Memorial Sloan Kettering Cancer Center

Git/GitHub Training Part 1

Introduction to Git & Individual Workflows

March 7, 2022
Jessica Lavery, Epi-Bio
Daniel D. Sjoberg, Epi-Bio
Caroline Kostrzewa, Epi-Bio
Karissa Whiting, Epi-Bio
Shannon Pileggi, PCCTC
Karolyn Ismay, Strategy & Innovation

Agenda

Why git?

Concepts & Vocabulary

- Git
- GitHub
  - GitHub & GitHub Enterprise
- Repositories (“Repos”)
  - Creating a repository
  - Remote/local versions of repositories
  - Committing
  - Pushing Changes
  - Diffs

Using git for MSK projects

- PHI requirements
- Additional tools for R users

Practice!

Goals

- Get a basic understanding of why git is useful
- Be able to:
  - Create repositories
  - Commit changes
  - Push and pull changes
  - See differences between commits

Prior to Today

- Signed up for a GitHub Enterprise account and set it up for use with your MSK workstation
- Installed GitHub Desktop
  - Installation link
  - Documentation

Git: What and Why

What is git?

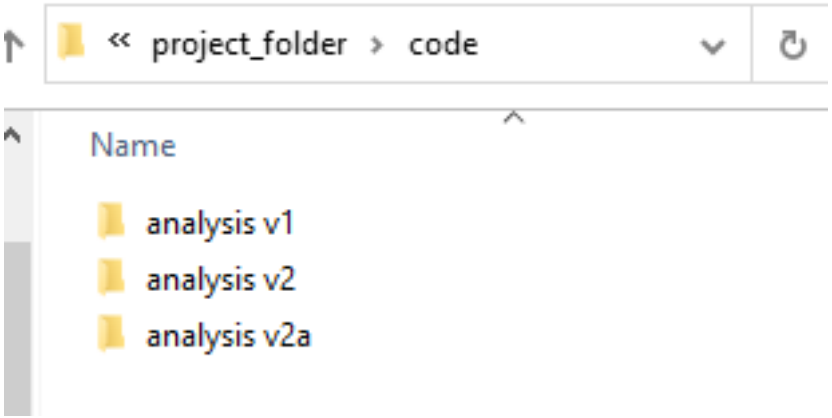
- git is a free, open-source **version control system** that can be used to track changes across code files

“In every project you have at least one other collaborator; future-you. You don’t want future-you to curse past-you.”
- Hadley Wickham

Why git?

- Version control
  - Formalizes the update process
  - Understand changes
  - Can always go back
    - Investigator asks for one version of the analysis, you give it to them, then they ask for a new version, only to revert back to the old version
- Collaboration & code review (training part II)

Without git

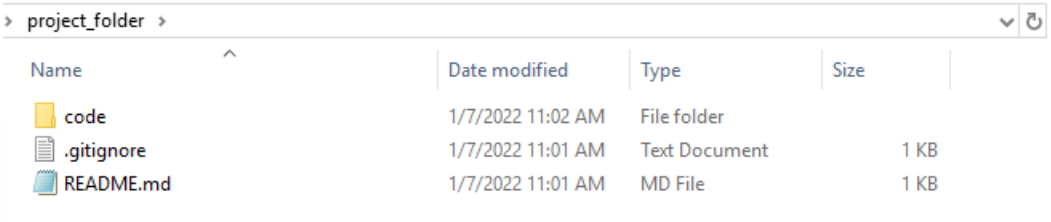


When the paper comes back for revisions:

Which is the primary analysis? What were the changes across versions? Why v1 and then v2/v2a?

With git

- Your files still exist on your C:/G:/H:/One drive in whichever file structure you prefer



- Now have ability to:
  - Track changes that are made to a file
  - Revert to previous versions
  - Collaborate on code with others (part II of course)

With git: Each time you update your code script

code\analysis.Rmd

```
@@ -3,6 +3,8 @@ title: "Example R Script"
 author: "Jessica Lavery"
 date: "1/7/2022"
 output: html_document
+editor_options:
+  chunk_output_type: console
 ---

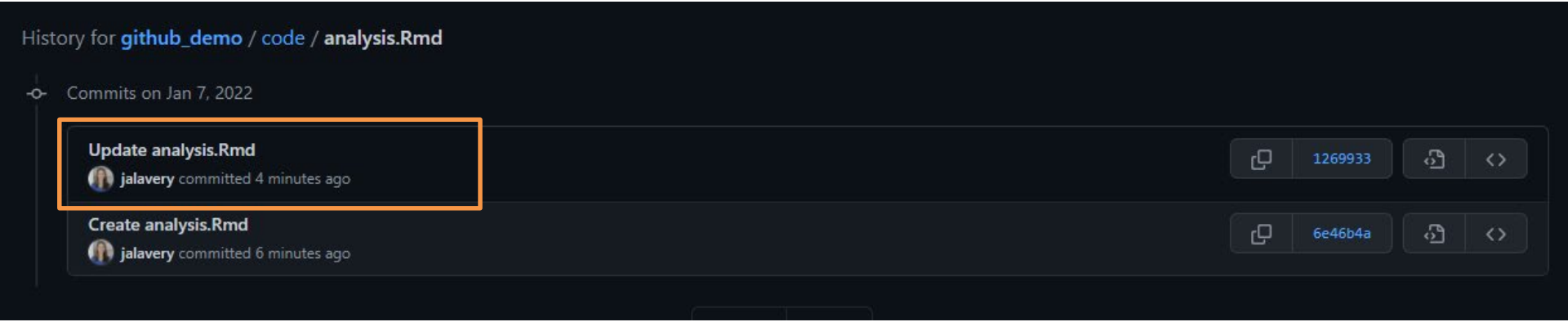
 ```{r setup, include=FALSE}
@@ -16,7 +18,7 @@ This is an R Markdown document. Markdown is a simple formatting syntax for autho
 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

 ```{r cars}
-summary(cars)
+summary(mtcars)
 ```

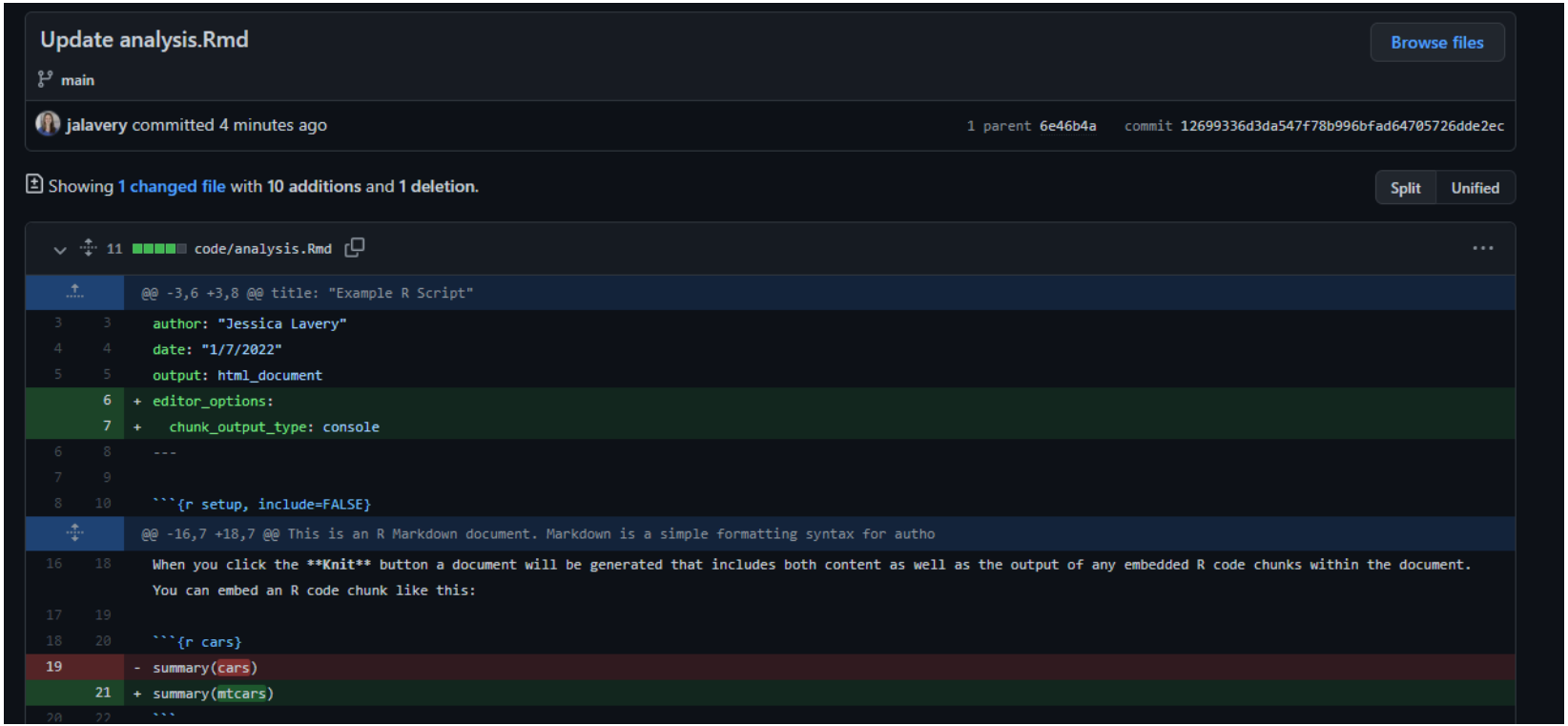
 ## Including Plots
@@ -28,3 +30,10 @@ plot(pressure)
 ```

 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
+
+## Part 2 of analysis
+
+```{r}
+summary(gtsummary::trial)
+```
+
```

With git: History of what changed & when



With git: History of what changed & when



Questions on git?

At this stage, you should:
Understand what git is, but still have no idea how to use it

What is GitHub?

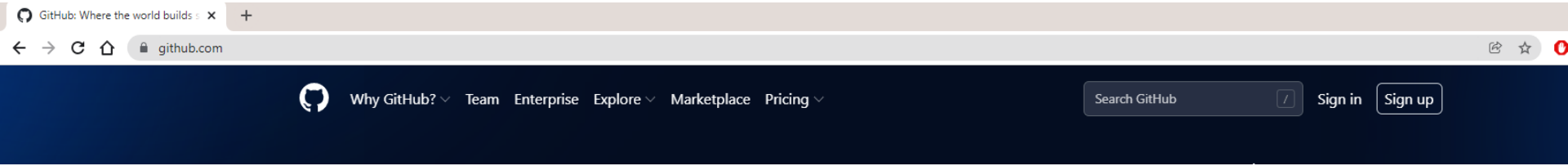
- GitHub is a hosting service for git-based projects
  - **Git** = version control system
  - **GitHub** = tool for working with that version control system that doesn't involve the command line
- Think of GitHub like Google Drive or Dropbox/Box
  - Files stored locally on your computer
  - Sync with Drive/Dropbox/Box on the cloud and can be accessed via those websites and shared with others

GitHub

- www.github.com
- Public website
- Individual repositories can be made public or private (only you or invited collaborators can see repo)

Registering for GitHub

1. Navigate to https://github.com/
2. Click ‘Sign up’ on the top right corner



3. Create your account using your personal email address (not your MSK email)

Installing GitHub Desktop

1. Navigate to GitHub Desktop website
2. Select Installing and configuring GitHub Desktop
3. Follow installation instructions
4. Confirm installation was successful by opening GitHub Desktop

Questions on GitHub/GitHub Enterprise?

At this stage you should:

- Understand how git is related to GitHub
- Still not know how to use either for version control

Repositories (“Repos”)

What's a repository?

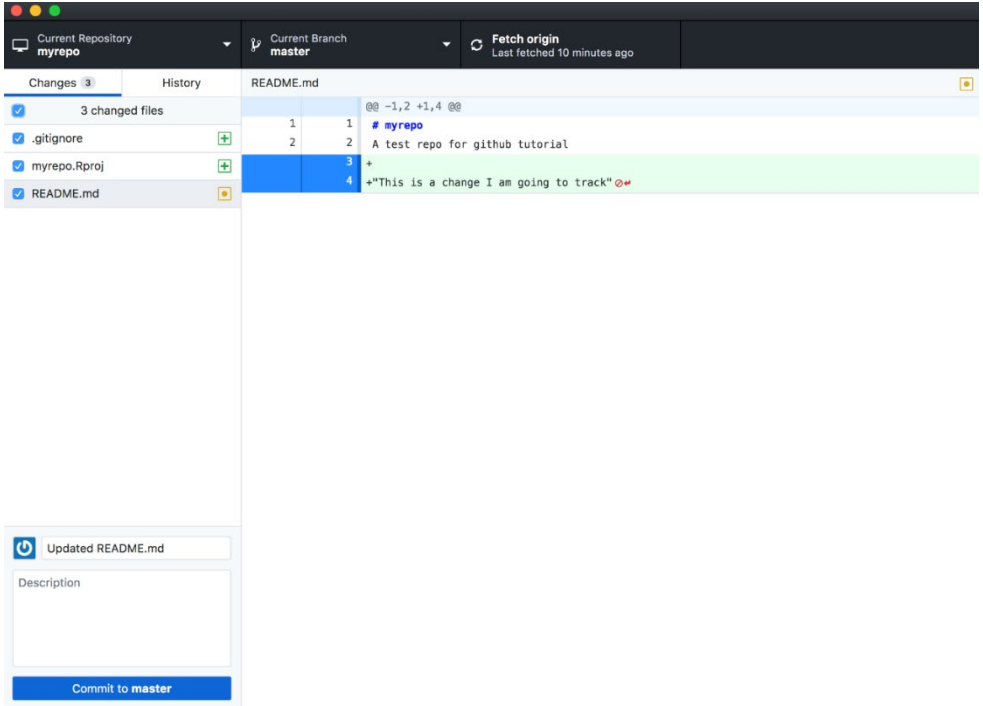
- Think of a **repository** as a project folder
- Just like you have separate folders for each of your projects now, you can have separate repositories for each of your projects

Repo Locations

- **Local repository** - Version of your project that lives on your local computer. *This is the version you make edits to.*
- **Remote repository** - Version that lives remotely on GitHub.
  - GitHub offers a suite of features for managing, organizing and searching your project code.

