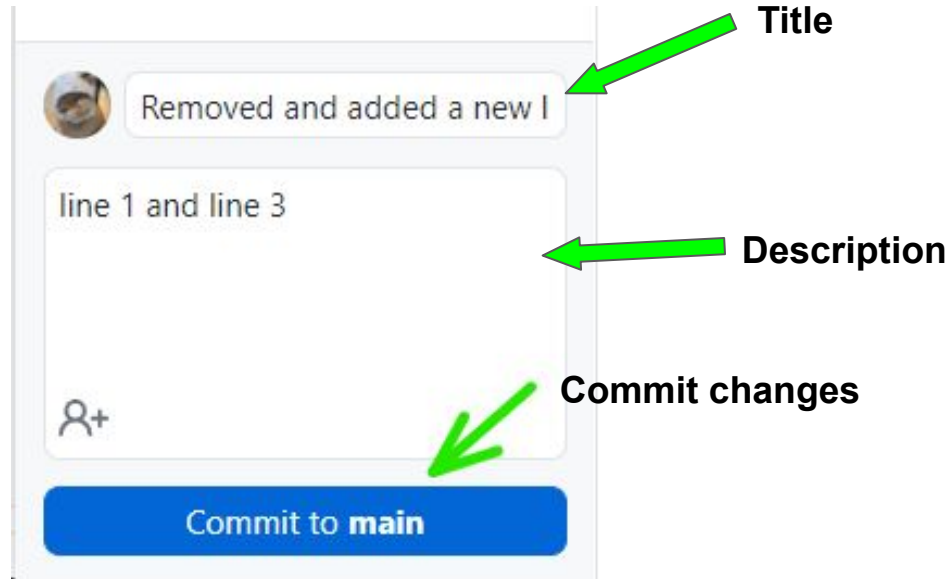
The screenshot shows the Visual Studio Code interface after changes have been made to 'index.html'. The Explorer sidebar shows 'index.html' with a green 'M' (modified) icon. The code in the editor is as follows:

```
index.html > p
1 <h2>Below is a Git cheatsheet for your reference!</h2>
2 <p>Change are being made here</p>
3 <iframe width="100%" height="90%" src="https://about.gitlab.com/images/press/git-cheat-sheet.pdf"></iframe>
```

After changes

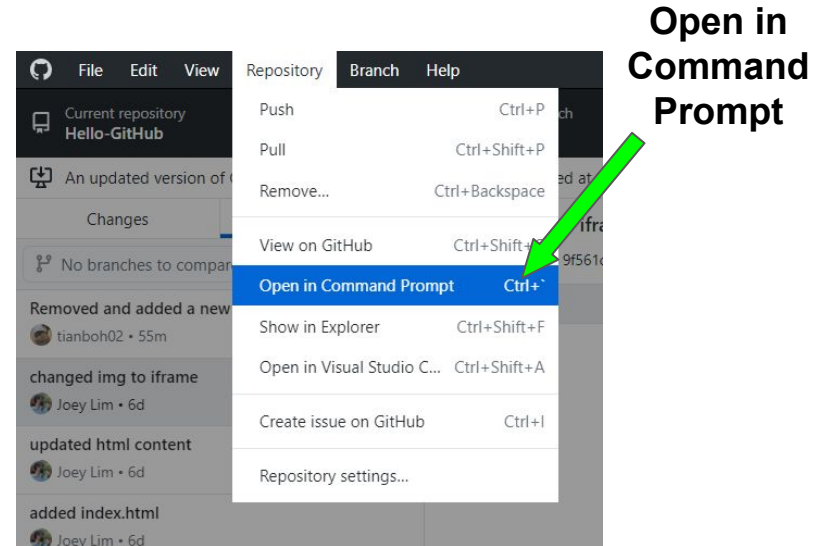
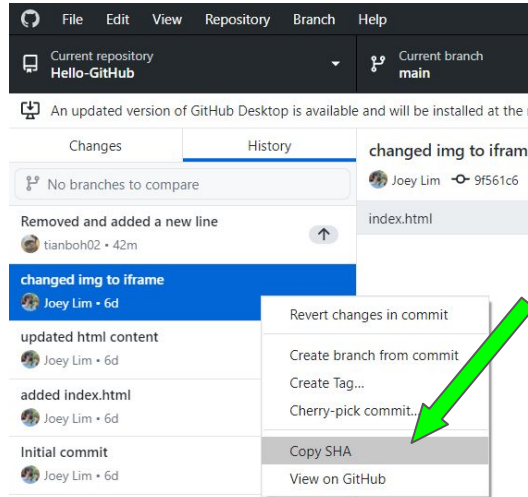
How to “commit”?

