



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

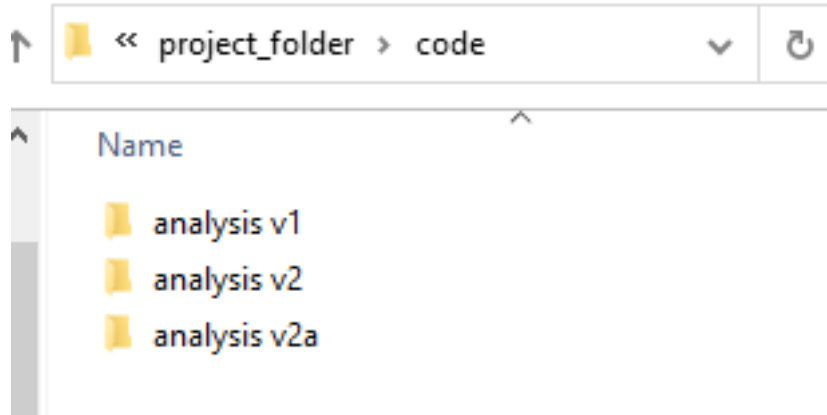
“In every project you have at least one other collaborator; future-you. You don’t want future-you to curse past-you.”

- 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

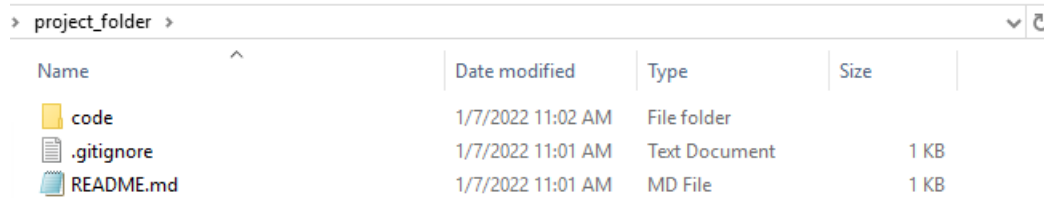


When the paper comes back for revisions:

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



The screenshot shows a Windows File Explorer window with the address bar set to 'project_folder'. The window displays a table of files and folders. The table has four columns: 'Name', 'Date modified', 'Type', and 'Size'. There are three items listed: a folder named 'code', a text file named '.gitignore', and a markdown file named 'README.md'. All items were modified on 1/7/2022 at 11:01 AM, except for the 'code' folder which was modified at 11:02 AM.

Name	Date modified	Type	Size
code	1/7/2022 11:02 AM	File folder	
.gitignore	1/7/2022 11:01 AM	Text Document	1 KB
README.md	1/7/2022 11:01 AM	MD File	1 KB

- 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"
3 3 author: "Jessica Lavery"
4 4 date: "1/7/2022"
5 5 output: html_document
6 +editor_options:
7 + chunk_output_type: console
6 8 ---
7 9
8 10 ```{r setup, include=FALSE}
@@ -16,7 +18,7 @@ This is an R Markdown document. Markdown is a simple formatting syntax for autho
16 18 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:
17 19
18 20 ```{r cars}
19 -summary(cars)
21 +summary(mtcars)
20 22 ```
21 23
22 24 ## Including Plots
@@ -28,3 +30,10 @@ plot(pressure)
28 30 ```
29 31
30 32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
33 +
34 +## Part 2 of analysis
35 +
36 +```{r}
37 +summary(gtsummary::trial)
38 +```
39 +
```

With git: History of what changed & when

History for [github_demo](#) / [code](#) / [analysis.Rmd](#)

Commits on Jan 7, 2022

Update analysis.Rmd