Managing Repositories: GitHub Desktop App

- Provides an interface for connecting the local and remote version of the repository



Types of Repositories

Private Repositories

- Only visible to authorized users
- Not visible unless you are invited to contribute

Public Repositories

- Anyone can see this repository, even if they do not have a GitHub account

Types of Repositories

Private Repositories

- Pro: your work may not be ready to share, only you and others you invite can see it

Public Repositories

- Pro: Others can contribute to your work
- Con: Others may download/copy your code before it is ready to be shared

Creating a Repository

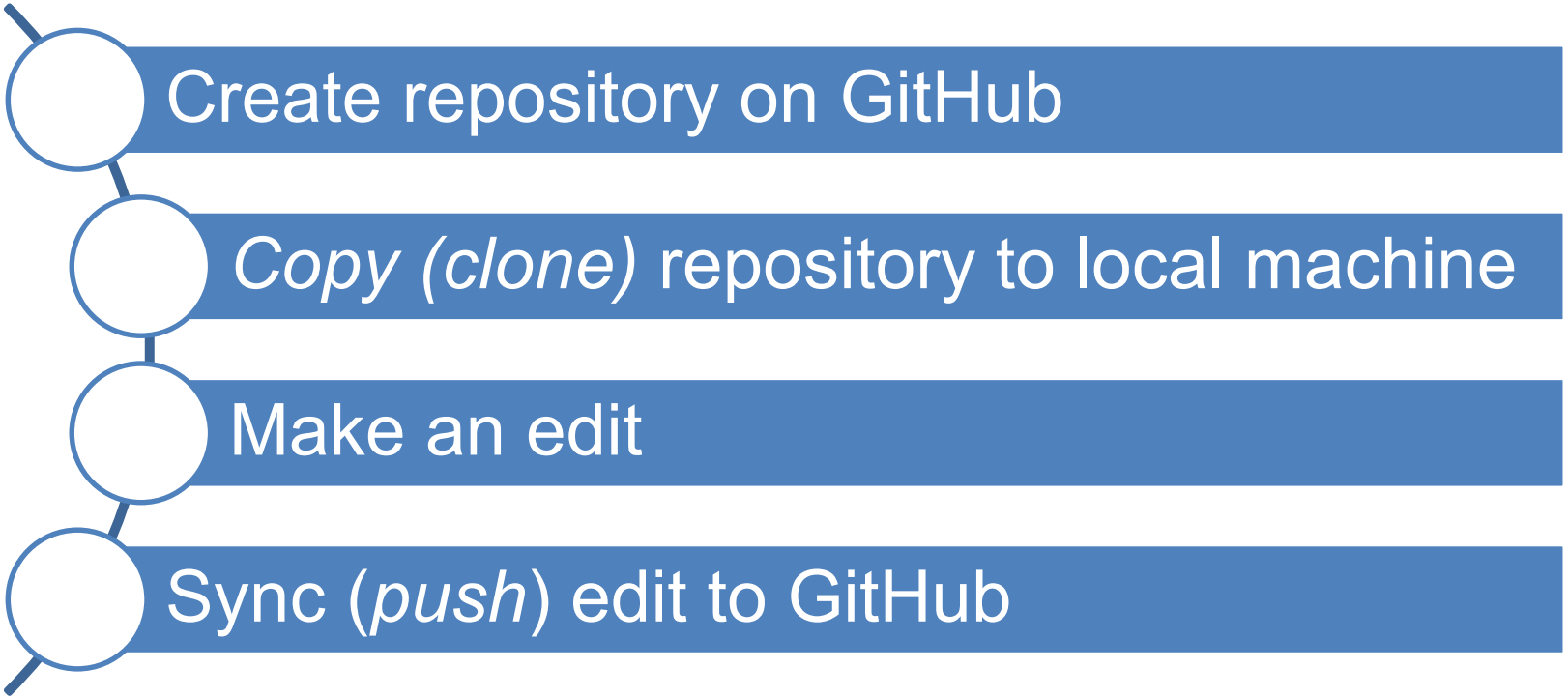
- There are multiple ways to create a repository
  - Create locally your machine, then tell GitHub about it
  - Create on GitHub, then copy it to your local machine*

GitHub Training Part II

- GitHub is useful individually, but is even more useful when collaborating with others
- GitHub training part II will cover working collaboratively on GitHub including terms like *branch, fork, pull request*

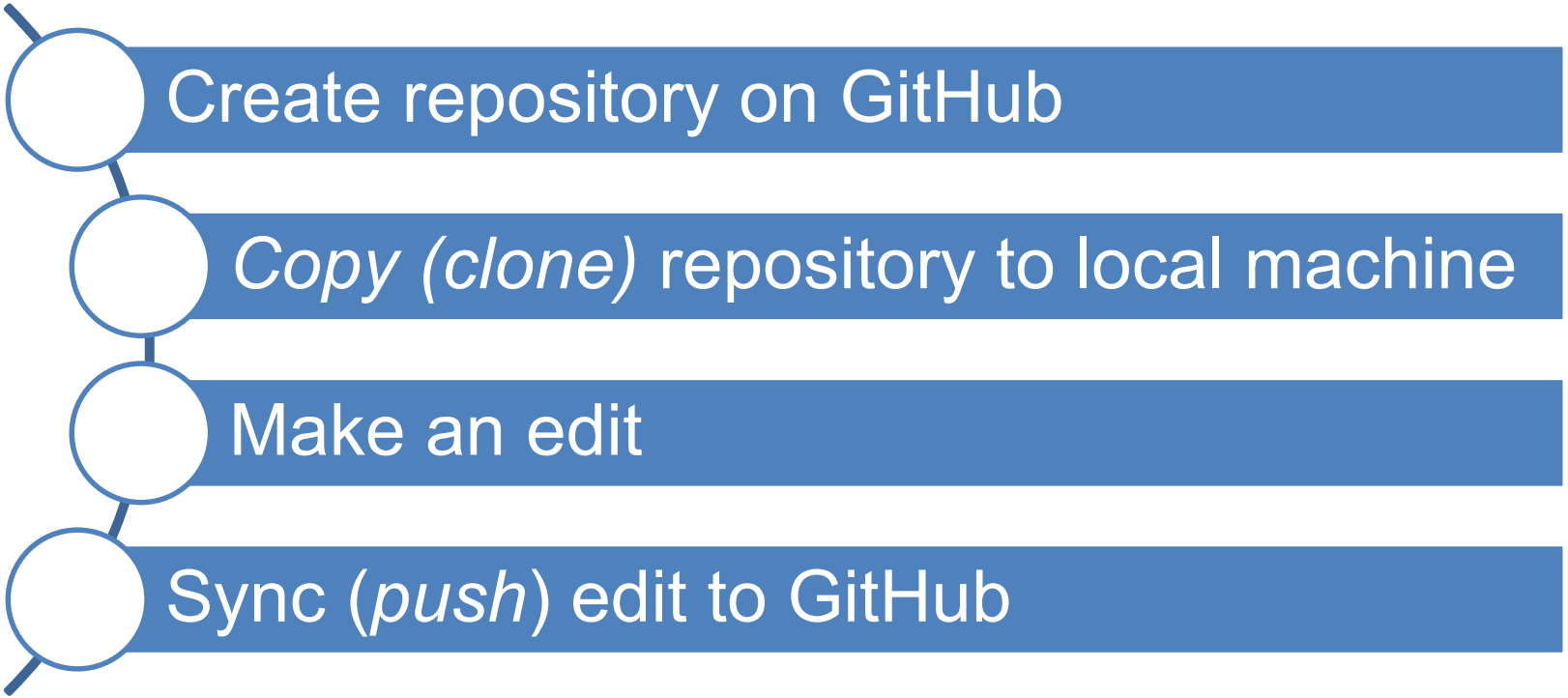
Questions?

Exercise: Create a Repository



Breakout Rooms

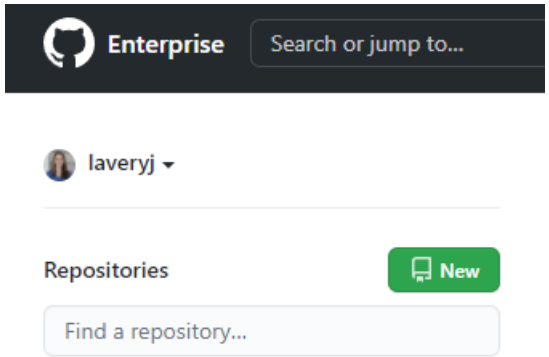
Exercise: Create a Repository



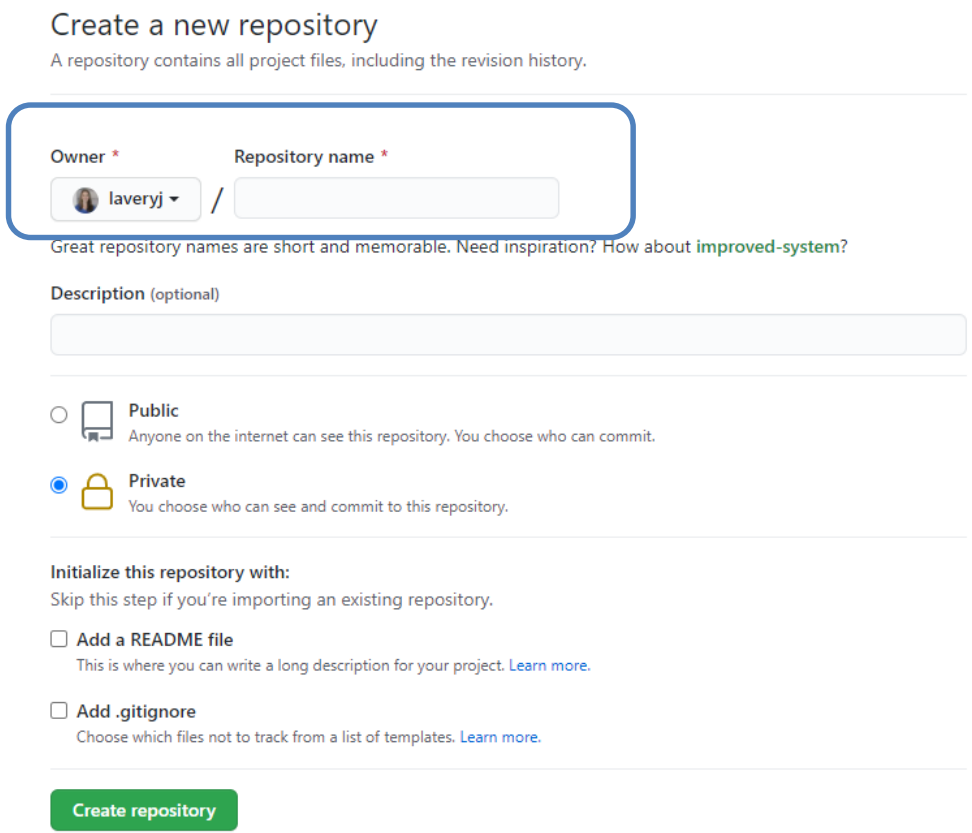
Exercise: Create a Repository

Create a Repository on GitHub

1. Navigate to https://github.mskcc.org/
2. Log in to your account via your MSK credentials
3. On the top left hand side, click 

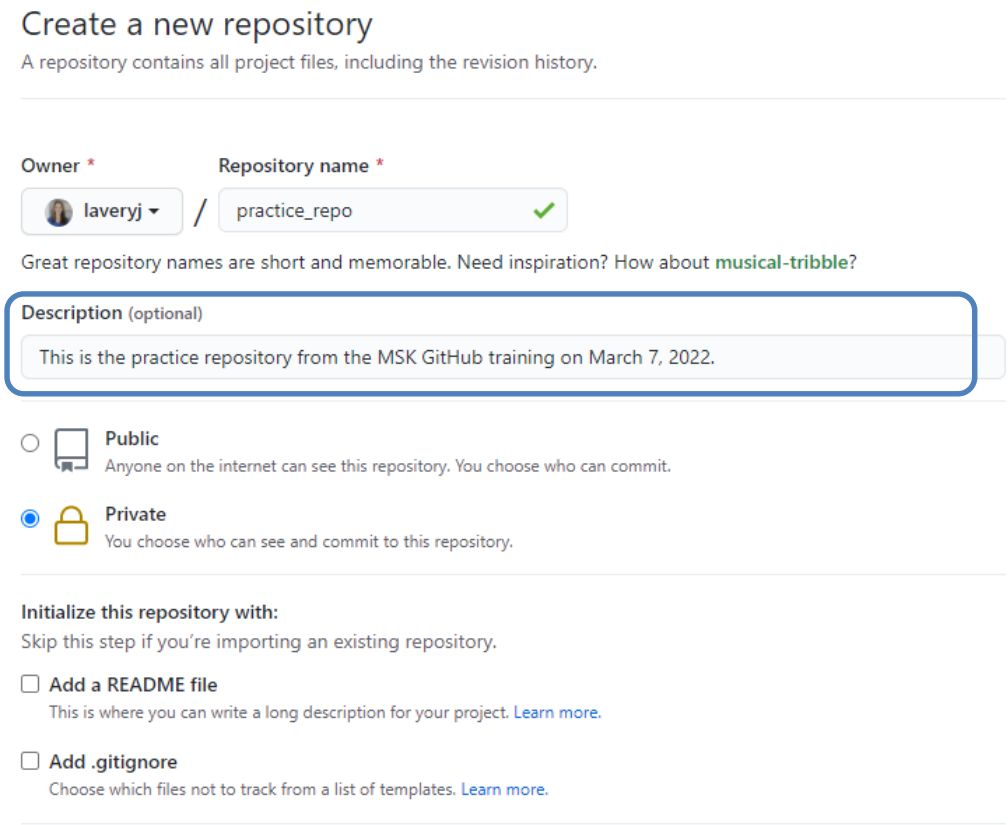


Create a Repository on GitHub



Name your repository something informative. Today we can call this “practice_repo”

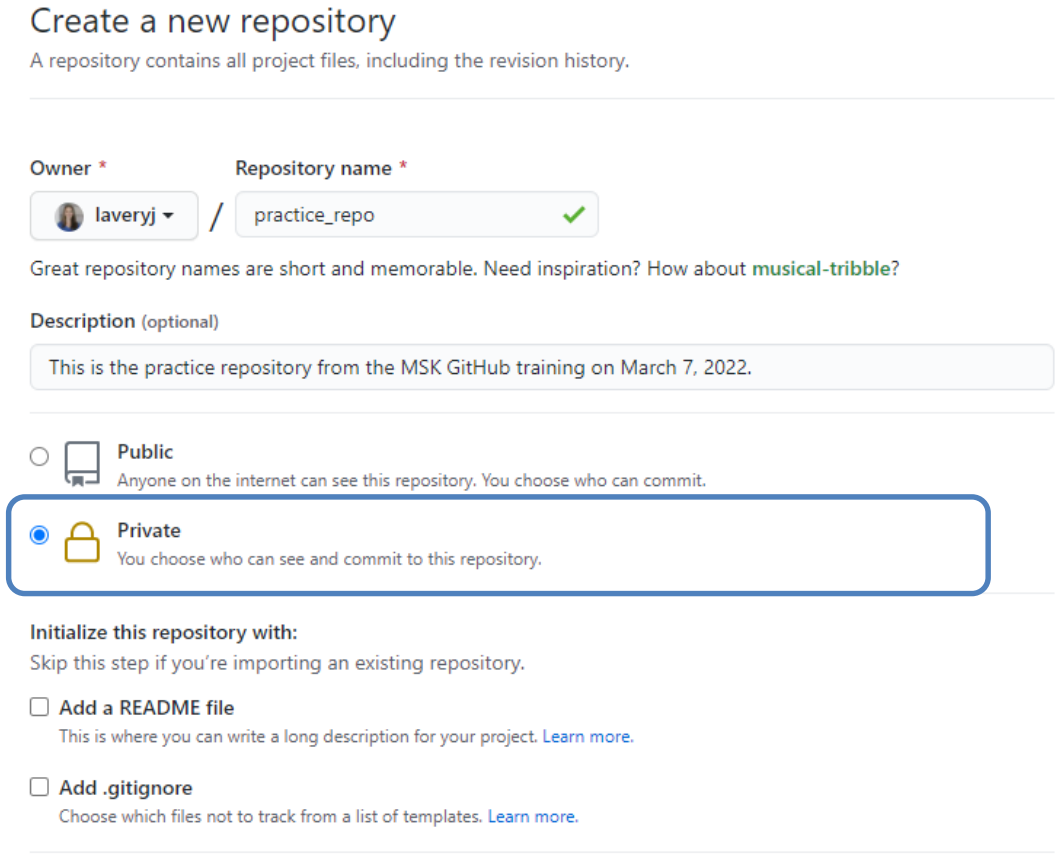
Create a Repository on GitHub



Provide an informative description.
For today we can write: *This is the practice repository from the MSK GitHub training on March 7, 2022.*

In practice, you might want to specify the investigator/cancer site/project topic.