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”

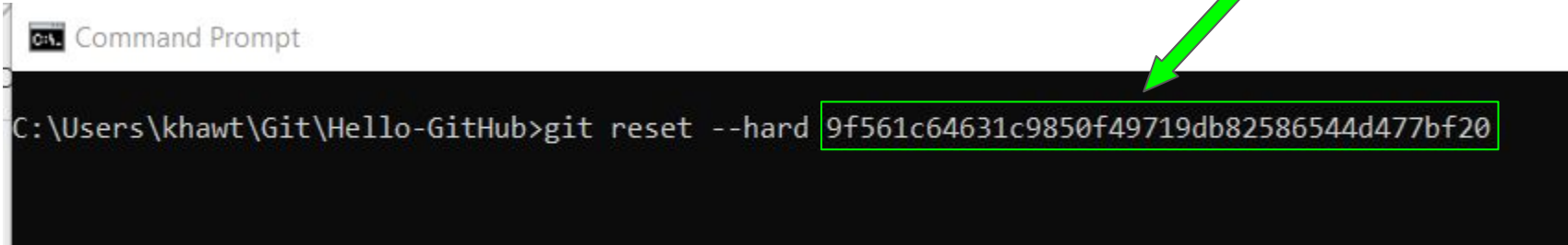


How to “revert”?

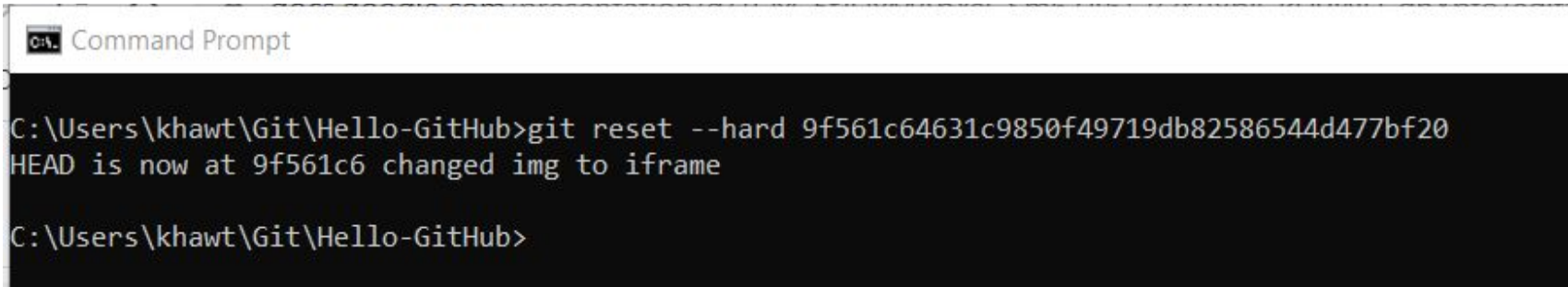
3. In the command prompt, type the following and enter:

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

SHA-KEY

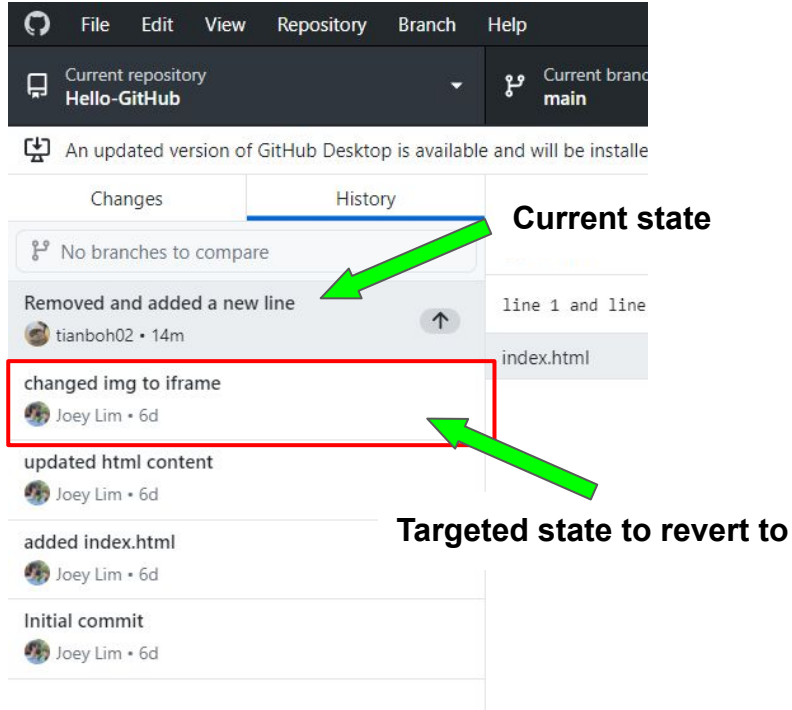


```
C:\Users\khawt\Git\Hello-GitHub>git reset --hard 9f561c64631c9850f49719db82586544d477bf20
```