 **jalavery** committed 4 minutes ago



1269933



Create analysis.Rmd

 **jalavery** committed 6 minutes ago




6e46b4a



With git: History of what changed & when

Update analysis.Rmd

 jalavery committed 4 minutes ago

1 parent 6e46b4a commit 12699336d3da547f78b996bfad64705726dde2ec

Showing 1 changed file with 10 additions and 1 deletion.

code/analysis.Rmd

@@ -3,6 +3,8 @@ title: "Example R Script"

3 3 author: "Jessica Lavery"

4 4 date: "1/7/2022"

5 5 output: html_document

6 + editor_options:

7 + chunk_output_type: console

6 8 ---

7 9

8 10 ```{r setup, include=FALSE}

@@ -16,7 +18,7 @@ This is an R Markdown document. Markdown is a simple formatting syntax for autho

16 18 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.

17 19 You can embed an R code chunk like this:

18 20 ```{r cars}

19 - summary(cars)

21 + summary(mtcars)

20 22 ```

Questions on git?

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

What is GitHub?

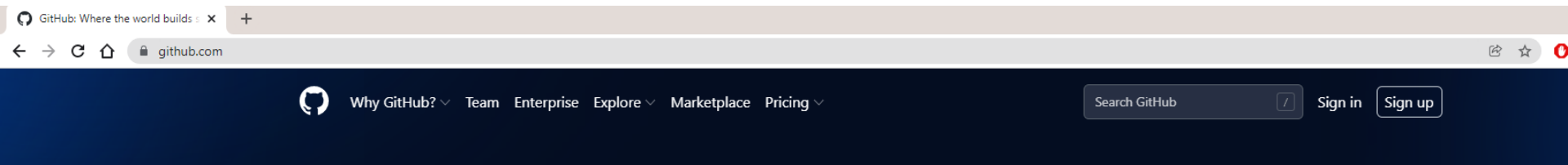
- GitHub is a hosting service for git-based projects
 - **Git** = version control system
 - **GitHub** = tool for working with that version control system that doesn't involve the command line
- Think of GitHub like Google Drive or Dropbox/Box
 - Files stored locally on your computer
 - Sync with Drive/Dropbox/Box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

GitHub

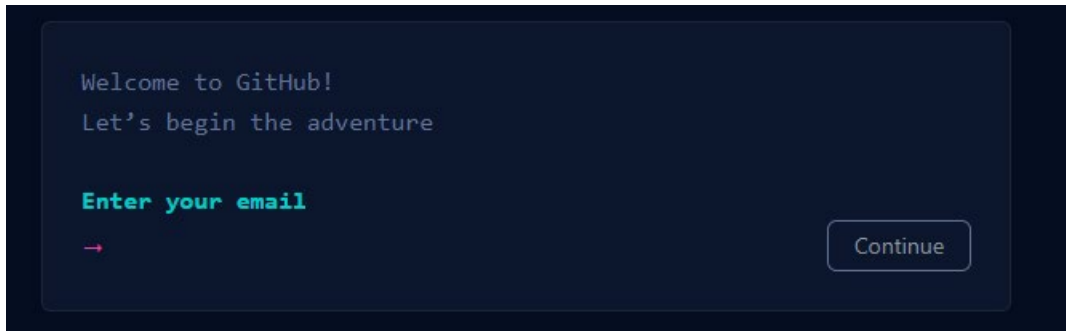
- www.github.com
- Public website
- Individual repositories can be made public or private (only you or invited collaborators can see repo)

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

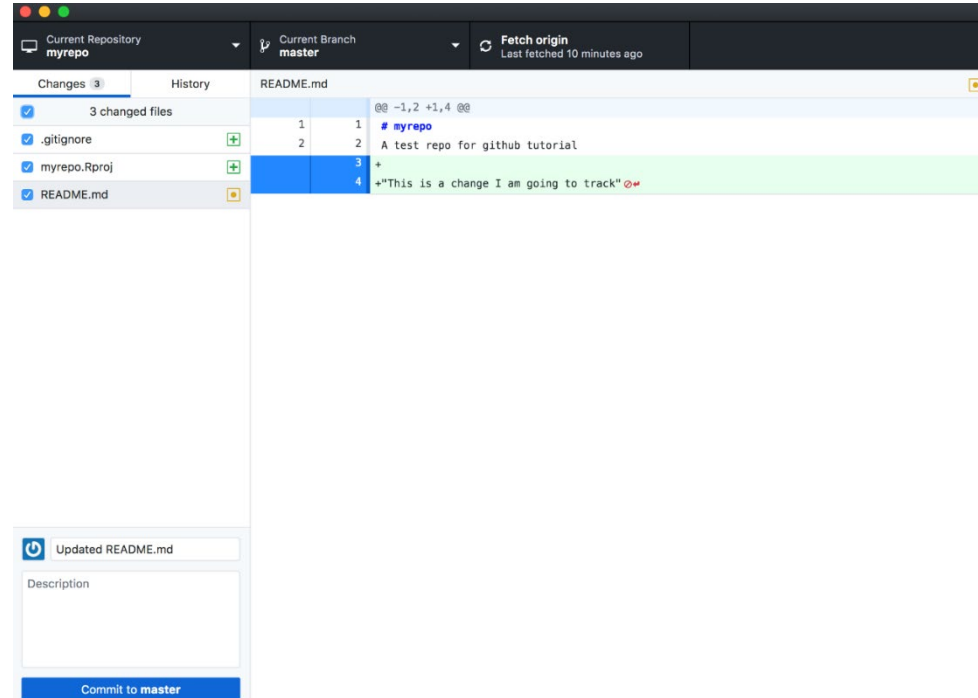
Repo Locations

- **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*

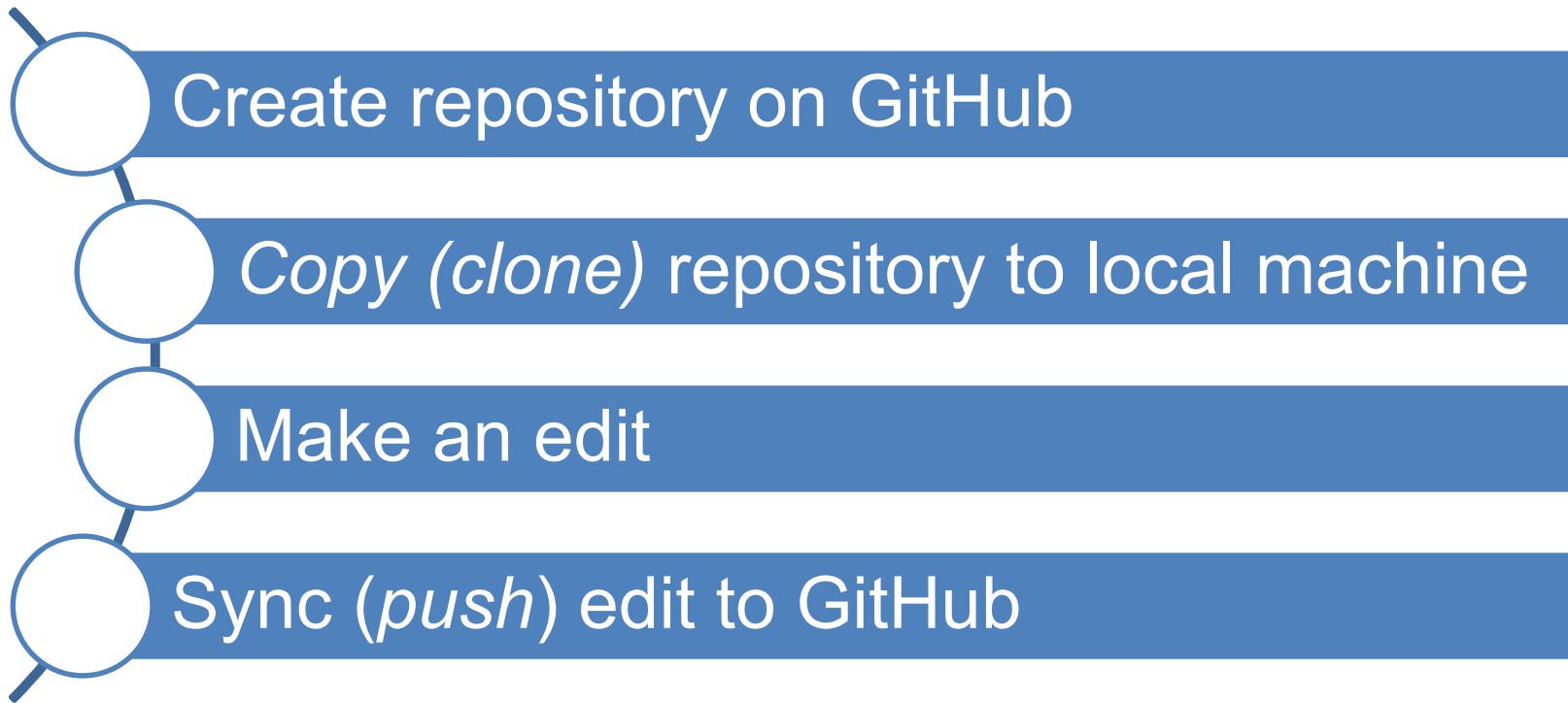
GitHub Training Part II

- 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?