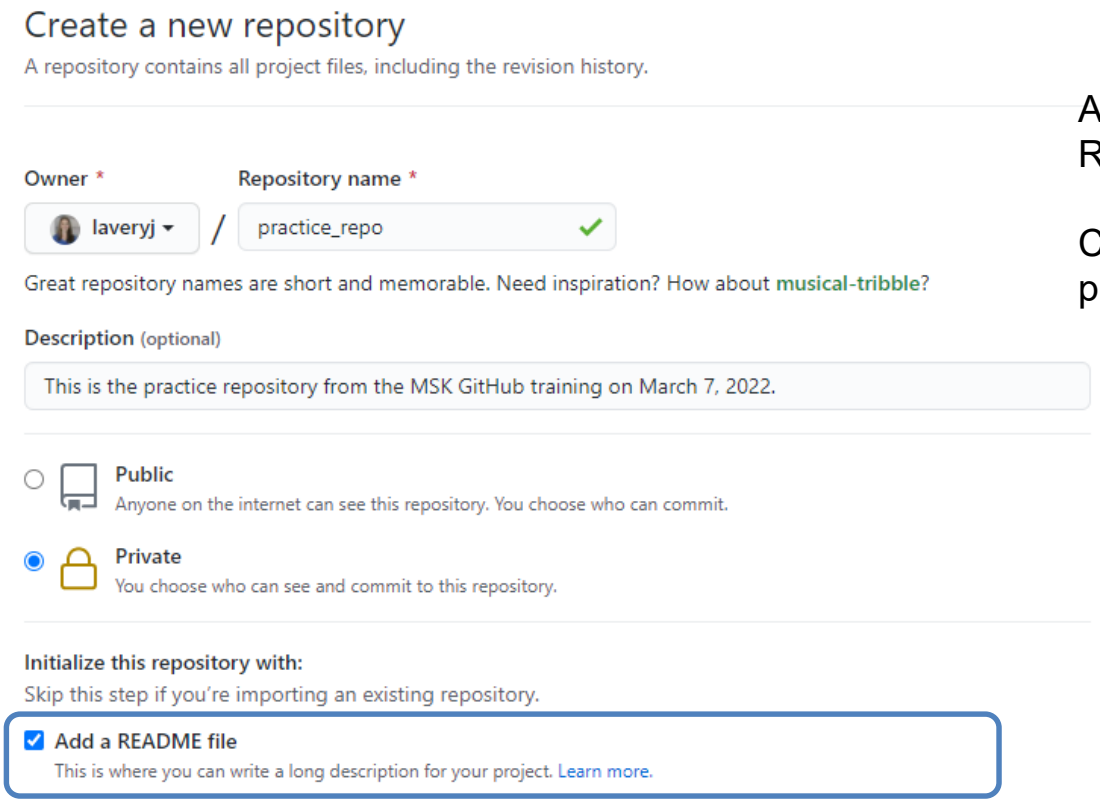
Create a Repository on GitHub



Generally, it's best practice to set repositories to private.

More to come on this later with respect to PHI.

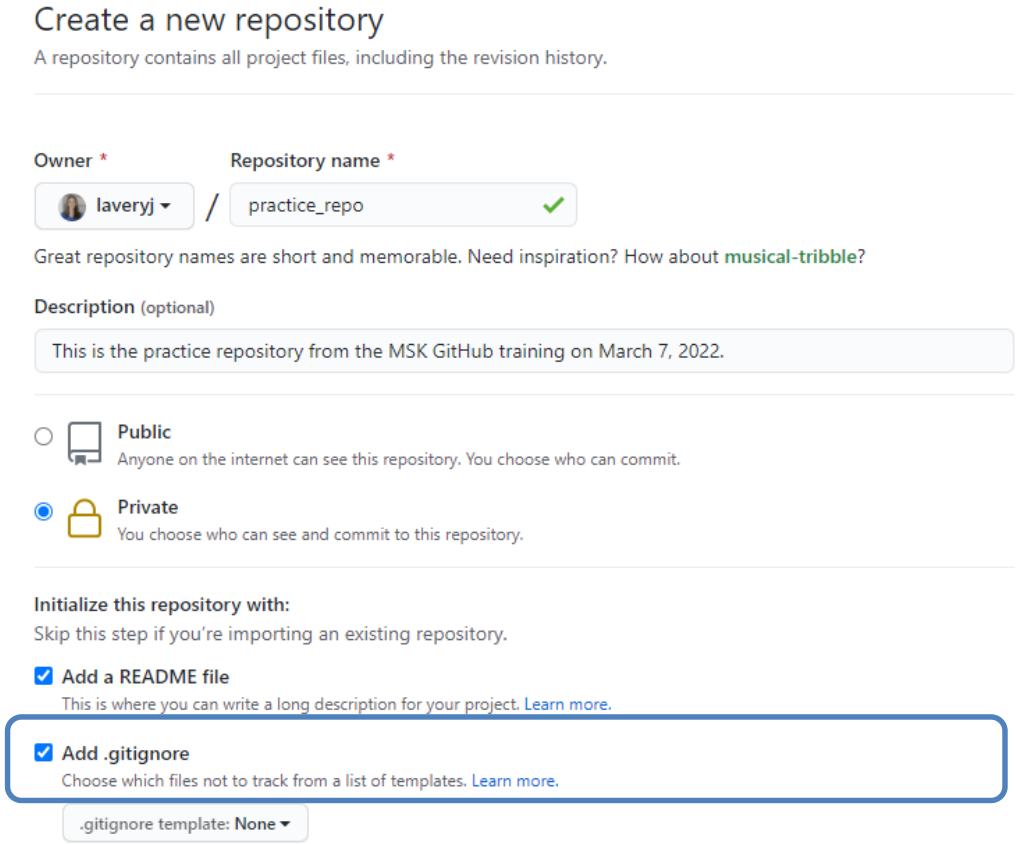
Create a Repository on GitHub



Always add a README.

Can expand on the project description.

Create a Repository on GitHub



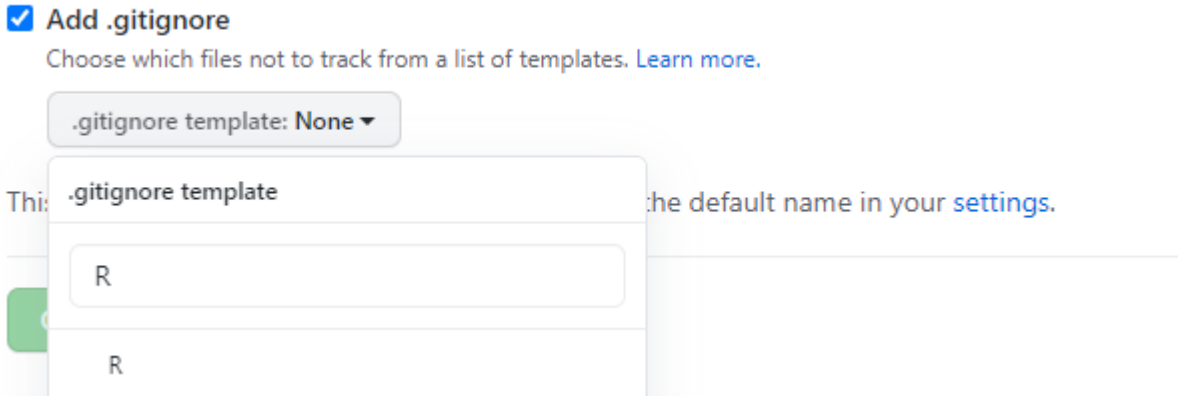
Also always add a .gitignore.

This tells git which files to not pay attention to.

.gitignore

- There are files you don't want Git to check in to GitHub.
  - –This means they will only live locally on your machine
- Git uses the .gitignore file to determine which files and directories to ignore, before you make a commit
- This is important because there are files that don’t belong on GitHub (e.g., data, .Rhistory, etc.)
- Some systems hide files by default that begin with a dot, so you won’t be able to see your .gitignore file. Update the default in Windows Explorer to show all files.
- To prevent PHI from mistakenly being pushed to MSKCC GitHub, immediately initialize a new repository with the provided .gitignore file

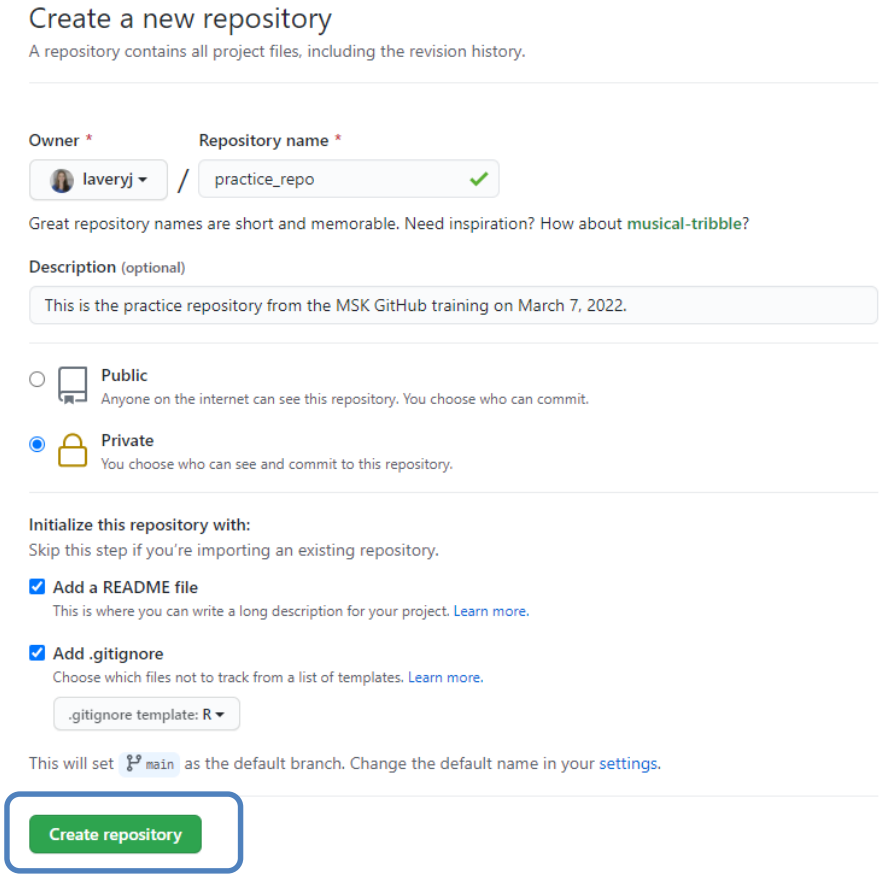
Create a Repository on GitHub



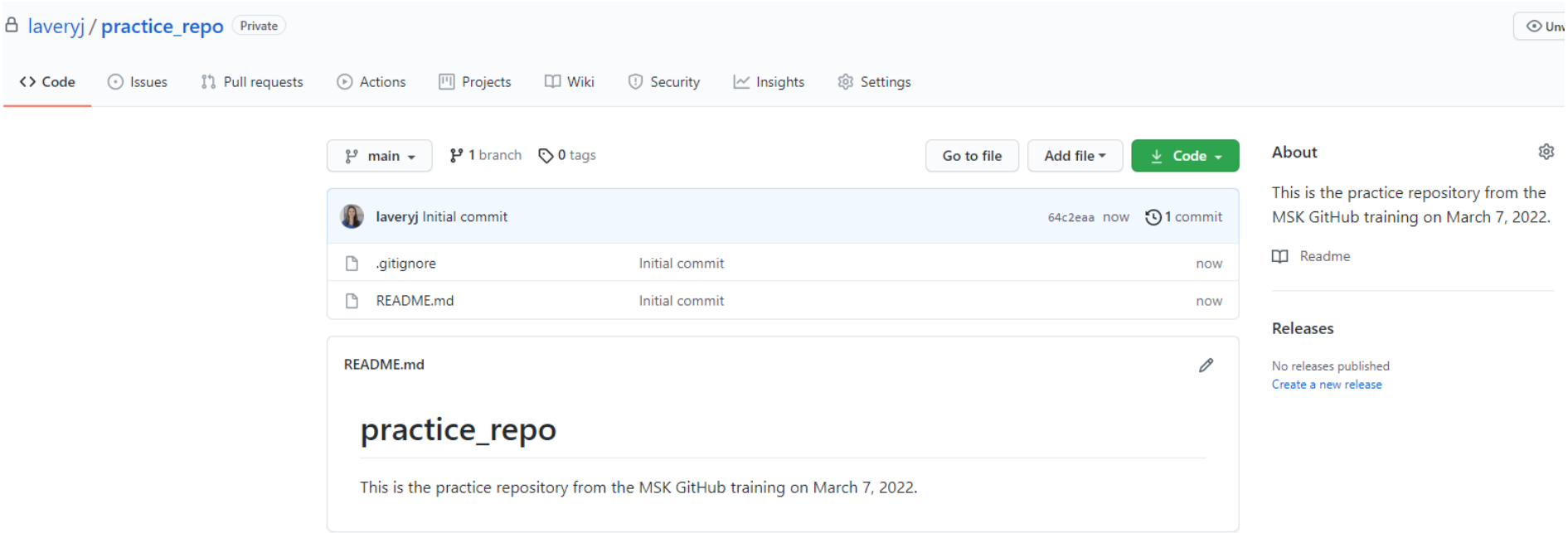
Can select from a pre-populated list of .gitignore templates.

The .gitignore template for R will ignore things like .Rhistory files, .RData files, etc.

Create a Repository on GitHub



Create repo!