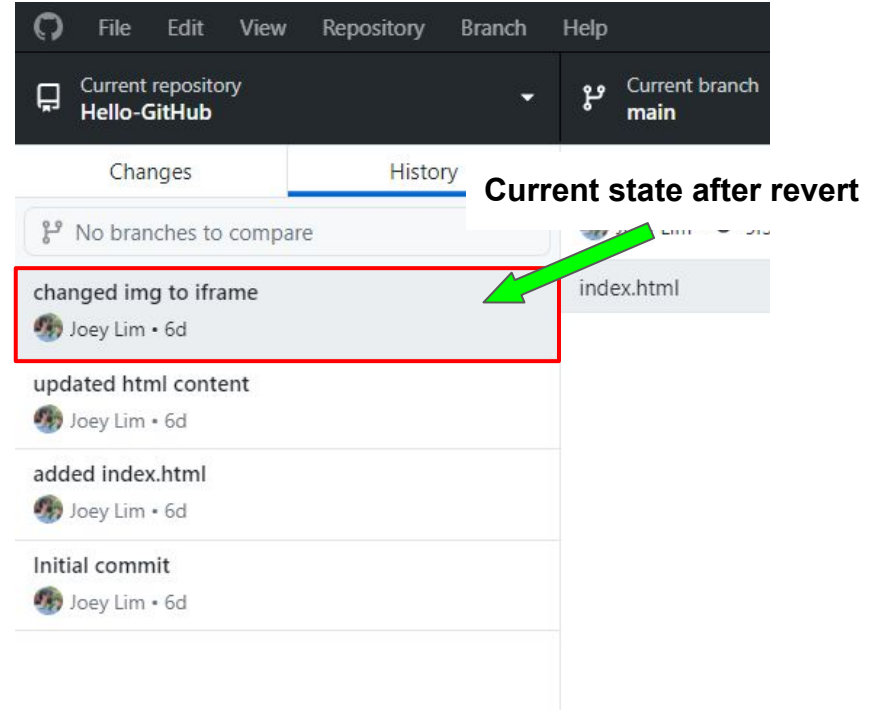


```
C:\Users\khawt\Git\Hello-GitHub>git reset --hard 9f561c64631c9850f49719db82586544d477bf20
HEAD is now at 9f561c6 changed img to iframe
C:\Users\khawt\Git\Hello-GitHub>
```

How to “revert”?