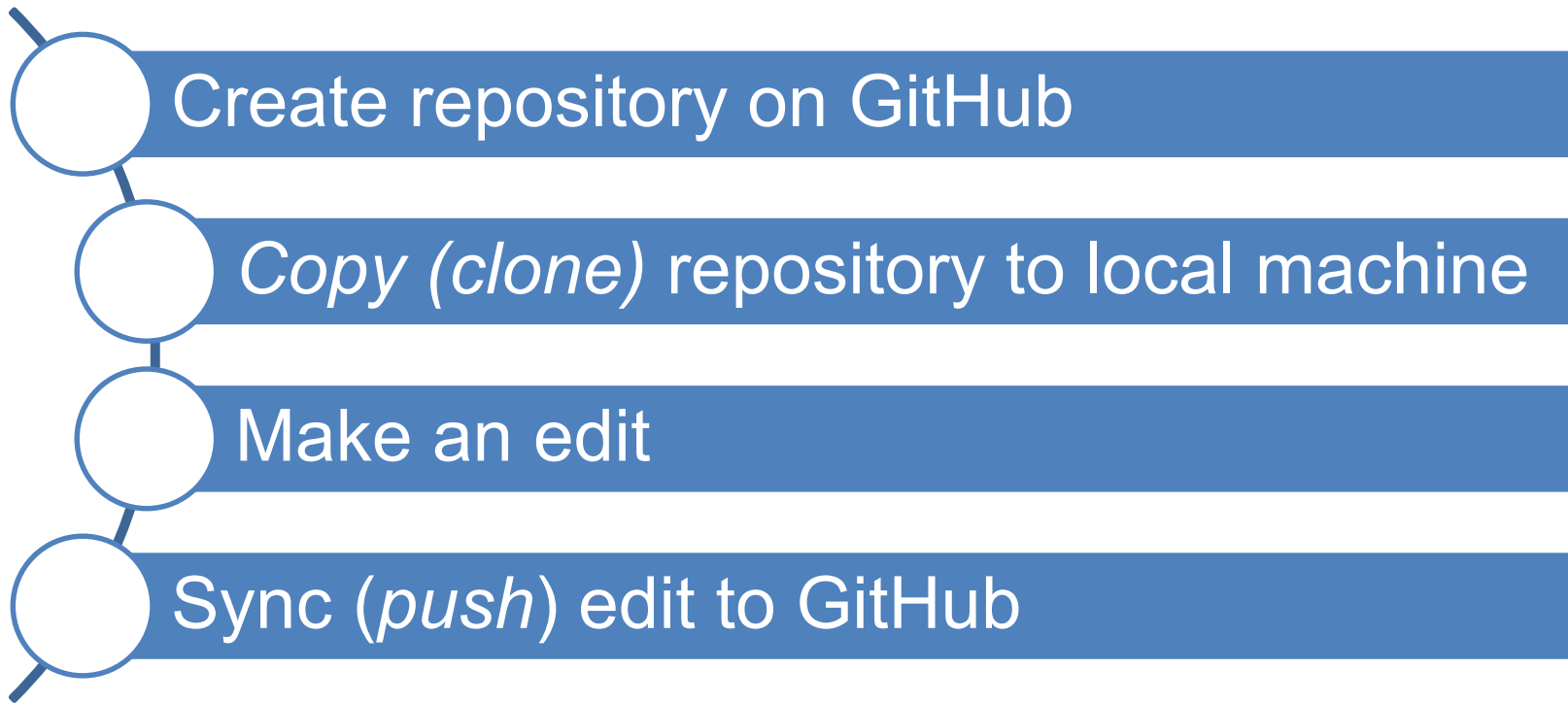
Exercise: Create a Repository



Breakout Rooms



Exercise: Create a Repository




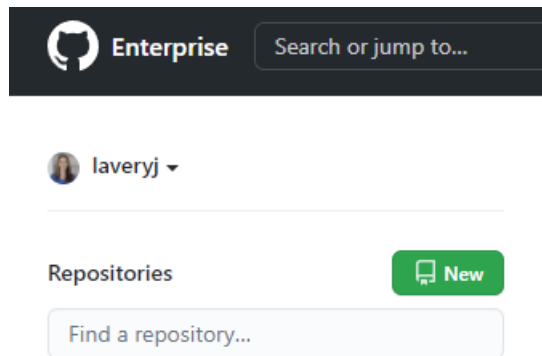
Exercise: Create a Repository



Create repository on GitHub

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 

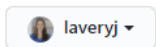


Create a Repository on GitHub

Create a new repository

A repository contains all project files, including the revision history.

Owner *



Repository name *

Great repository names are short and memorable. Need inspiration? How about [improved-system?](#)

Description (optional)

☐

Public

Anyone on the internet can see this repository. You choose who can commit.

☒

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐

Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

Create repository

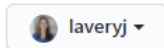
Name your repository something informative. Today we can call this “practice_repo”

Create a Repository on GitHub

Create a new repository

A repository contains all project files, including the revision history.

Owner *



Repository name *



Great repository names are short and memorable. Need inspiration? How about [musical-tribble?](#)

Description (optional)

This is the practice repository from the MSK GitHub training on March 7, 2022.



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

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

Create a new repository

A repository contains all project files, including the revision history.

Owner *



laveryj ▾

Repository name *

practice_repo



Great repository names are short and memorable. Need inspiration? How about [musical-tribble?](#)

Description (optional)

This is the practice repository from the MSK GitHub training on March 7, 2022.



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

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

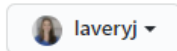
More to come on this later with respect to PHI.

Create a Repository on GitHub

Create a new repository

A repository contains all project files, including the revision history.

Owner *



Repository name *



Great repository names are short and memorable. Need inspiration? How about [musical-tribble?](#)

Description (optional)

This is the practice repository from the MSK GitHub training on March 7, 2022.



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Always add a README.

Can expand on the project description.

Create a Repository on GitHub

Create a new repository

A repository contains all project files, including the revision history.

Owner *



Repository name *



Great repository names are short and memorable. Need inspiration? How about [musical-tribble?](#)

Description (optional)

This is the practice repository from the MSK GitHub training on March 7, 2022.



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: **None** ▼

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

☒ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

This .gitignore template

the default name in your [settings](#).

R

R

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

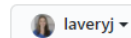
The .gitignore template for R will ignore things like .Rhistory files, .RData files, etc.

Create a Repository on GitHub

Create a new repository

A repository contains all project files, including the revision history.

Owner *



laveryj

Repository name *

practice_repo



Great repository names are short and memorable. Need inspiration? How about [musical-tribble?](#)

Description (optional)

This is the practice repository from the MSK GitHub training on March 7, 2022.



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☒ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: R

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository




Create repo!


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**

-



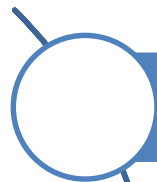
GitHub

**Remote
Repository**

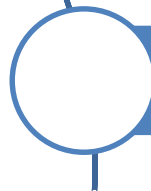
Repository
exists here

Questions / Roadblocks?

Exercise: Create a Repository



Create repository on GitHub



*Copy (**clone**)* repository to local machine

Copying the Repository from GitHub to Your Workstation

🔒 laveryj / practice_repo Private

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

laveryj Initial commit

.gitignore Initial commit

README.md Initial commit

README.md

practice_repo

This is the practice repository from the MSK GitHub training on March 7, 2022.

Go to file

Add file

Code

Clone

HTTPS SSH GitHub CLI

https://github.mskcc.org/laveryj/practice_

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Copying the Repository from GitHub to Your Workstation

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



Initial commit



.gitignore

Initial commit



README.md

Initial commit

Launching GitHub Desktop...

If nothing happens, [download GitHub Desktop](#) and try again.