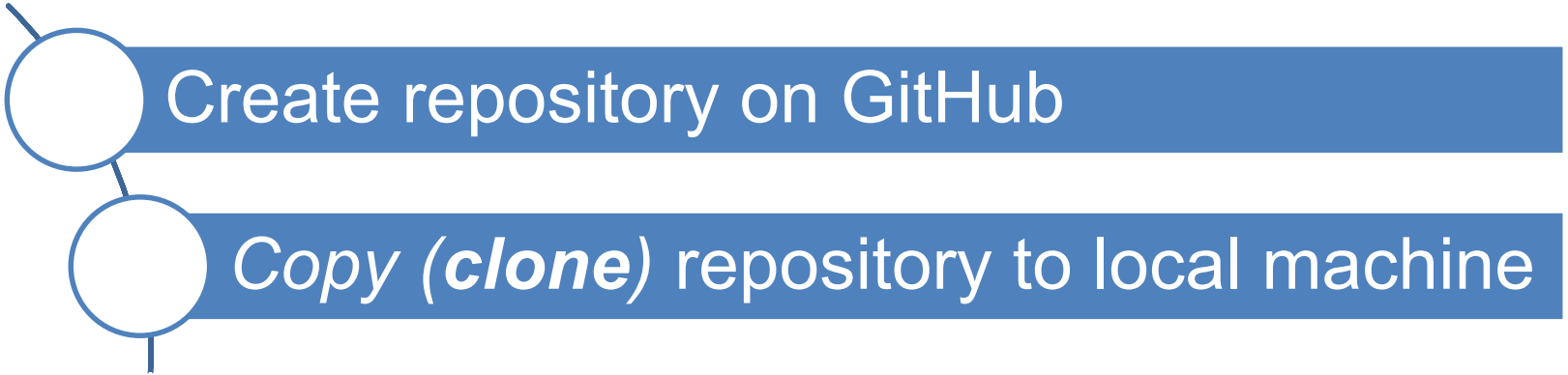

laveryj / practice_repo
Private
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
laveryj Initial commit
64c2eaa now
1 commit
.gitignore
Initial commit
now
README.md
Initial commit
now
README.md
practice_repo
This is the practice repository from the MSK GitHub training on March 7, 2022.
About
This is the practice repository from the MSK GitHub training on March 7, 2022.
Readme
Releases
No releases published
Create a new release

Current Status

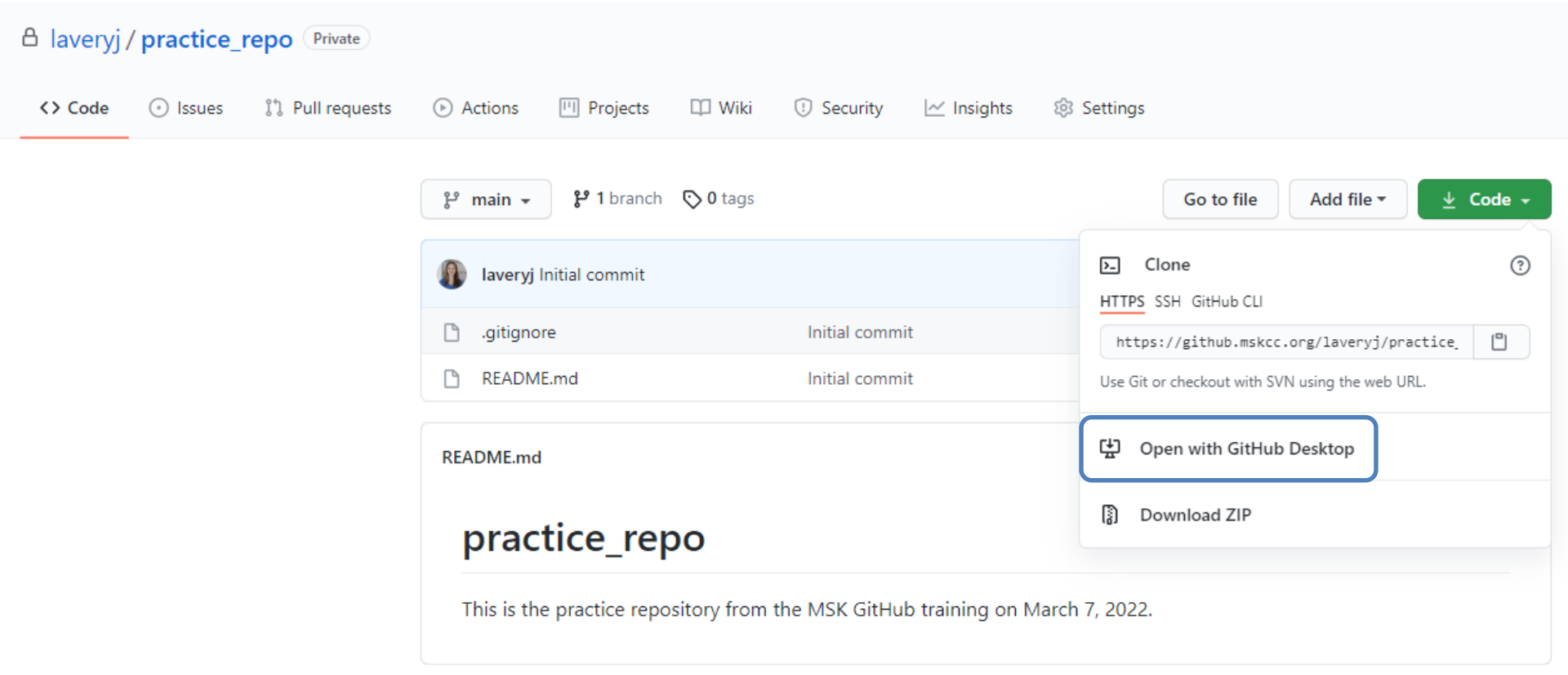
 Local Folder	 Local Repository	 Remote Repository
-	-	Repository exists here

Questions / Roadblocks?

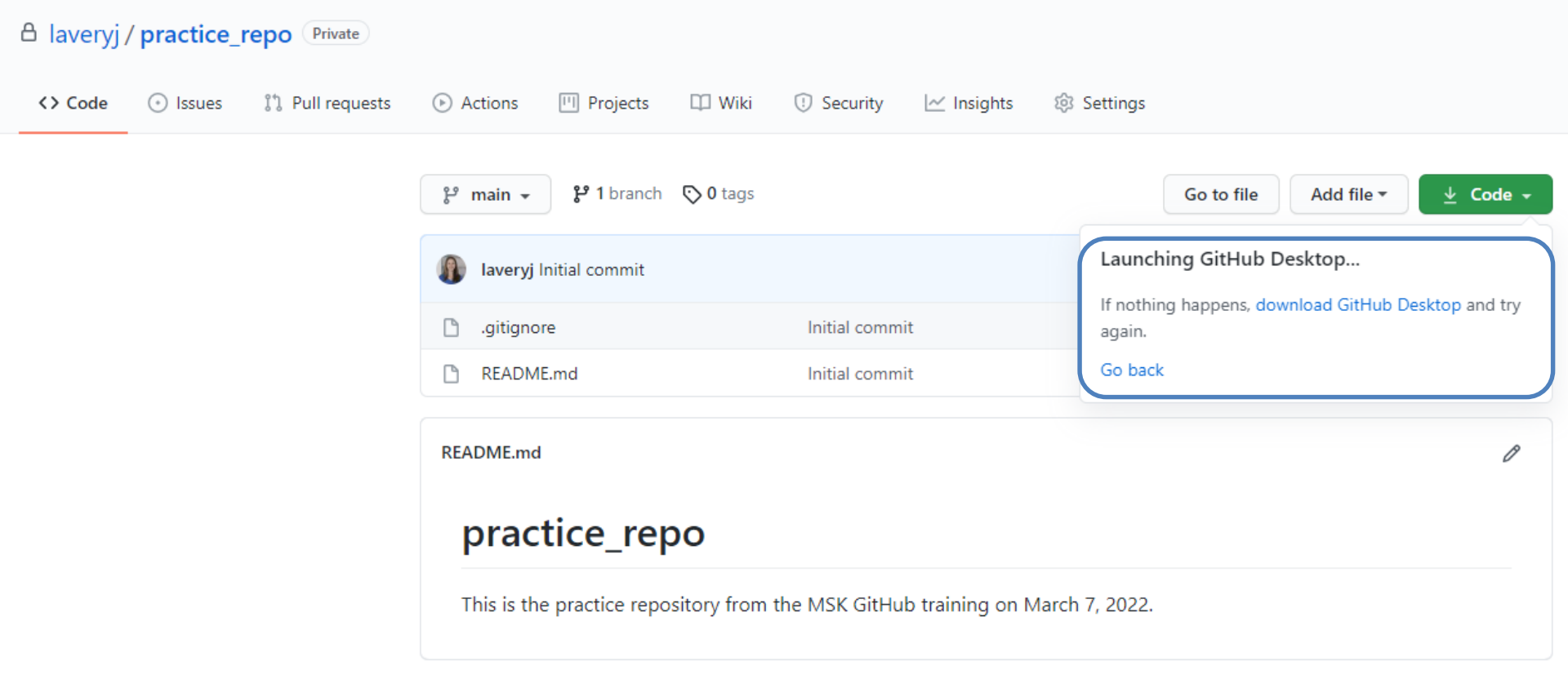
Exercise: Create a Repository



Copying the Repository from GitHub to Your Workstation

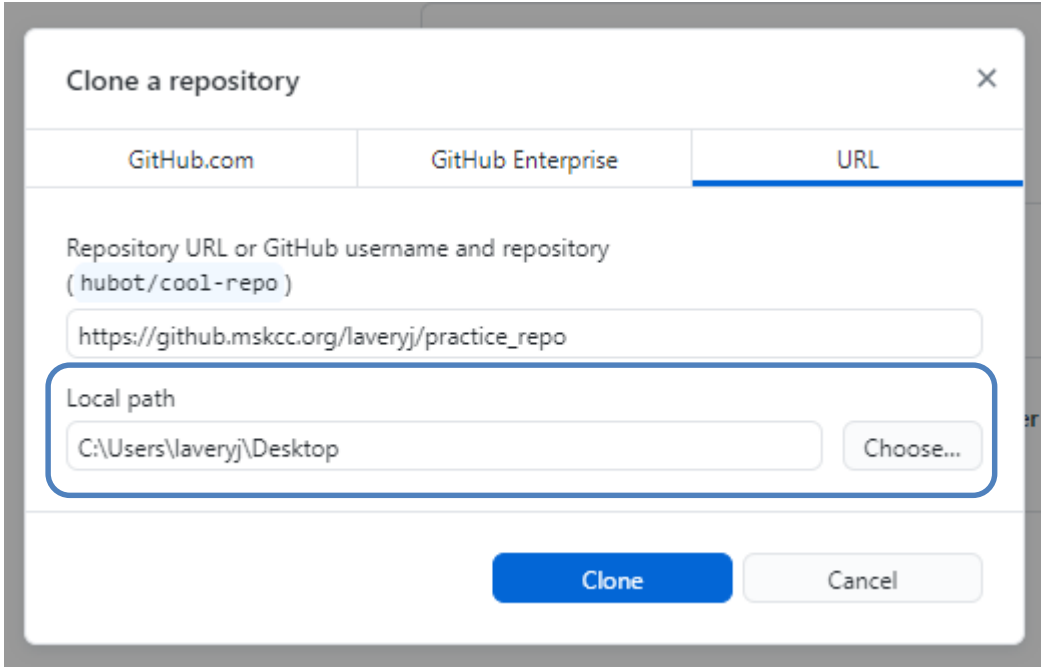


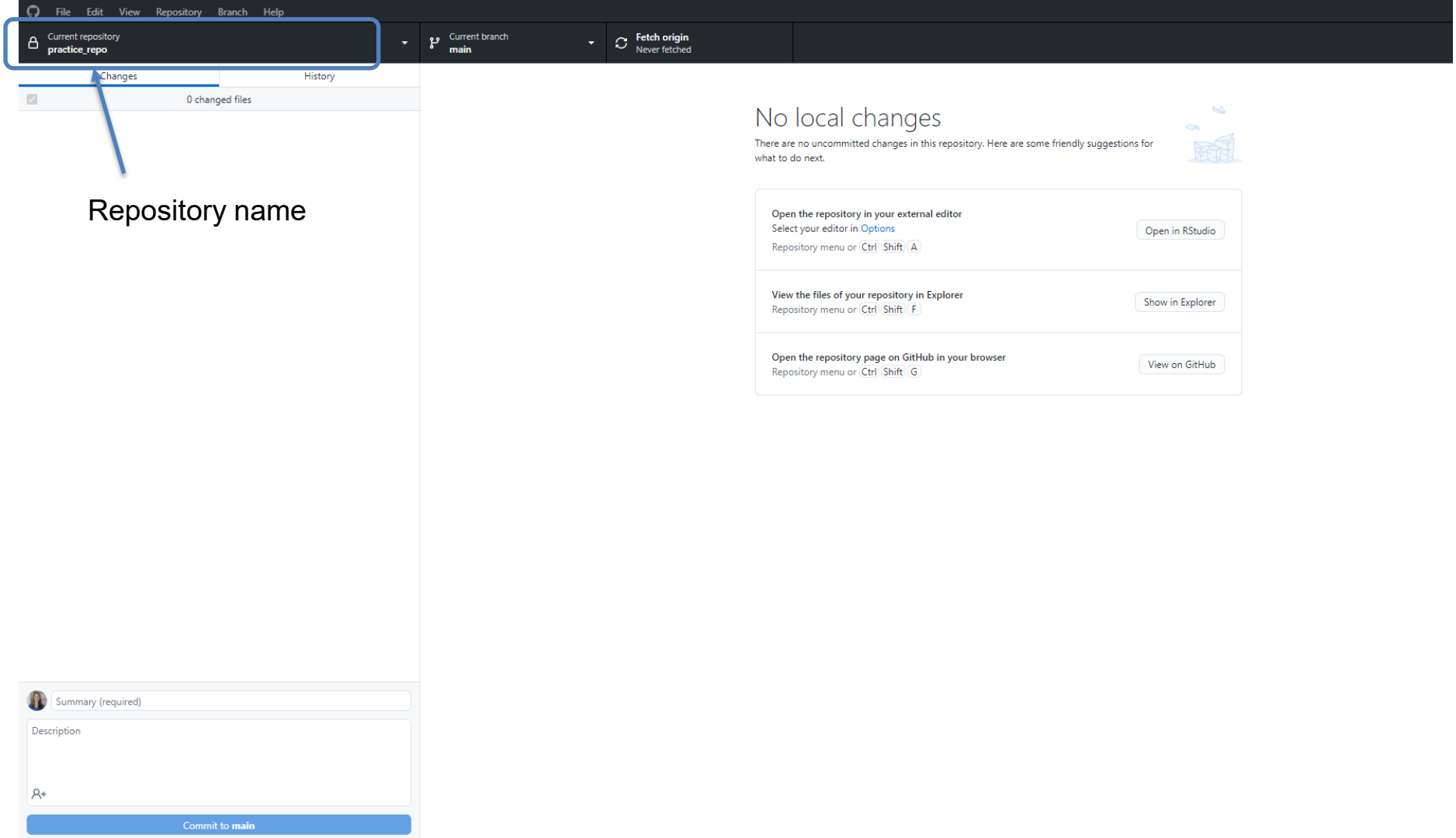
Copying the Repository from GitHub to Your Workstation