Before Revert



After Revert

Git and GitHub



git

Git



GitHub

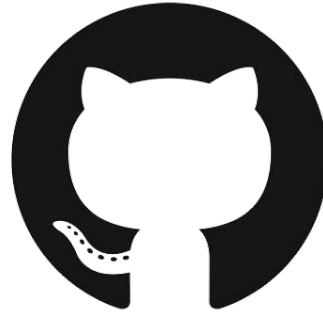
What really is Git

- 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

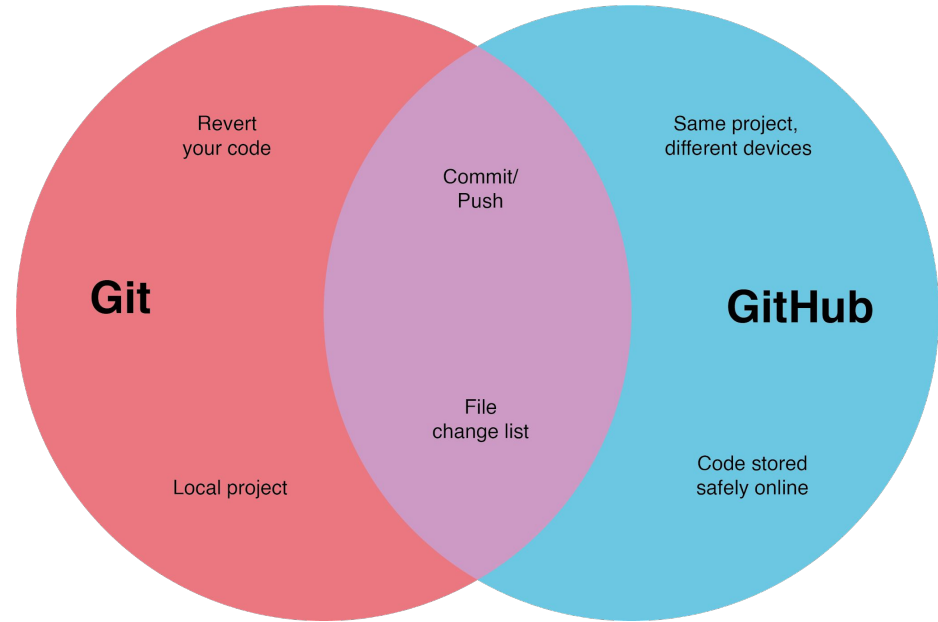
- GitHub is a free online service for storing your repositories
- Keeps your code safely stored online preventing loss of code
- Is a bit similar to Google Drive or OneDrive