[Go back](#)

README.md

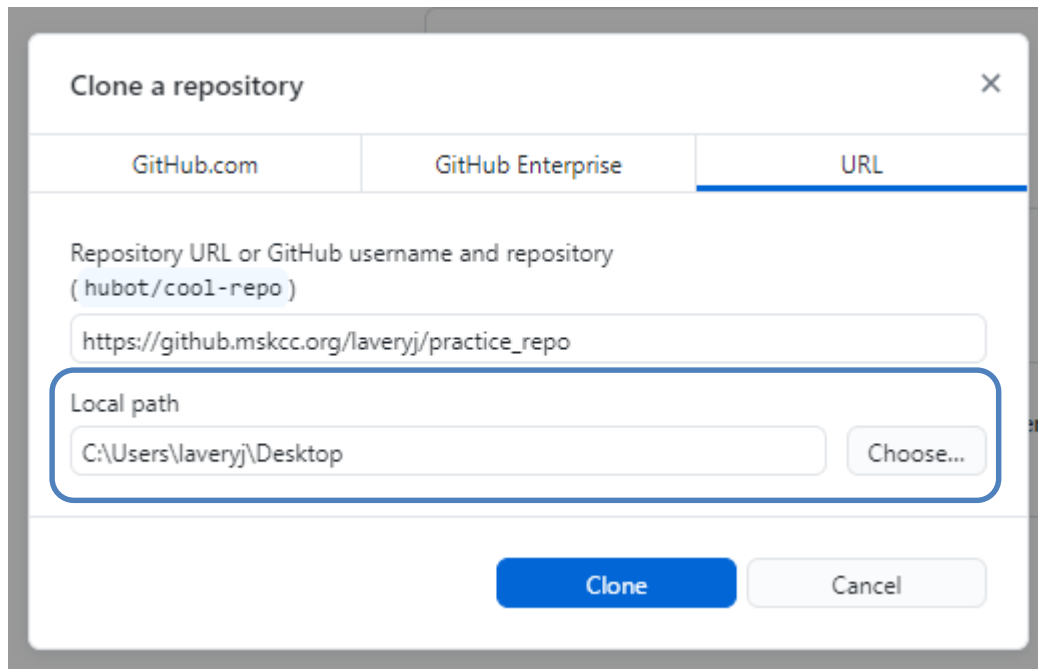


practice_repo

This is the practice repository from the MSK GitHub training on March 7, 2022.

Copying the Repository from GitHub to Your Workstation

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



The screenshot shows the 'Clone a repository' dialog box in GitHub Desktop. The dialog has a title bar with a close button (X). Below the title bar are three tabs: 'GitHub.com', 'GitHub Enterprise', and 'URL'. The 'URL' tab is selected. The main area contains a text input field for the 'Repository URL or GitHub username and repository' with the placeholder text '(hubot/cool-repo)'. The input field contains the URL 'https://github.mskcc.org/laveryj/practice_repo'. Below this is a section for the 'Local path' with a text input field containing 'C:\Users\laveryj\Desktop' and a 'Choose...' button. At the bottom right are two buttons: 'Clone' (blue) and 'Cancel' (gray).

Clone a repository

GitHub.com GitHub Enterprise URL

Repository URL or GitHub username and repository
(hubot/cool-repo)

https://github.mskcc.org/laveryj/practice_repo

Local path

C:\Users\laveryj\Desktop Choose...

Clone Cancel

Current repository
practice_repoCurrent branch
main**Fetch origin**
Never fetched

Changes

History



0 changed files

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 ExplorerRepository menu or **Ctrl** | **Shift** | **F**

Show in Explorer

Open the repository page on GitHub in your browserRepository menu or **Ctrl** | **Shift** | **G**

View on GitHub



Summary (required)

Description

Commit to **main**

Current repository
practice_repoCurrent branch
mainFetch origin
Never fetched

Changes

History



0 changed files

The version on
our workstation
matches the
version on GitHub

No local changes

There are no uncommitted changes in this repository. Here are some handy 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

Changes History

0 changed files

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



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

File Explorer | 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