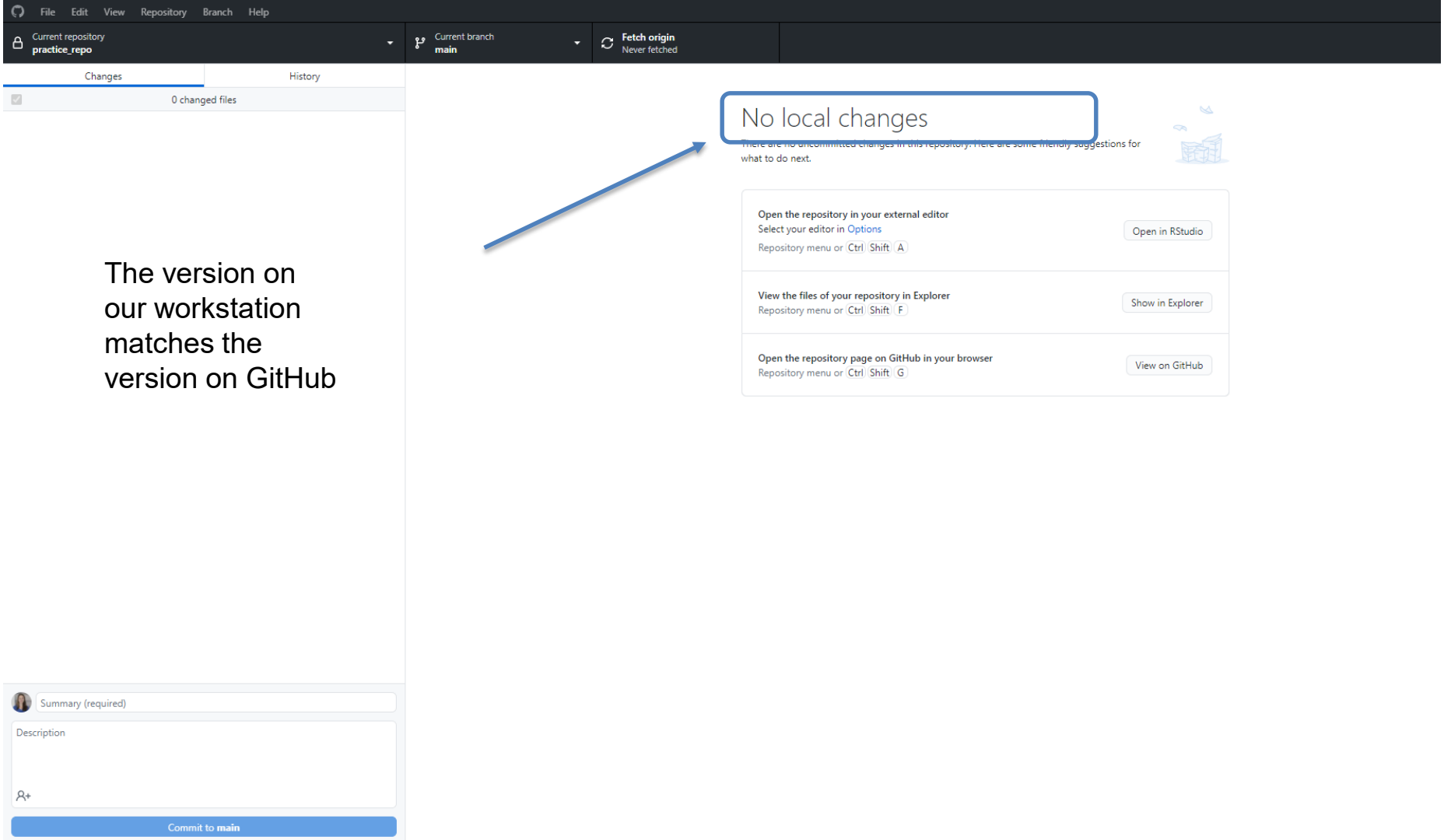


Copying the Repository from GitHub to Your Workstation

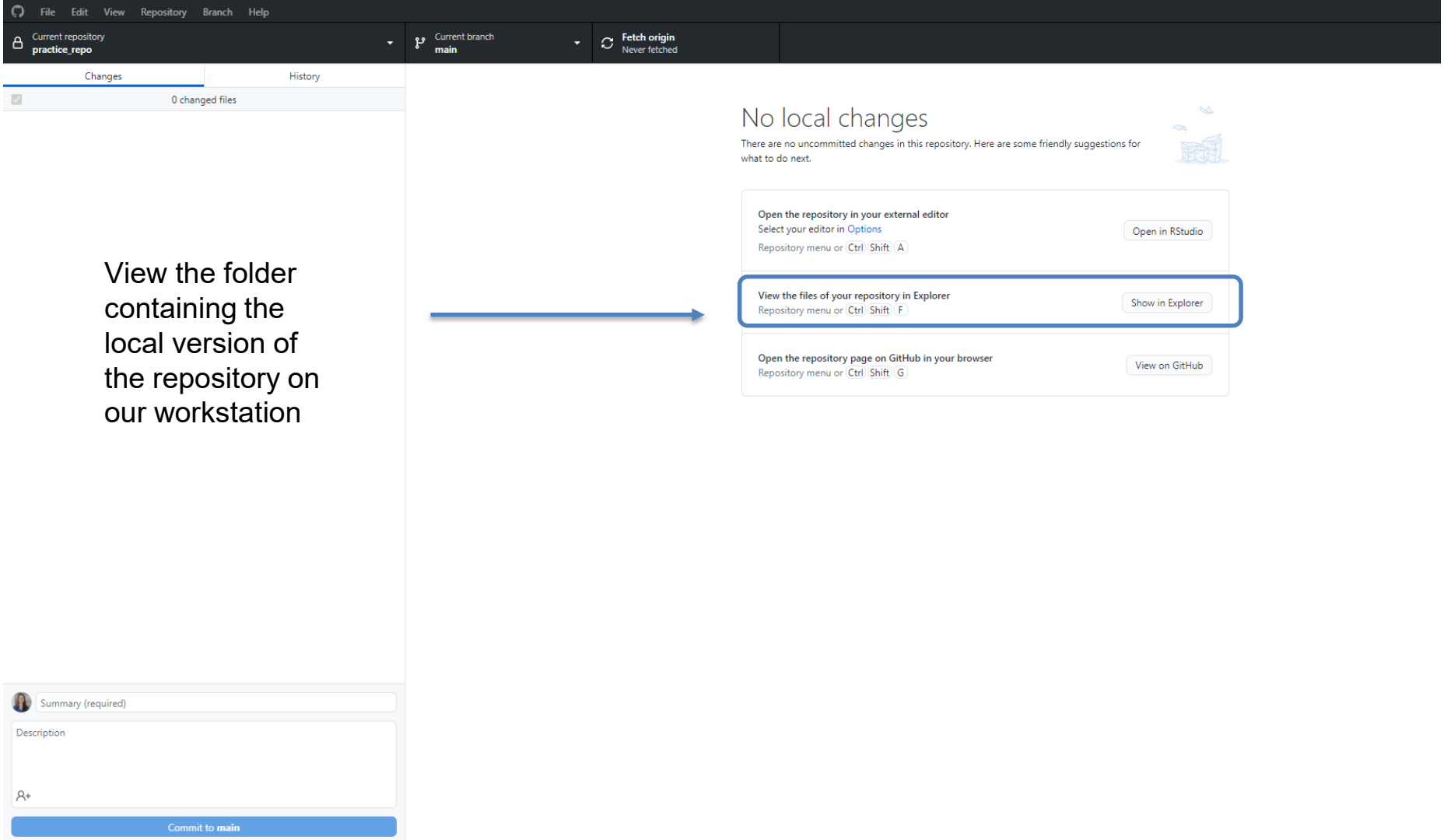
- GitHub Desktop
- Change the local path to your preferred location for the folder



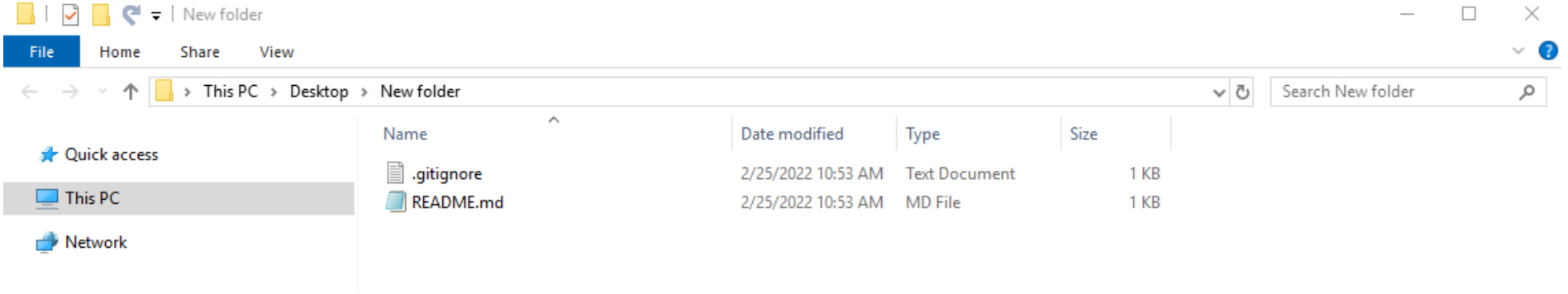

File Edit View Repository Branch Help
Current repository
practice_repo
Current branch
main
Fetch origin
Never fetched
Changes
History
0 changed files
Repository name
No local changes
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.
Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A
Open in RStudio
View the files of your repository in Explorer
Repository menu or Ctrl Shift F
Show in Explorer
Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G
View on GitHub
Summary (required)
Description
Commit to main



The version on our workstation matches the version on GitHub



View the folder containing the local version of the repository on our workstation


New folder
File
Home
Share
View
This PC
Desktop
New folder
Search New folder
Quick access
This PC
Network
Name
Date modified
Type
Size
.gitignore
2/25/2022 10:53 AM
Text Document
1 KB
README.md
2/25/2022 10:53 AM
MD File
1 KB

Current Status

Local Folder

Local Repository

Remote Repository

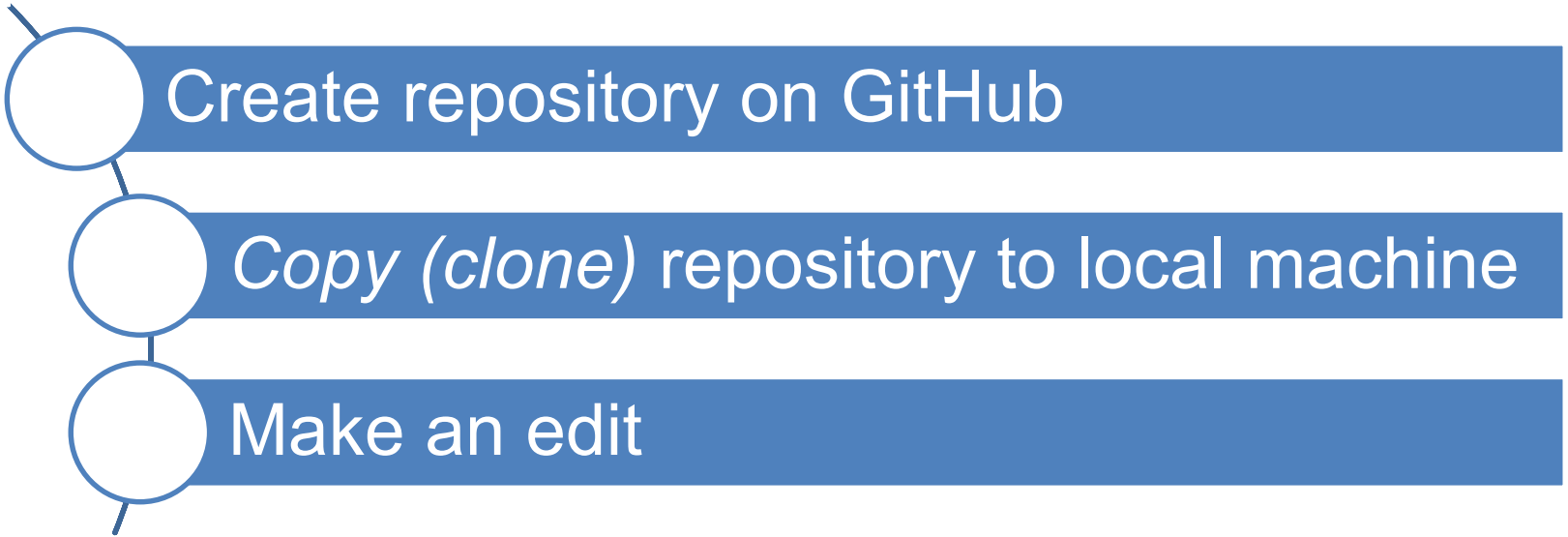
Same version of repository exists on GitHub and on your workstation

Questions / Roadblocks?

We have:

- Created a repository
- Copied that repository from GitHub to our workstation

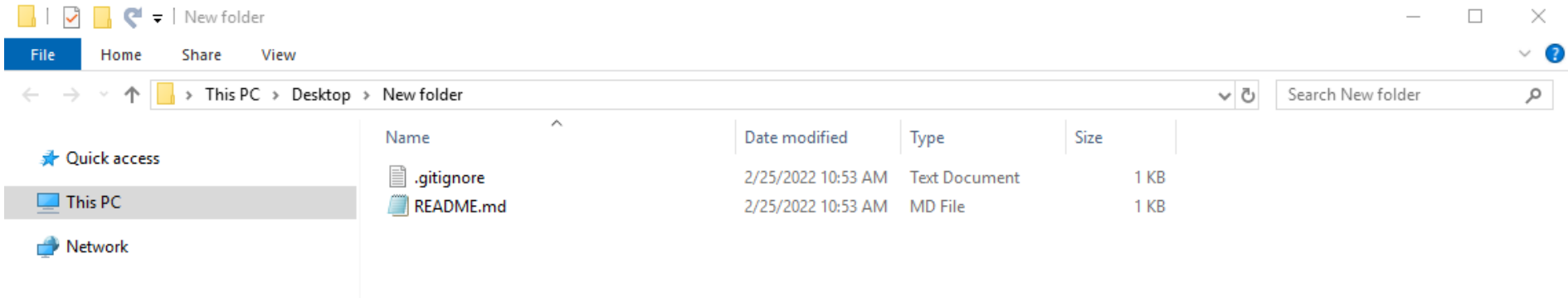
Exercise: Create a Repository

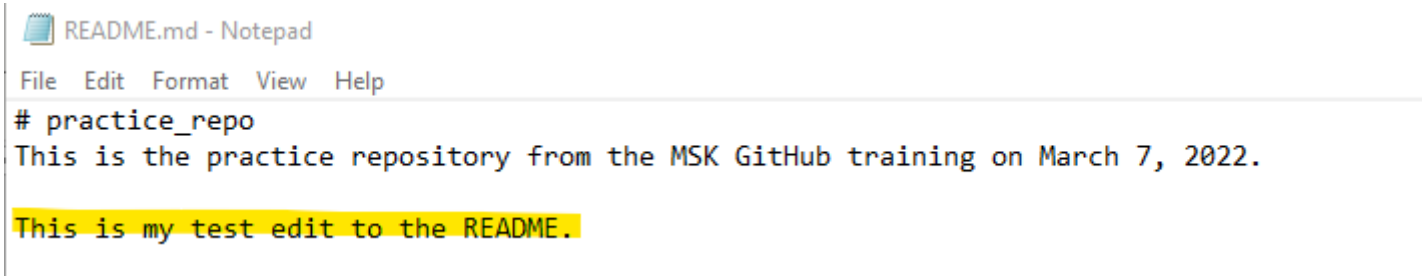


Edit the README.md File

1. Navigate to the local version of your repository
2. Open the README.md file (Notepad should open it by default)
3. Add a line of text to the file
4. Hit Save