GitHub

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



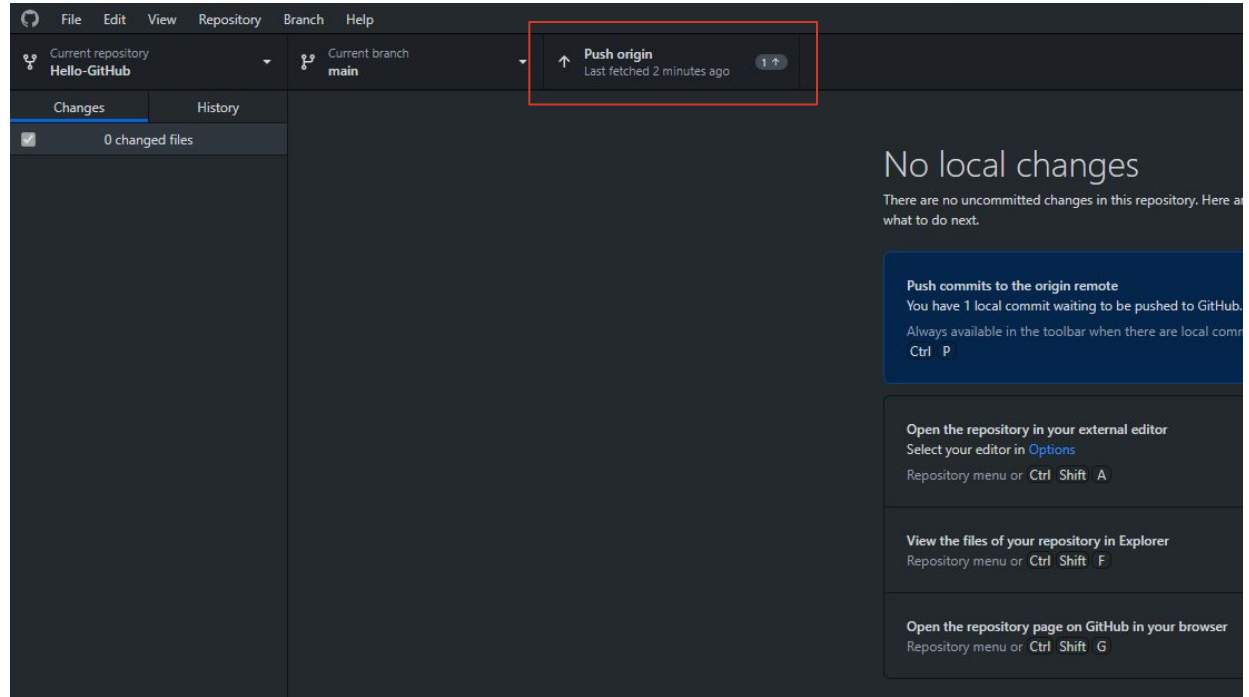
With GitHub, this won't be a disaster



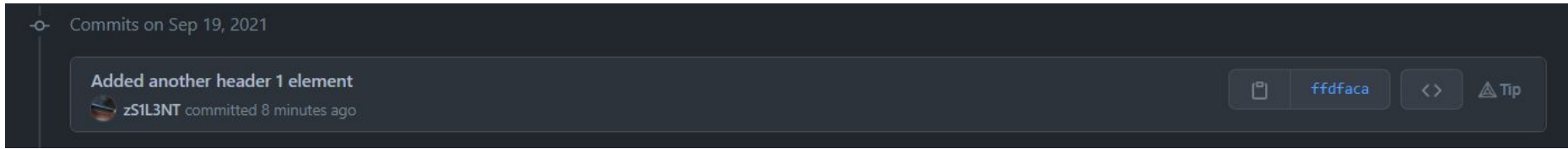
What does “Push to GitHub” mean?

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

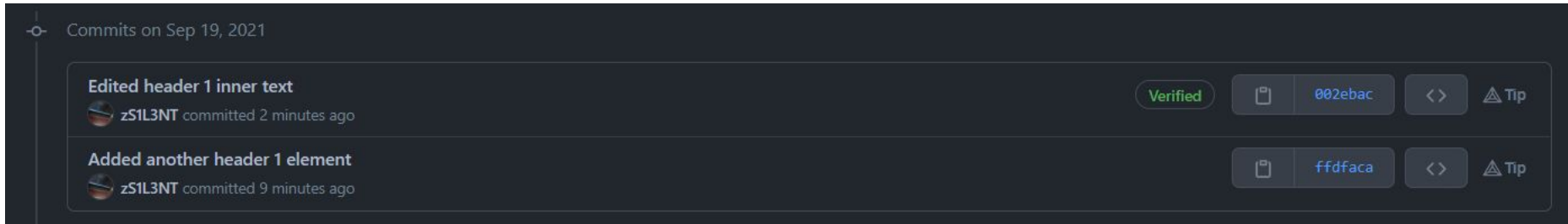
Once you make a commit to git, go to GitHub Desktop and click “Push origin”