README.md

2/25/2022 10:53 AM Text Document
2/25/2022 10:53 AM MD File

1 KB
1 KB

Current Status



**Local
Folder**



**Local
Repository**



GitHub

**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

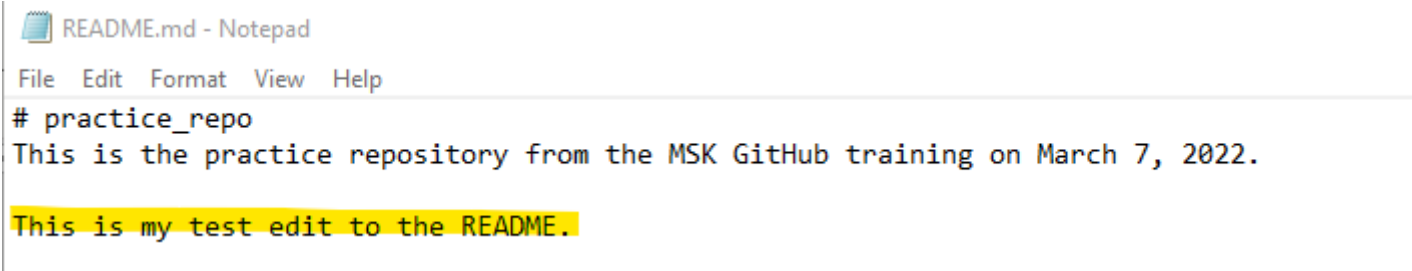
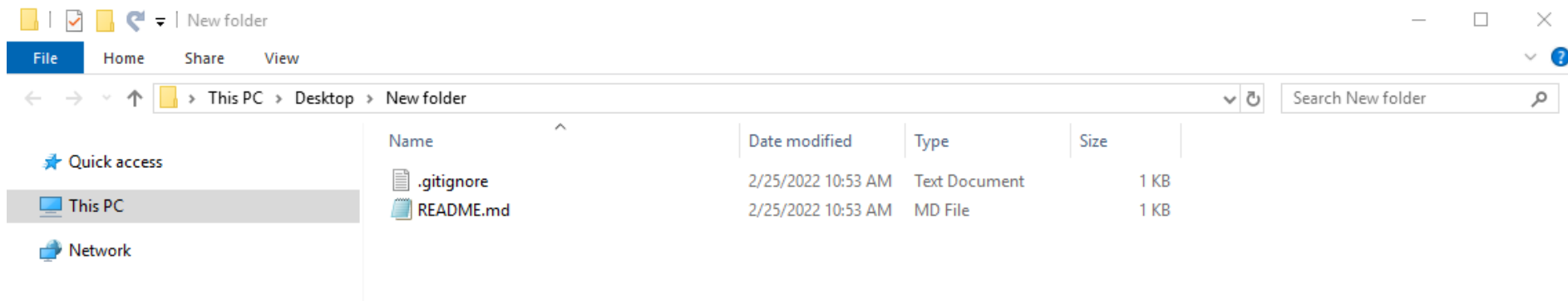
Exercise: Create a Repository

- 
- Create repository on GitHub
 - Copy (clone)* repository to local machine
 - Make an edit

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**

**Modified
repository**



**Local
Repository**

**Original,
unchanged
repository**



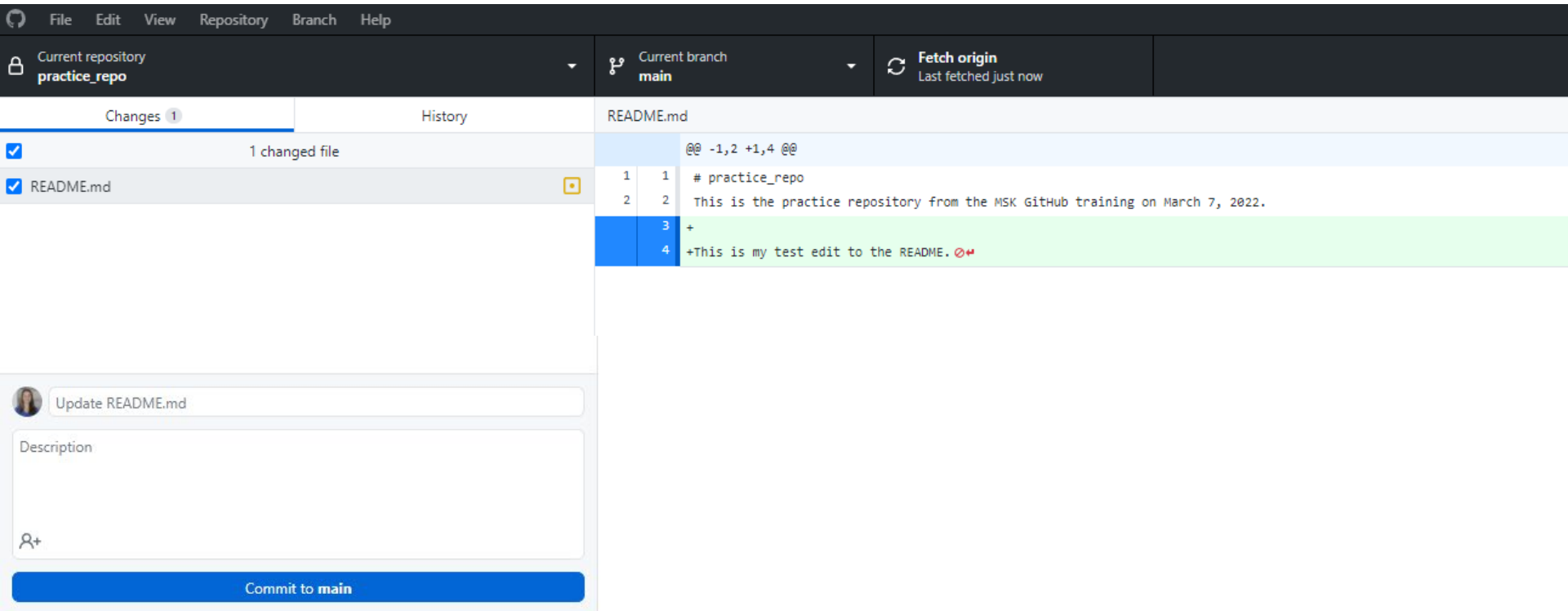
**Remote
Repository**

**Original,
unchanged
repository**

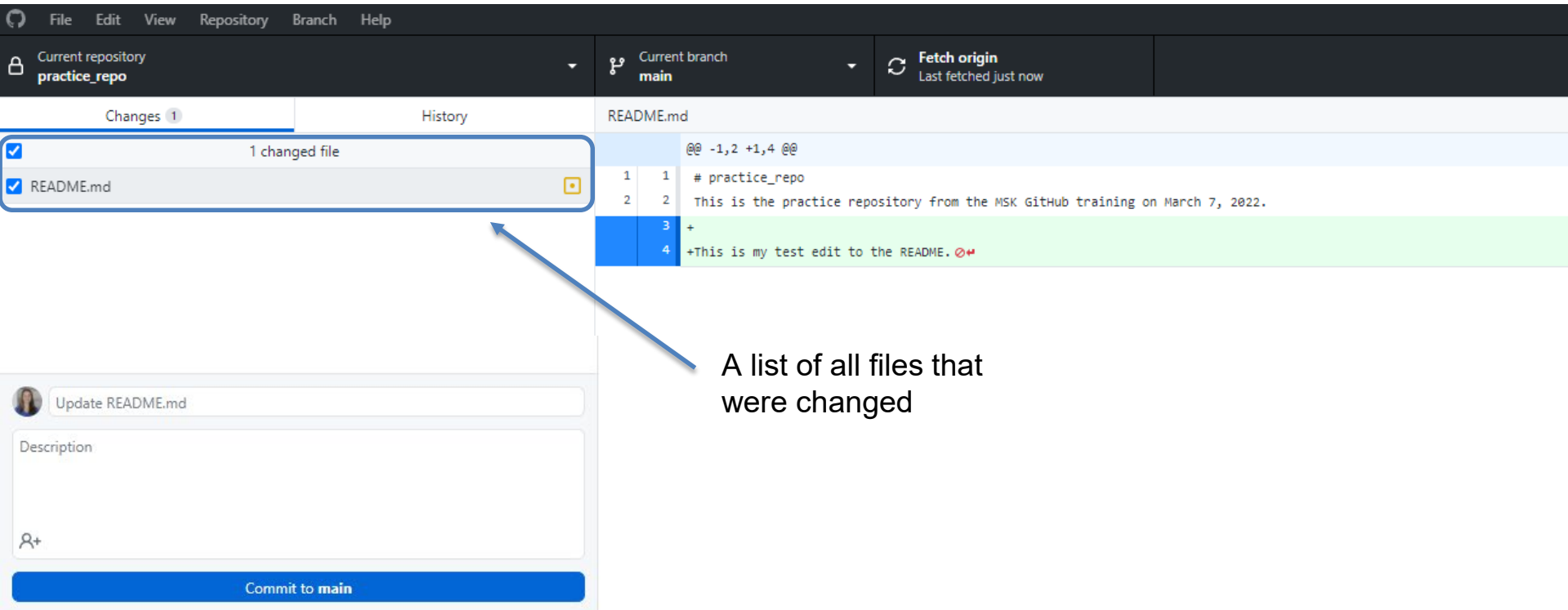
Git Commits

- 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



The screenshot displays the GitHub Desktop application interface. At the top, a dark navigation bar contains the menu items: File, Edit, View, Repository, Branch, and Help. Below this, a status bar shows the current repository as 'practice_repo', the current branch as 'main', and a 'Fetch origin' button indicating the last fetch was just now. The main workspace is divided into three sections. On the left, the 'Changes' tab is active, showing a list of changes with a blue border. It indicates '1 changed file' and lists 'README.md' with a yellow square icon. On the right, the diff view for 'README.md' is shown, displaying line-by-line changes with a blue header and a green background for the new content. The diff shows the addition of a new line: '+This is my test edit to the README.' with a red squiggly line indicating a conflict. At the bottom left, there is a section for updating the README.md file, including a description field and a 'Commit to main' button. A blue arrow points from the text 'A list of all files that were changed' to the 'Changes' tab.

Current repository: **practice_repo**

Current branch: **main**

Fetch origin
Last fetched just now

Changes **1**

History

1 changed file

README.md

@@ -1,2 +1,4 @@

1 1 # practice_repo

2 2 This is the practice repository from the MSK GitHub training on March 7, 2022.