Edit the README.md File





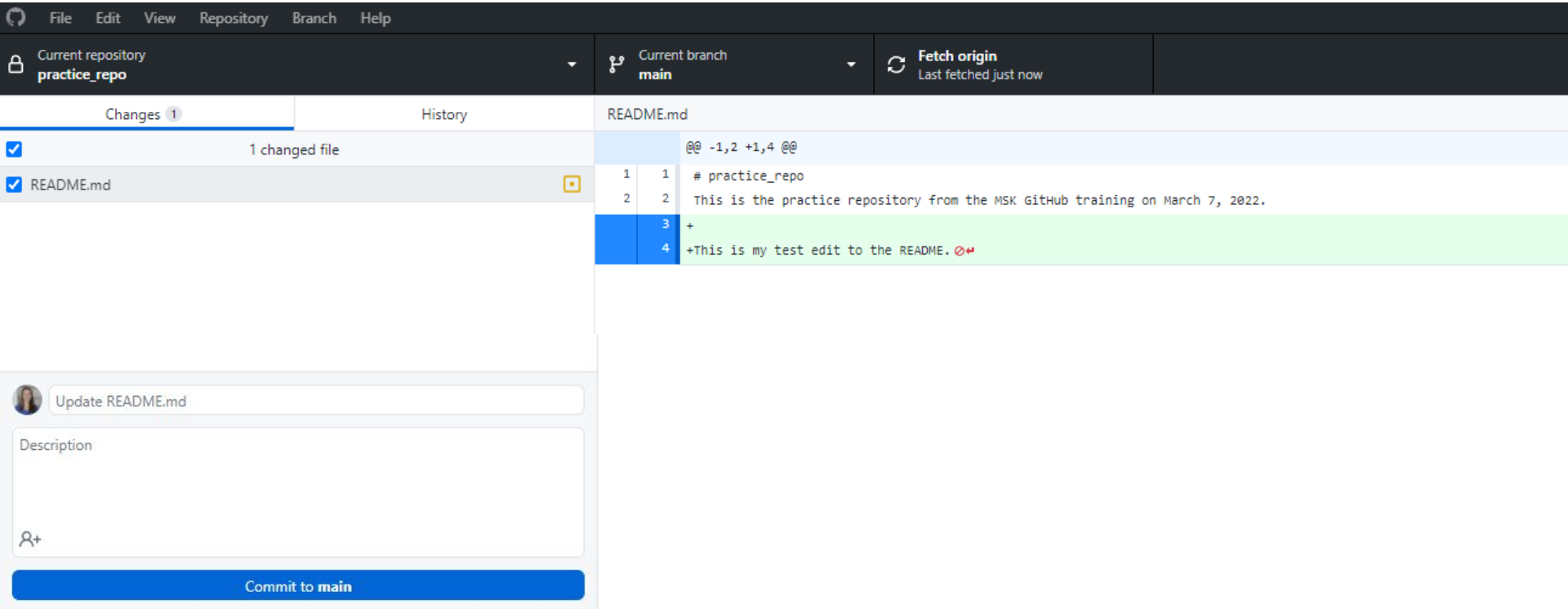
Current Status

Local Folder	Local Repository	Remote Repository
Modified repository	**Original, unchanged** repository	**Original, unchanged** repository

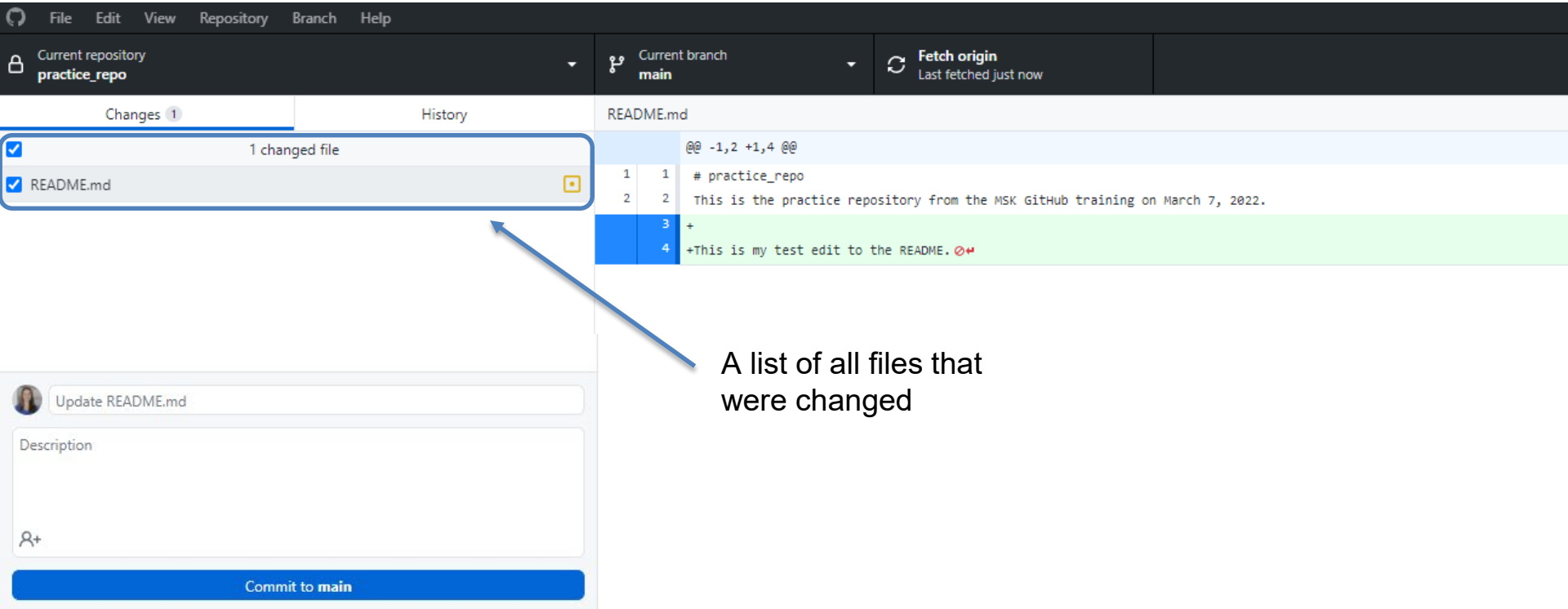
Git Commits

- When you make a change locally, git doesn't actually track the change until you tell it to. When you are ready to finalize changes that you made, you ***commit*** changes to the repository.
  - This **only** commits the change locally
  - Git is aware of your change, GitHub is not
- Commit changes (i.e., notify git of changes) using GitHub Desktop

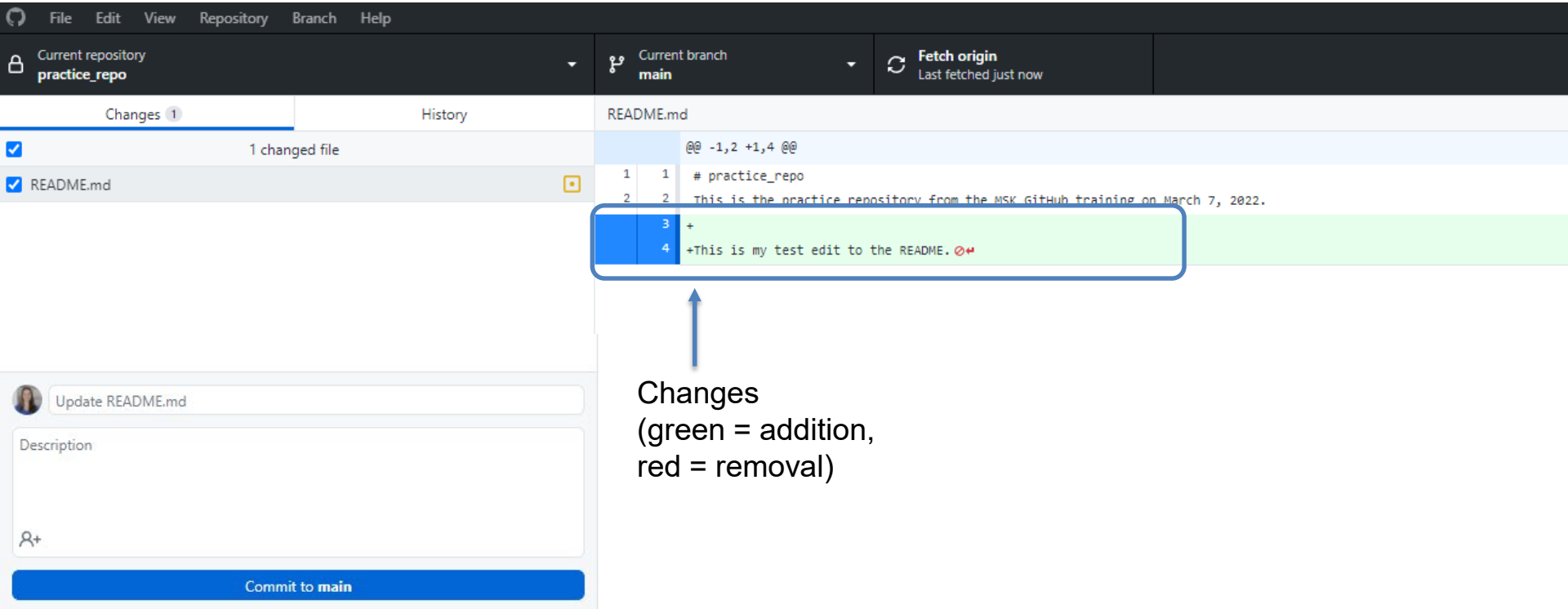
View Repository in GitHub Desktop



View Repository in GitHub Desktop

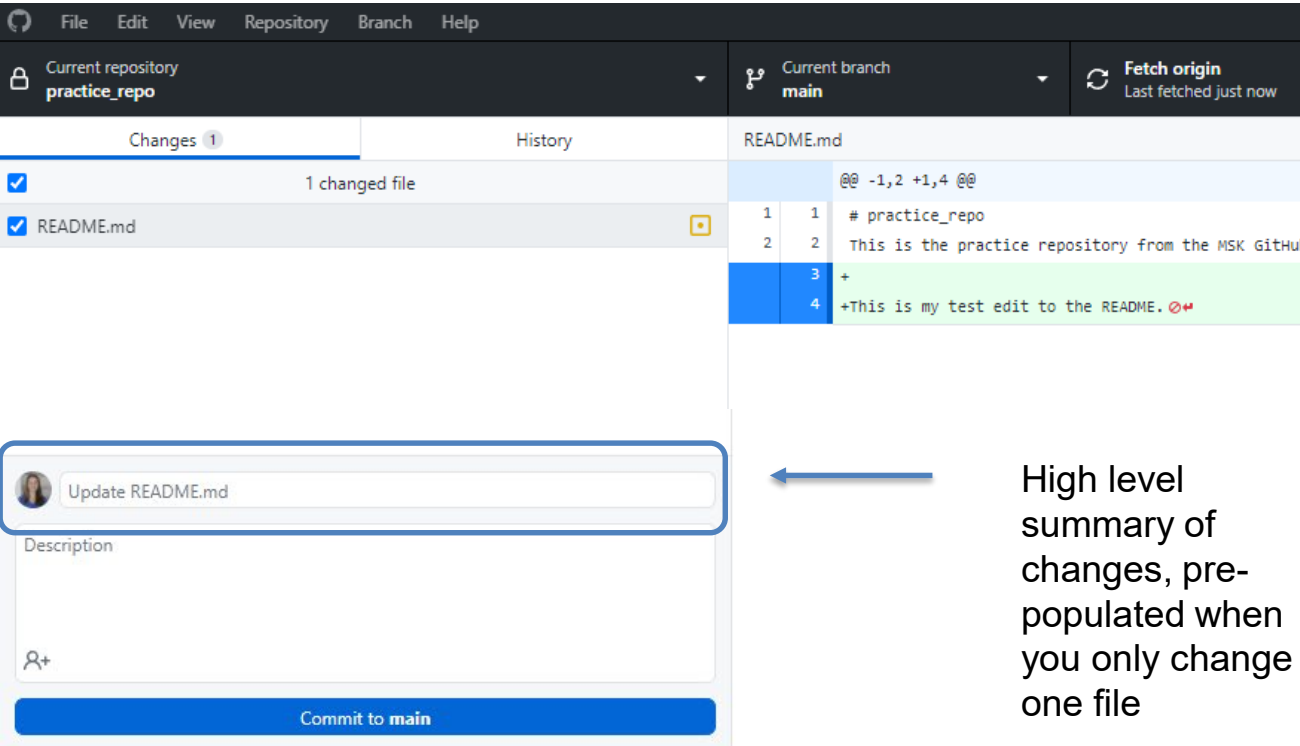


View Repository in GitHub Desktop



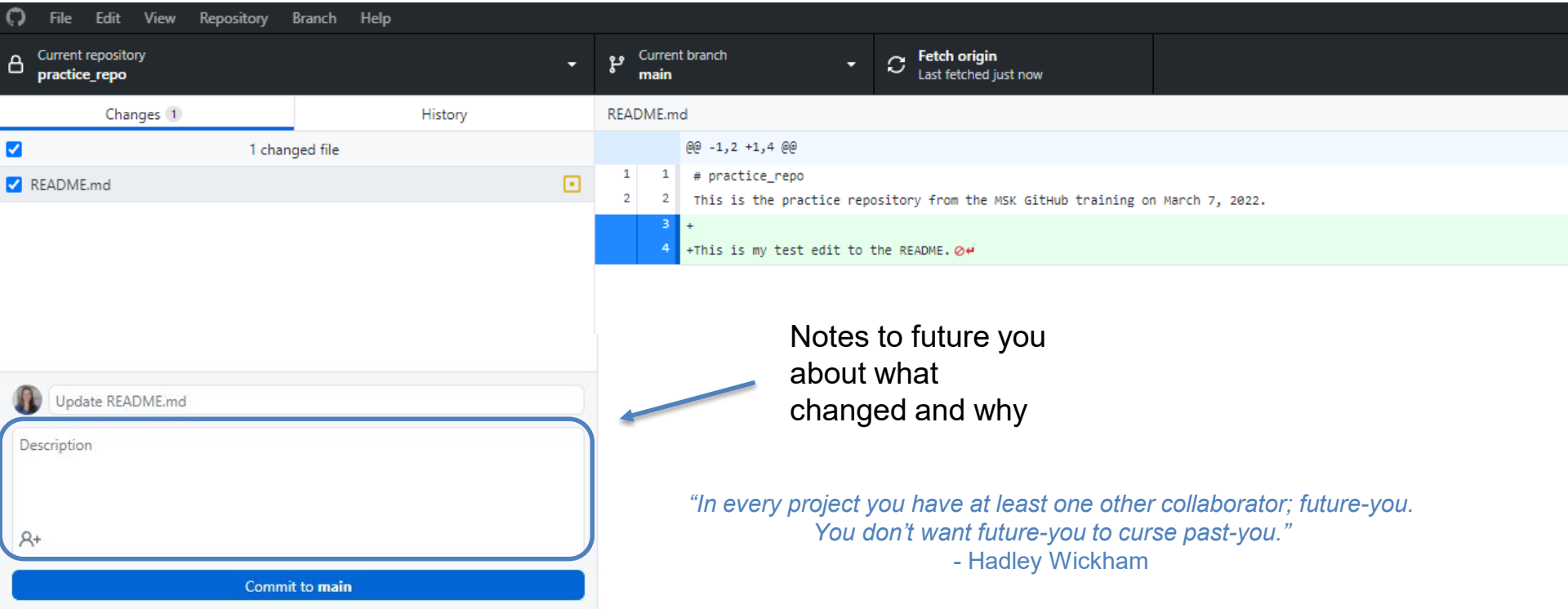
Changes
(green = addition,
red = removal)