Before and after the “Push” on GitHub



Before push

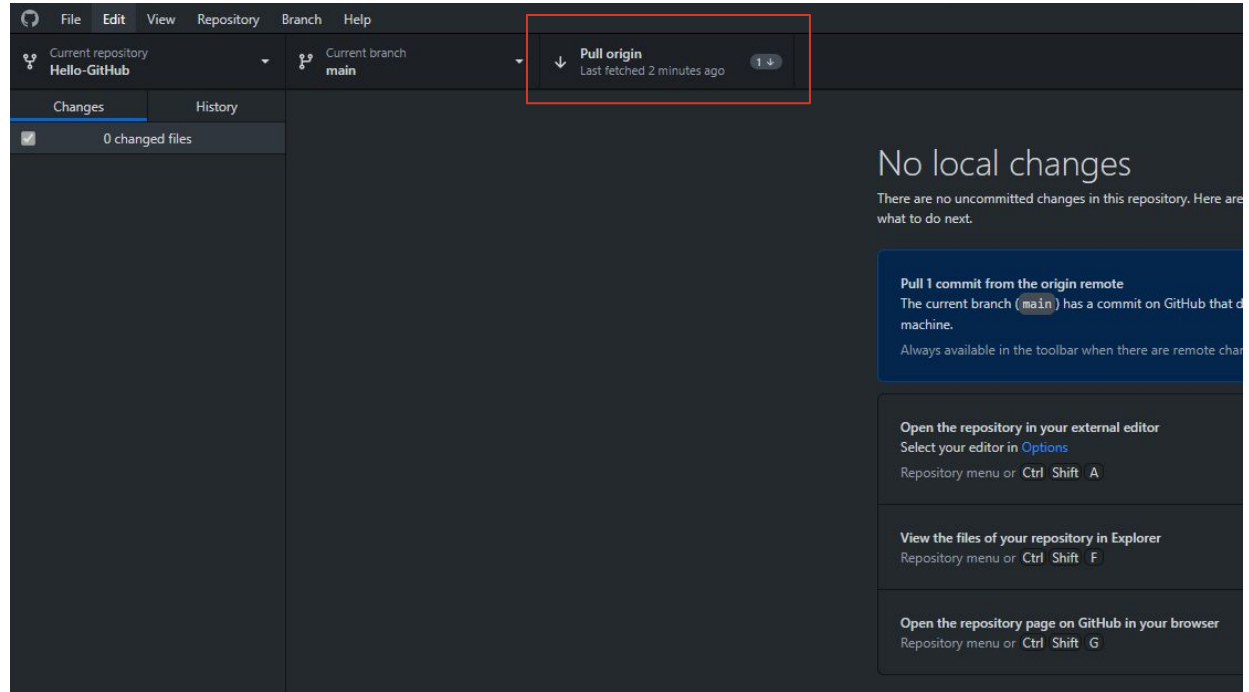


After push

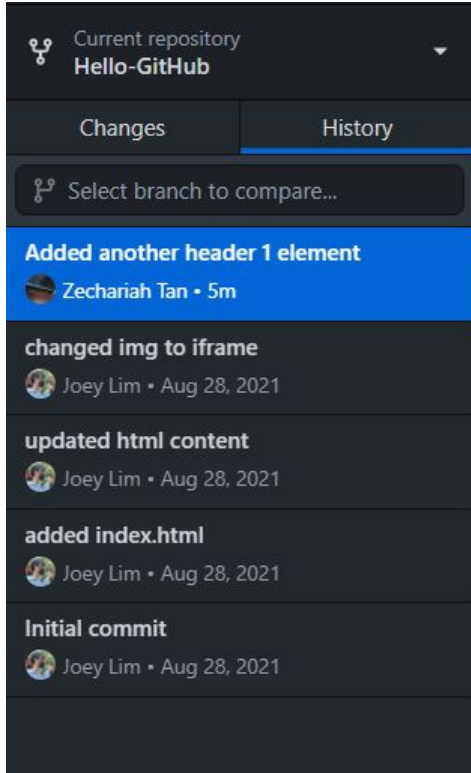
What does “Pull from GitHub” mean?

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

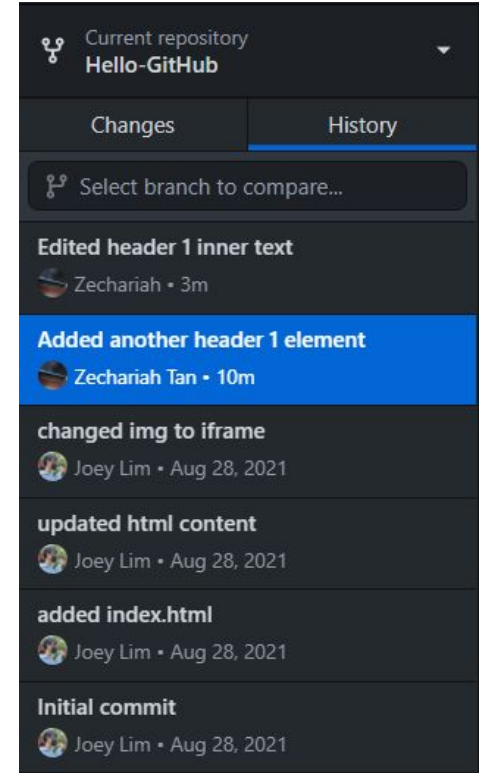
If you pushed code from your laptop and want the code on your desktop, go to GitHub Desktop and click “Pull origin” on your desktop