3 +

4 +This is my test edit to the README. ❌

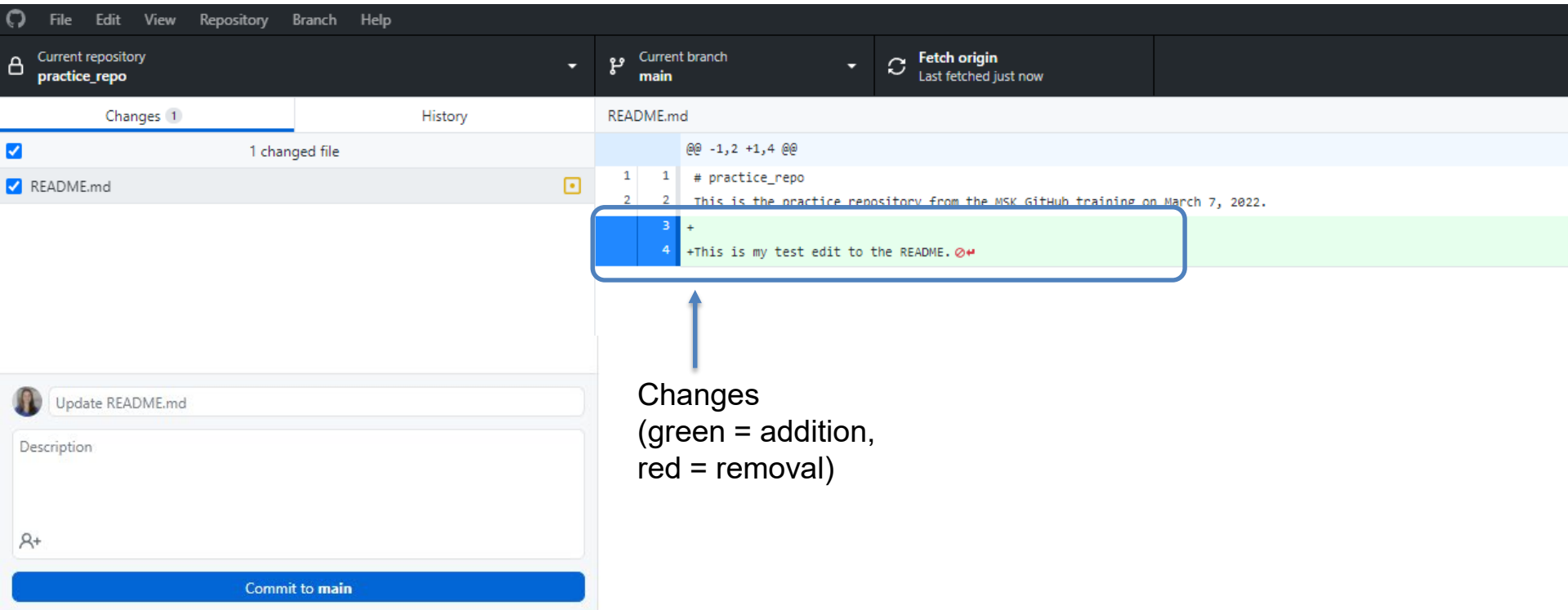
Update README.md

Description

Commit to **main**

A list of all files that were changed

View Repository in GitHub Desktop



The screenshot shows the GitHub Desktop application interface. At the top, the menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu bar, the status bar shows the current repository as 'practice_repo', the current branch as 'main', and a 'Fetch origin' button. The main area is divided into three panes: 'Changes' (showing 1 changed file), 'History', and a diff view for 'README.md'. The diff view shows a comparison between the current state and the previous commit. Line 3 is highlighted in green, indicating an addition, and line 4 is highlighted in red, indicating a removal. A blue box highlights the diff view, and a blue arrow points to it from the text 'Changes (green = addition, red = removal)'. The bottom of the interface shows a commit message input field with 'Update README.md', a description field, and a 'Commit to main' button.

Current repository: **practice_repo**

Current branch: **main**

Fetch origin
Last fetched just now

Changes 1

History

1 changed file

✓ README.md

@@ -1,2 +1,4 @@

1 1 # practice_repo

2 2 This is the practice repository from the MSK Github training on March 7, 2022.

3 +

4 +This is my test edit to the README. ✖

Update README.md

Description

Commit to main

Changes
(green = addition,
red = removal)

View Repository in GitHub Desktop

The screenshot shows the GitHub Desktop application interface. At the top, the menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu bar, the current repository is 'practice_repo' and the current branch is 'main'. The 'Fetch origin' button is available, showing it was last fetched just now. The 'Changes' tab is selected, showing 1 changed file: README.md. The diff view for README.md shows changes from the previous commit. The commit summary box at the bottom left is highlighted with a blue border and contains the text 'Update README.md' and a description field. A blue arrow points from the text 'High level summary of changes, pre-populated when you only change one file' to the commit summary box.

Current repository: **practice_repo**

Current branch: **main**

Fetch origin
Last fetched just now

Changes 1

History

1 changed file

✓ README.md

README.md

@@ -1,2 +1,4 @@

1 1 # practice_repo

2 2 This is the practice repository from the MSK GitHub training on March 7, 2022.

3 +

4 +This is my test edit to the README. 🍴

Update README.md

Description

Commit to main

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

View Repository in GitHub Desktop

Current repository: **practice_repo**

Current branch: **main**

Fetch origin
Last fetched just now

Changes (1)

1 changed file

✓ README.md

History

README.md

@@ -1,2 +1,4 @@

1 1 # practice_repo

2 2 This is the practice repository from the MSK GitHub training on March 7, 2022.

3 +

4 +This is my test edit to the README. 🍌

Update README.md

Description

Commit to main

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

File Edit View Repository Branch Help

Current repository: **practice_repo**

Current branch: **main**

Fetch origin
Last fetched just now

Changes **1** History

1 changed file

✓ README.md

README.md

@@ -1,2 +1,4 @@

```
1 1 # practice_repo
2 2 This is the practice repository from the MSK GitHub training on March 7, 2022.
3 +
4 +This is my test edit to the README. 🍌
```

Update README.md

Description

👤+

Commit to **main**

Commit!