View Repository in GitHub Desktop



High level summary of changes, pre-populated when you only change one file

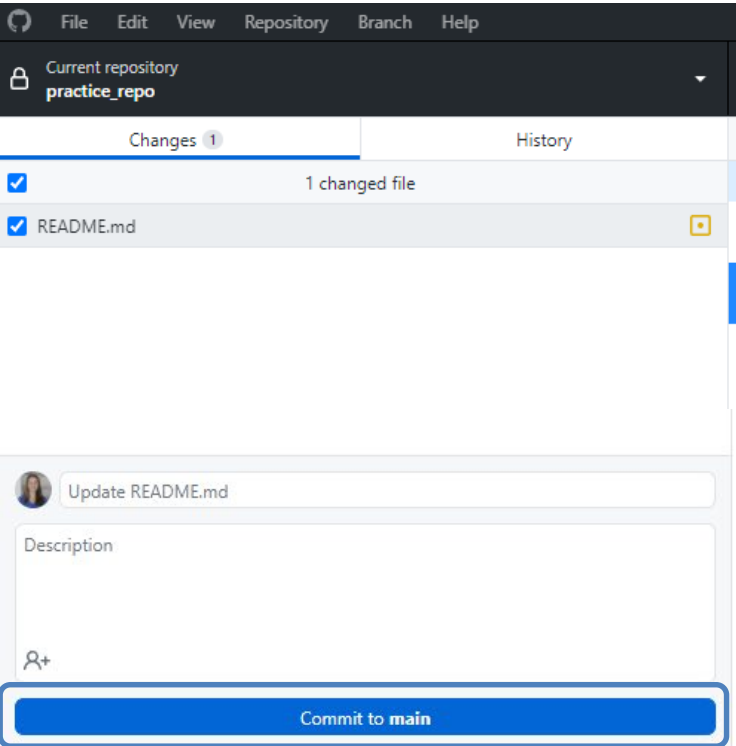
View Repository in GitHub Desktop

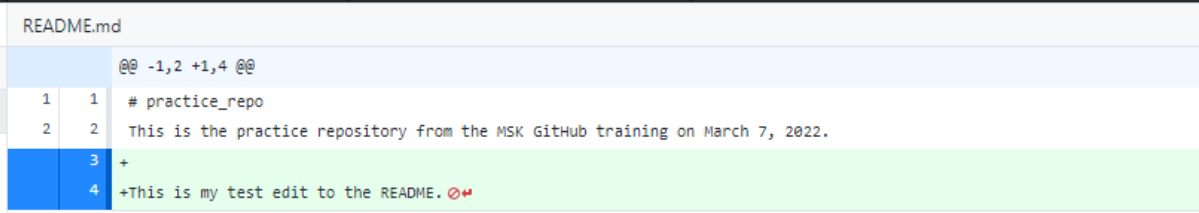


Notes to future you about what changed and why

"In every project you have at least one other collaborator; future-you. You don't want future-you to curse past-you."
- Hadley Wickham

View Repository in GitHub Desktop





Commit!

Current Status

Local Folder

Local Repository

Remote Repository

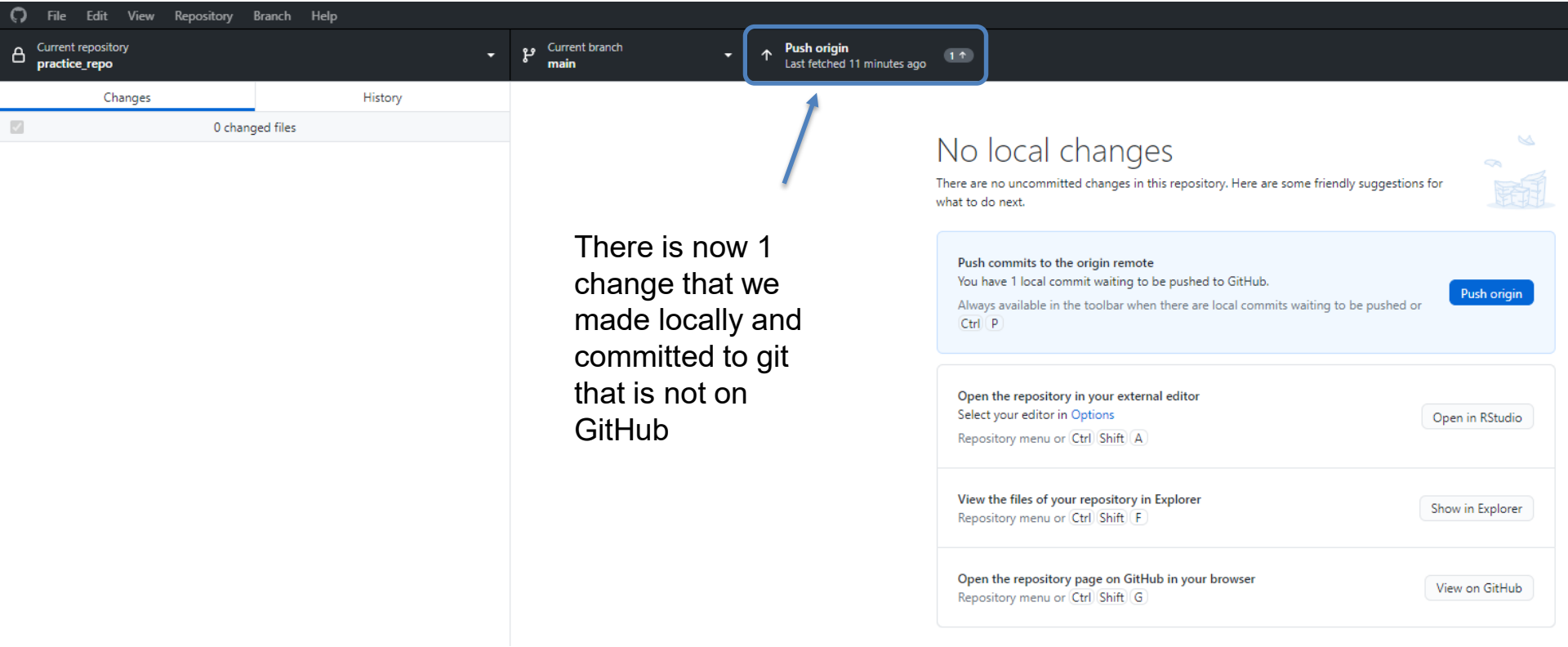
Modified repository

git commit

Modified repository

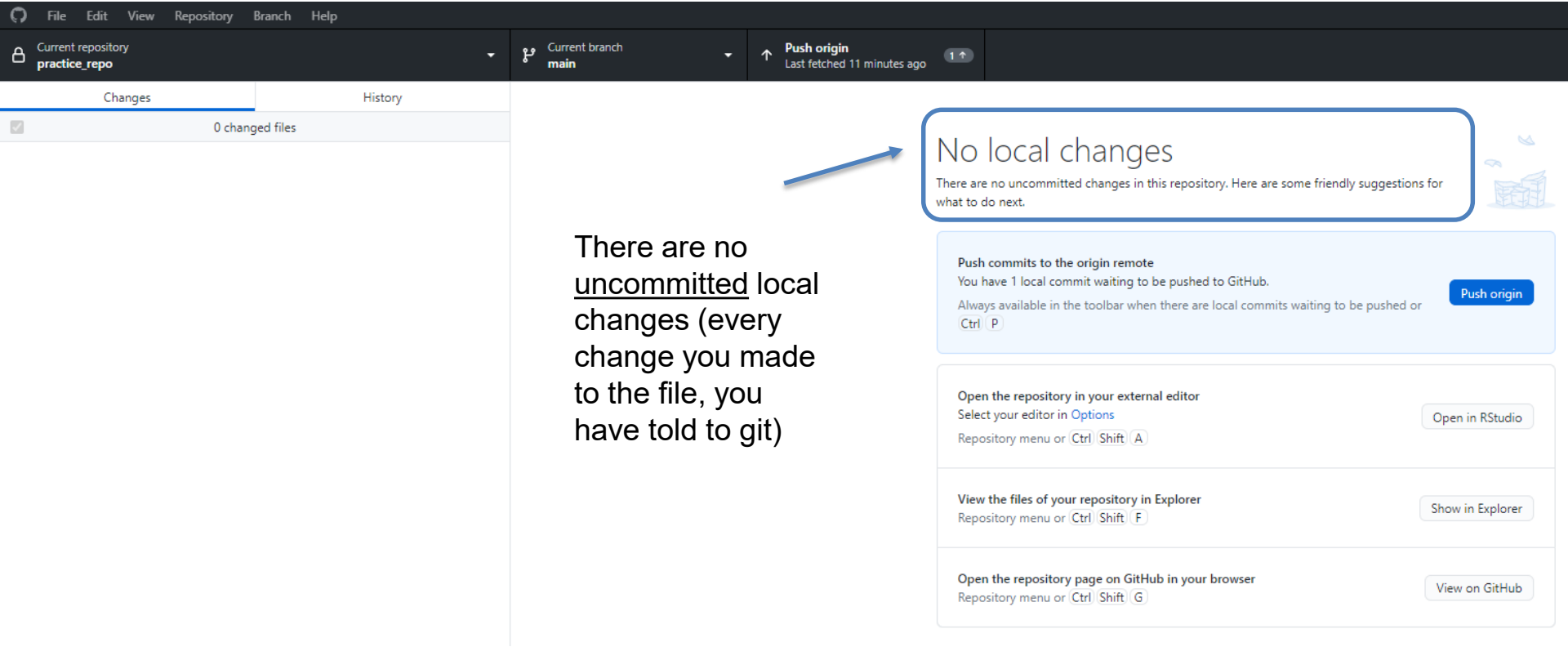
Original, unchanged repository

View Repository in GitHub Desktop



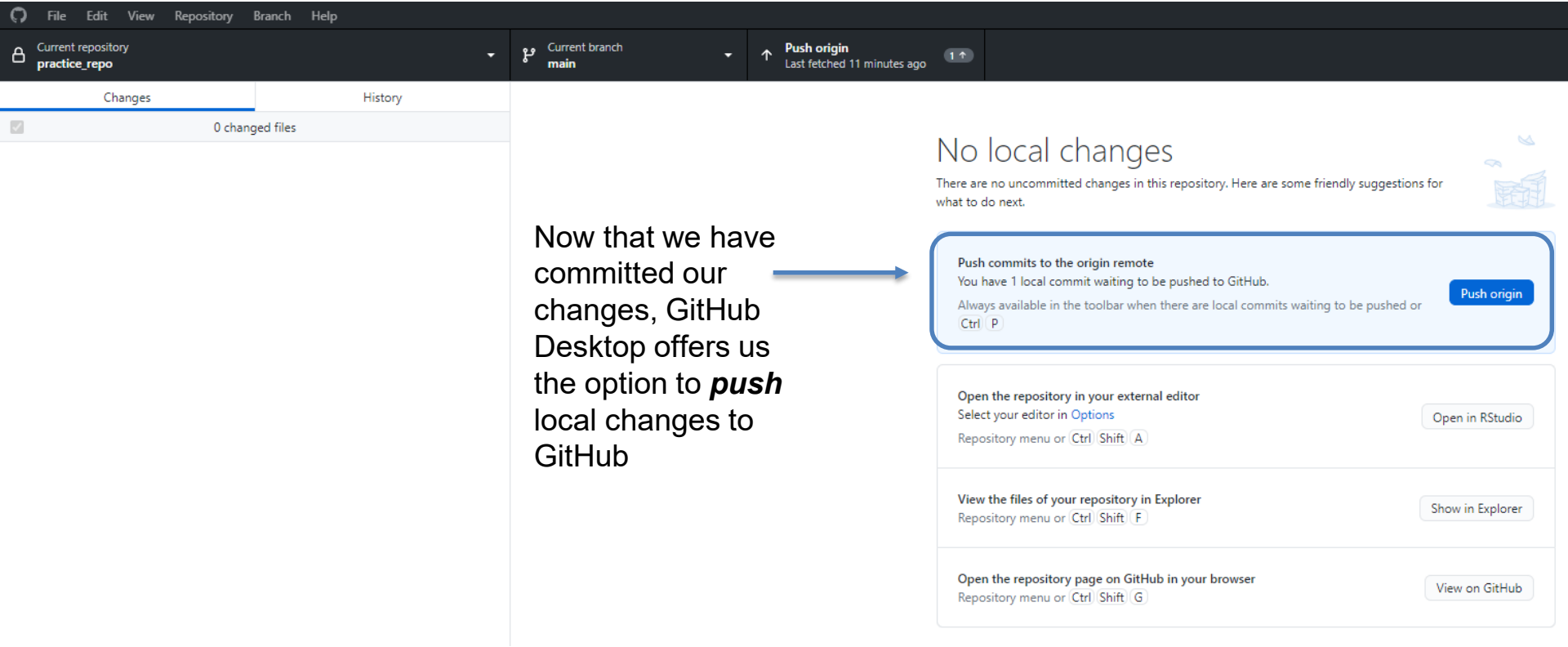
There is now 1 change that we made locally and committed to git that is not on GitHub

View Repository in GitHub Desktop



There are no uncommitted local changes (every change you made to the file, you have told to git)