Before and after the “Pull” on GitHub Desktop

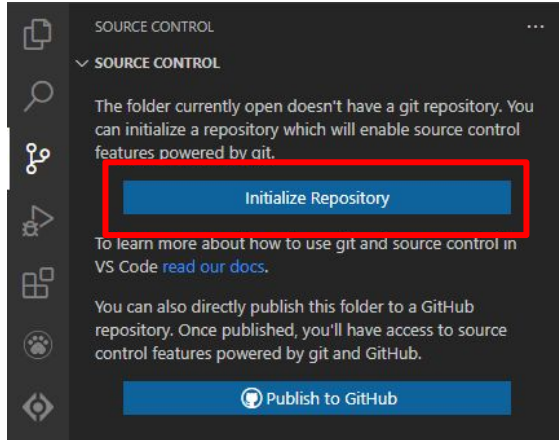


Before pull

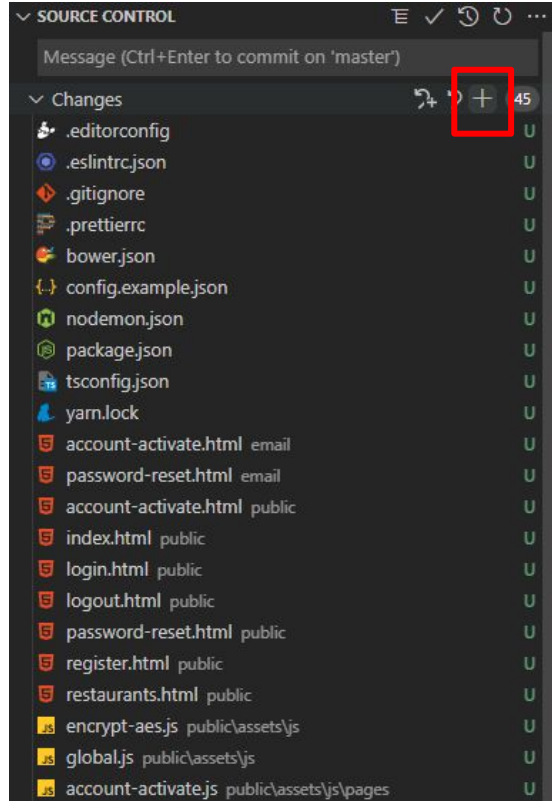


After pull

Adding Git to an existing project

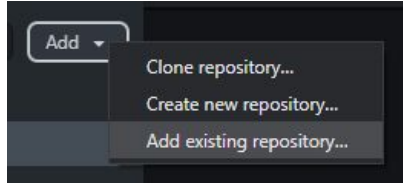


Initialise Git in your project

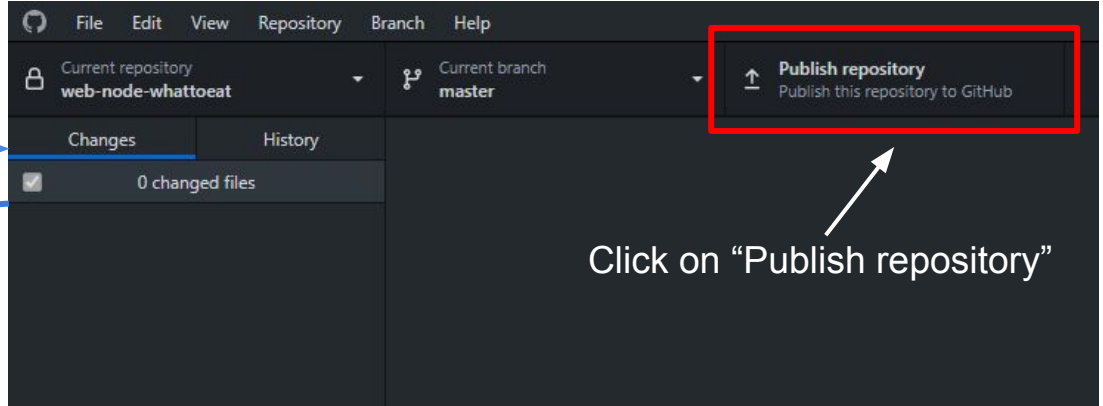


Add all these files to Git so Git will start watching them for changes. If you don't do this, Git will not care about your files

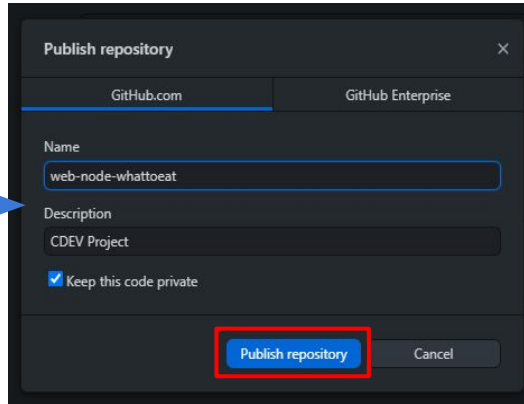
Creating the repository on GitHub



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 :)