Current Status



**Local
Folder**



**Local
Repository**



GitHub

**Remote
Repository**

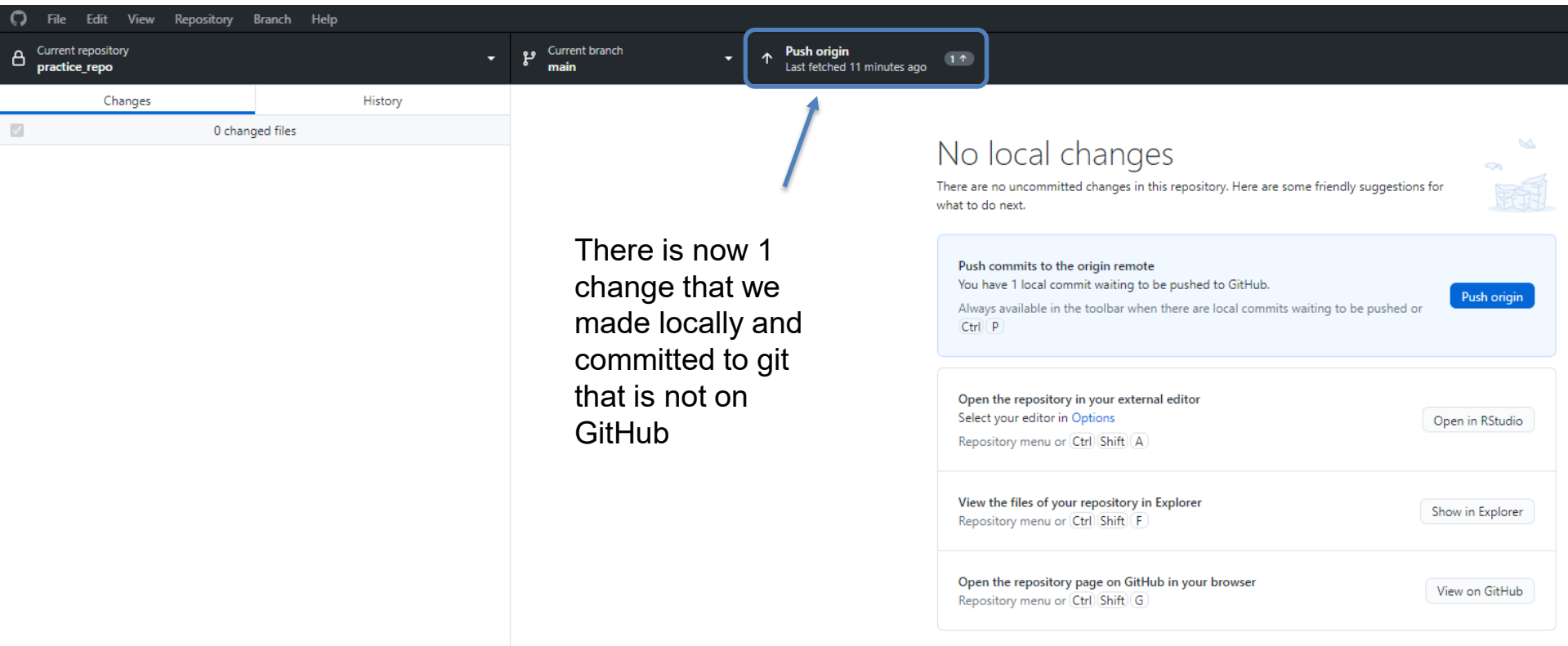
**Modified
repository**

git commit

**Modified
repository**

**Original,
unchanged
repository**

View Repository in GitHub Desktop



The screenshot shows the GitHub Desktop application interface. At the top, there is a dark menu bar with options: File, Edit, View, Repository, Branch, and Help. Below the menu bar, the 'Current repository' is set to 'practice_repo' and the 'Current branch' is 'main'. A blue box highlights the 'Push origin' button, which indicates 'Last fetched 11 minutes ago'. A blue arrow points from the text 'There is now 1 change that we made locally and committed to git that is not on GitHub' to the 'Push origin' button. The left sidebar shows the 'Changes' tab with '0 changed files'. The main area displays a 'No local changes' message and several action buttons: 'Push origin', 'Open in RStudio', 'Show in Explorer', and 'View on GitHub'.

Current repository: **practice_repo**

Current branch: **main**

Push origin
Last fetched 11 minutes ago

Changes | History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

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

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl** **P**

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or **Ctrl** **Shift** **A**

View the files of your repository in Explorer
Repository menu or **Ctrl** **Shift** **F**

Open the repository page on GitHub in your browser
Repository menu or **Ctrl** **Shift** **G**

[Push origin](#)

[Open in RStudio](#)

[Show in Explorer](#)

[View on GitHub](#)

View Repository in GitHub Desktop

The screenshot shows the GitHub Desktop application window. The top menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu, the status bar shows the current repository is 'practice_repo', the current branch is 'main', and the 'Push origin' button is available, indicating the last fetch was 11 minutes ago. The main workspace is divided into two panes: 'Changes' and 'History'. The 'Changes' pane shows '0 changed files'. A blue arrow points from the 'No local changes' message box to the 'Changes' pane.

Current repository: **practice_repo**

Current branch: **main**

Push origin
Last fetched 11 minutes ago

Changes | History

0 changed files

No local changes
There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

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

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl** **P**

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or **Ctrl** **Shift** **A**

View the files of your repository in Explorer
Repository menu or **Ctrl** **Shift** **F**

Open the repository page on GitHub in your browser
Repository menu or **Ctrl** **Shift** **G**

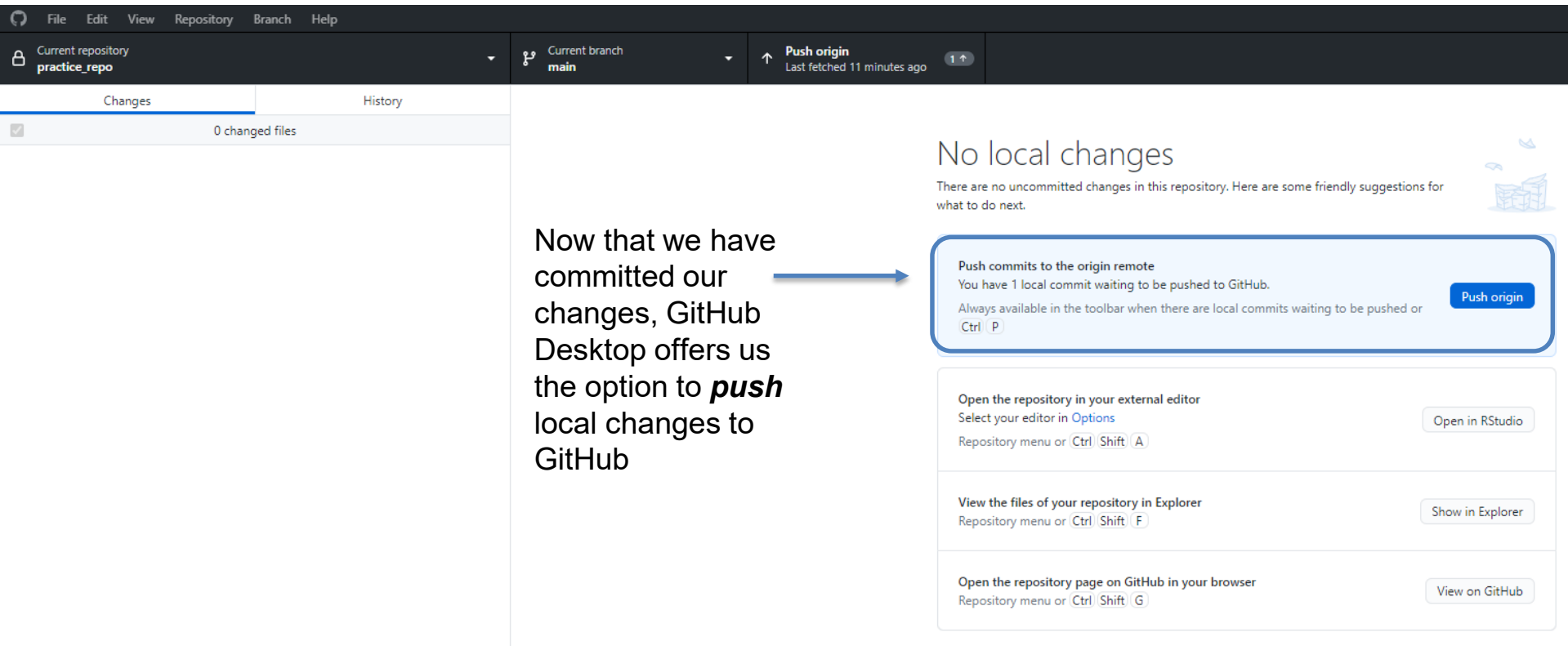
Push origin

Open in RStudio

Show in Explorer

View on GitHub

View Repository in GitHub Desktop



The screenshot shows the GitHub Desktop application interface. The top bar includes a menu (File, Edit, View, Repository, Branch, Help) and status information: 'Current repository: practice_repo', 'Current branch: main', and 'Push origin' (Last fetched 11 minutes ago). The left sidebar shows 'Changes' and 'History' tabs, with 'Changes' selected and displaying '0 changed files'. The main area displays a message: 'No local changes. There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.' A blue box highlights the 'Push origin' button, which is labeled 'Push commits to the origin remote' and 'You have 1 local commit waiting to be pushed to GitHub.' Below this, there are three suggestions: 'Open the repository in your external editor', 'View the files of your repository in Explorer', and 'Open the repository page on GitHub in your browser'. A blue arrow points from the text 'Now that we have committed our changes, GitHub Desktop offers us the option to **push** local changes to GitHub' to the 'Push origin' button.