View Repository in GitHub Desktop

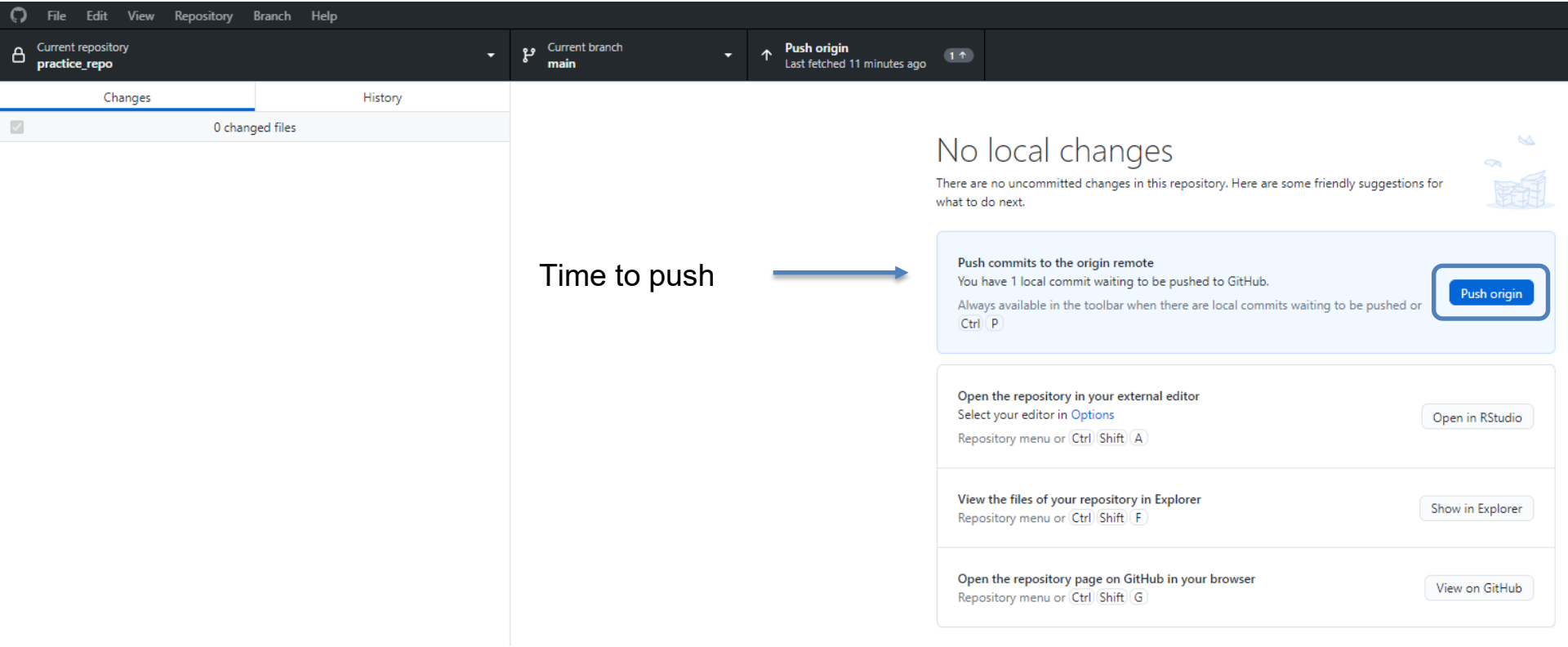


Now that we have committed our changes, GitHub Desktop offers us the option to ***push*** local changes to GitHub

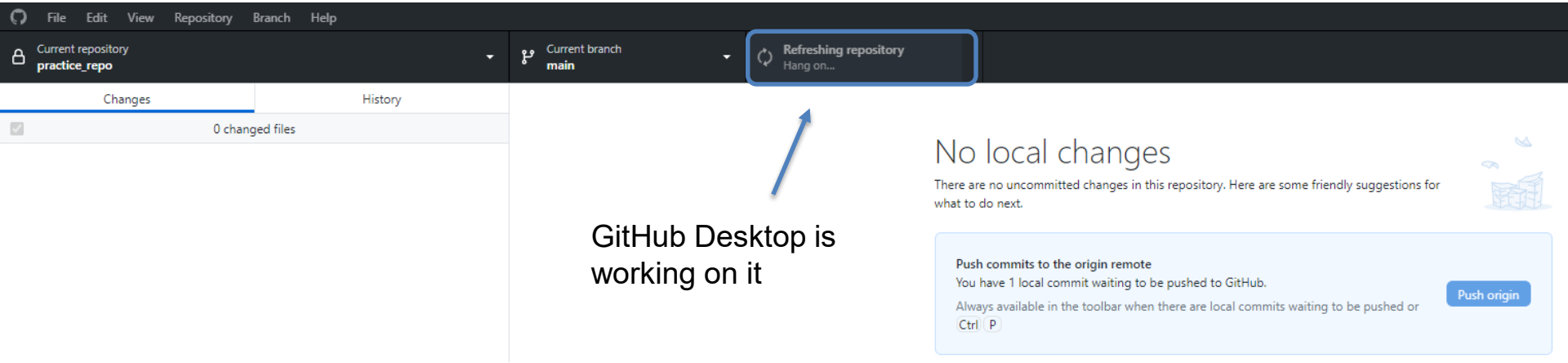
Git Push

- ***Push*** changes from the local version of your repository to the remote (GitHub) version of the repository
  - –Git notifies GitHub of your changes
- Push changes using GitHub Desktop

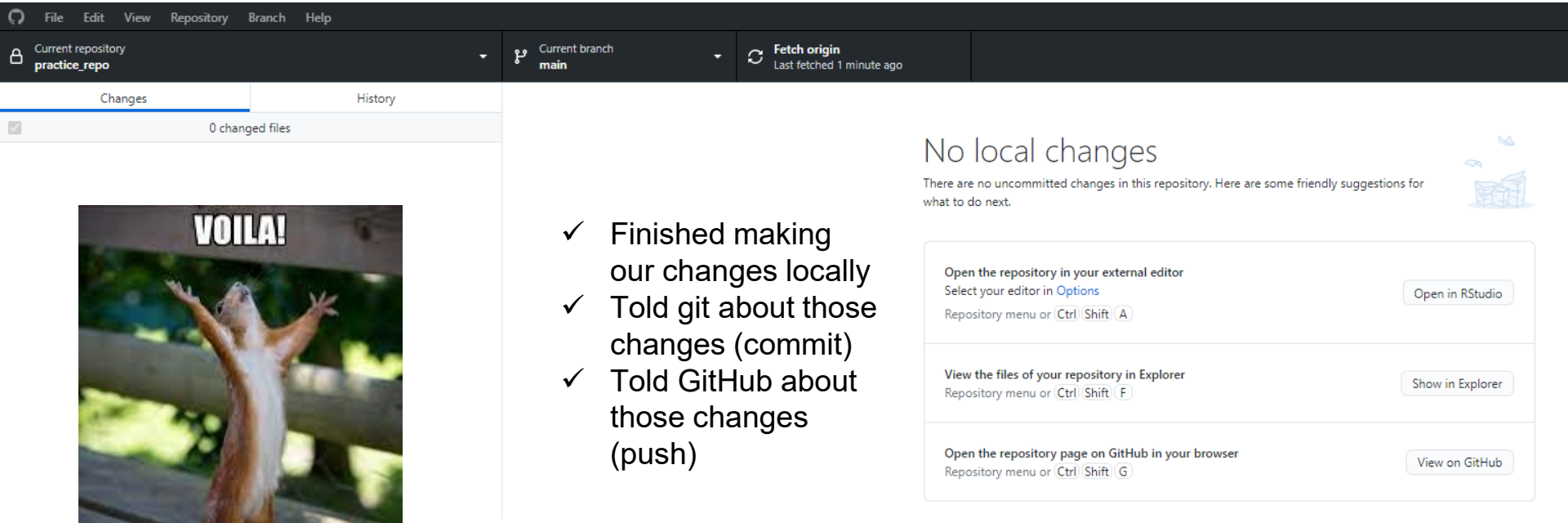
View Repository in GitHub Desktop



View Repository in GitHub Desktop

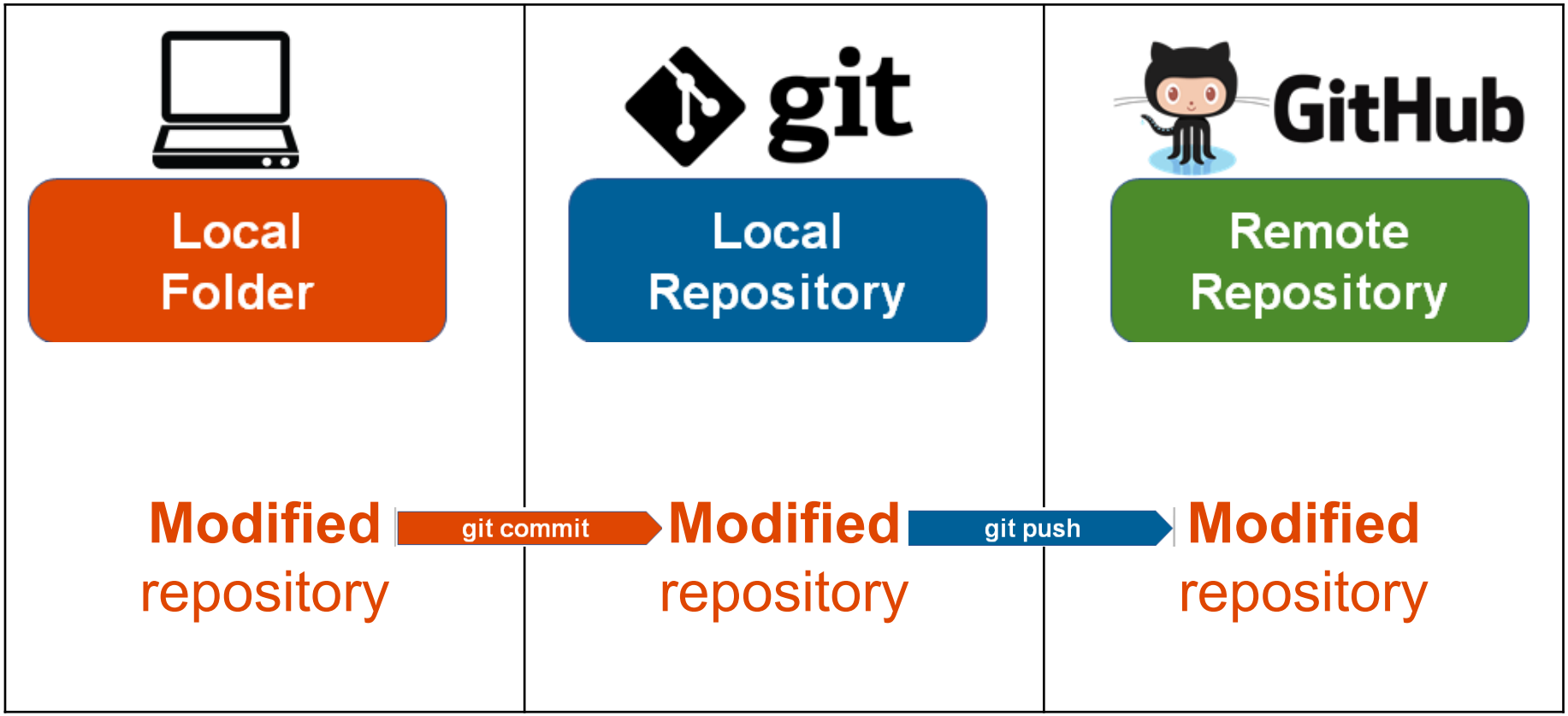


View Repository in GitHub Desktop



- ✓ Finished making our changes locally
- ✓ Told git about those changes (commit)
- ✓ Told GitHub about those changes (push)

Current Status



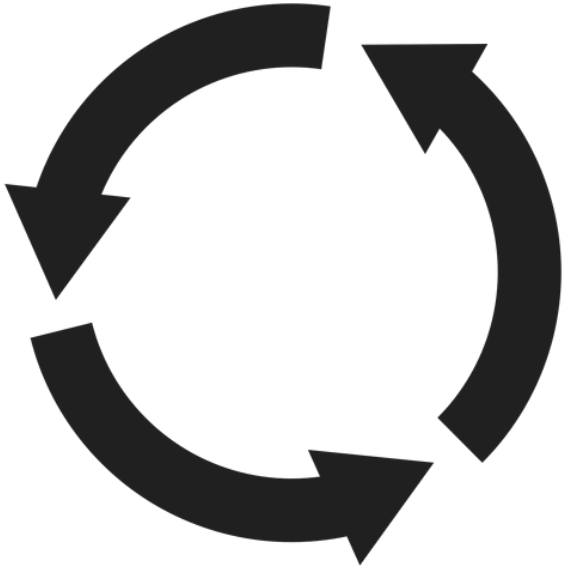
Questions / Roadblocks?

We have:

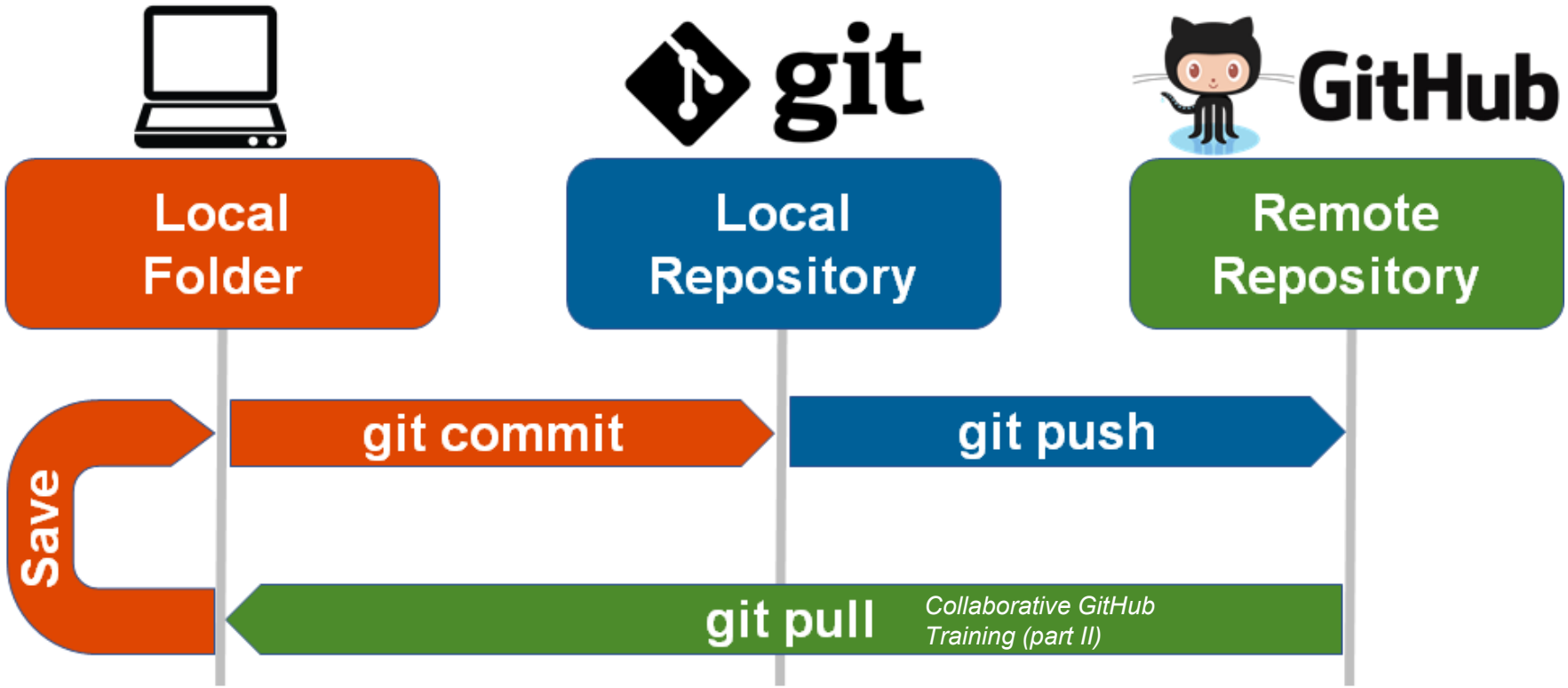
- Created a repository
- Copied that repository from GitHub to our workstation
- Made a change to the local version of the repository (on our workstations)
- Pushed those changes to the remote version of the repository

Recap: Workflow Steps

1. Create a GitHub **repo** on GitHub
2. **Clone** repo with GitHub Desktop
3. Start your project, make changes locally, save, and **commit**
4. **Push** local changes to GitHub
5. Repeat steps 3-4 as you update code



Git & GitHub Workflow



Additional Information

- RStudio has built-in integration with git (We still prefer GitHub Desktop)
- Can create a repository in many ways

Thank You!

Questions?