File Edit View Repository Branch Help

Current repository
practice_repo

Current branch
main

Push origin
Last fetched 11 minutes ago

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl** **P**

Push origin

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**

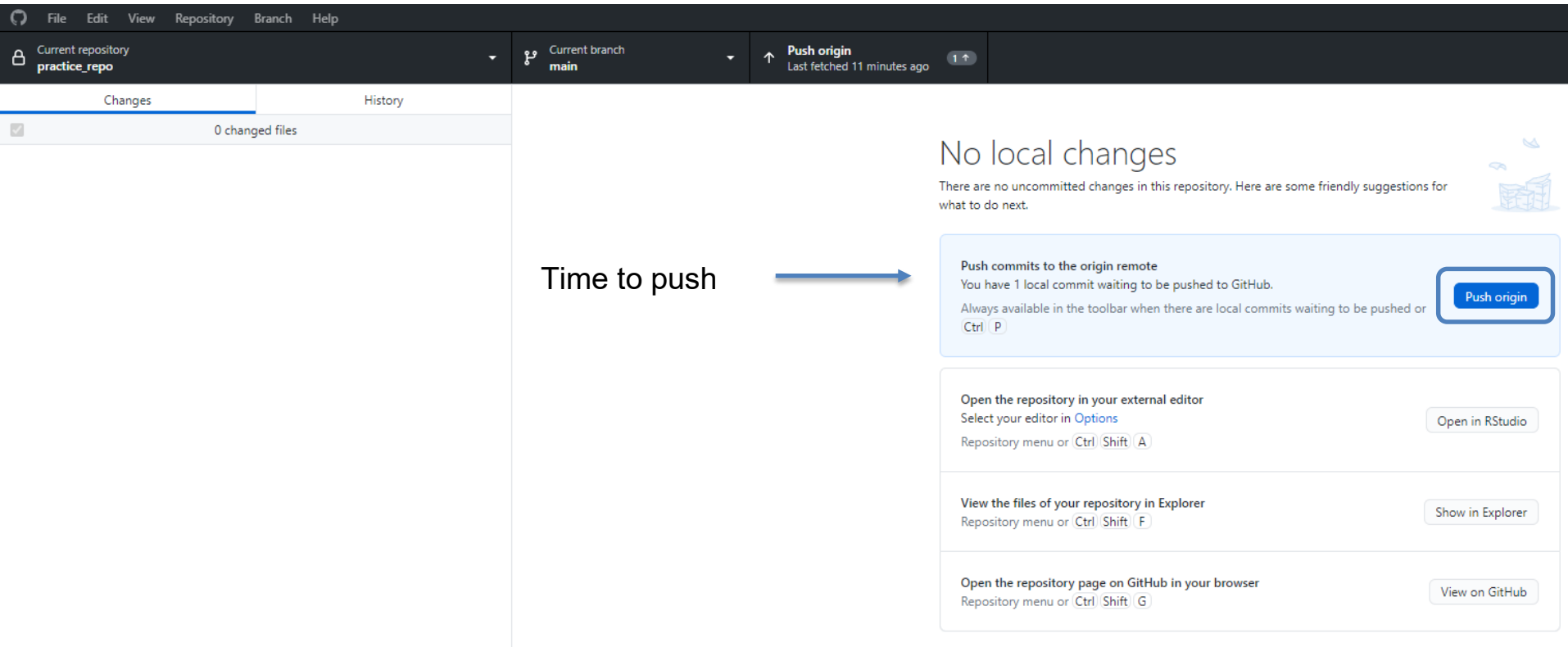
View on GitHub

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

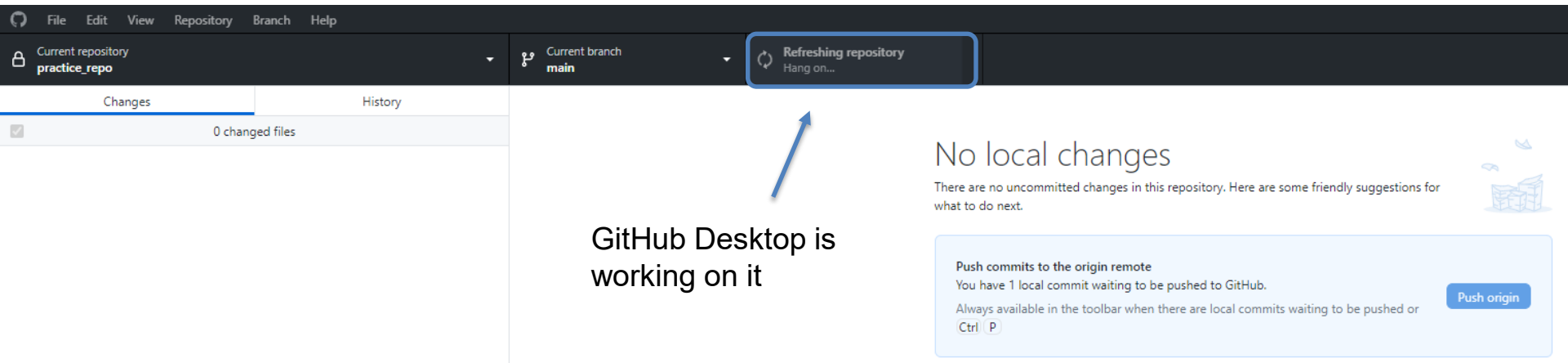


The screenshot shows the GitHub Desktop application interface. The top menu bar includes File, Edit, View, Repository, Branch, and Help. The left sidebar has tabs for Changes and History, with 'Changes' selected, showing '0 changed files'. The main area displays 'No local changes' with a message: 'There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.' Below this, there are three suggestions:

- Push commits to the origin remote**: You have 1 local commit waiting to be pushed to GitHub. Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl P**. A **Push origin** button is visible in the top right of this section.
- Open the repository in your external editor**: Select your editor in [Options](#). Repository menu or **Ctrl Shift A**. An **Open in RStudio** button is visible.
- View the files of your repository in Explorer**: Repository menu or **Ctrl Shift F**. A **Show in Explorer** button is visible.
- Open the repository page on GitHub in your browser**: Repository menu or **Ctrl Shift G**. A **View on GitHub** button is visible.

On the left side of the main area, the text 'Time to push' is followed by a blue arrow pointing towards the 'Push origin' button.

View Repository in GitHub Desktop



View Repository in GitHub Desktop

File Edit View Repository Branch Help

Current repository
practice_repo


Current branch
main

Fetch origin
Last fetched 1 minute ago

Changes

History

0 changed files



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

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

Current Status



**Local
Folder**



**Local
Repository**



GitHub

**Remote
Repository**

**Modified
repository**

git commit

**Modified
repository**

git push

**Modified
repository**

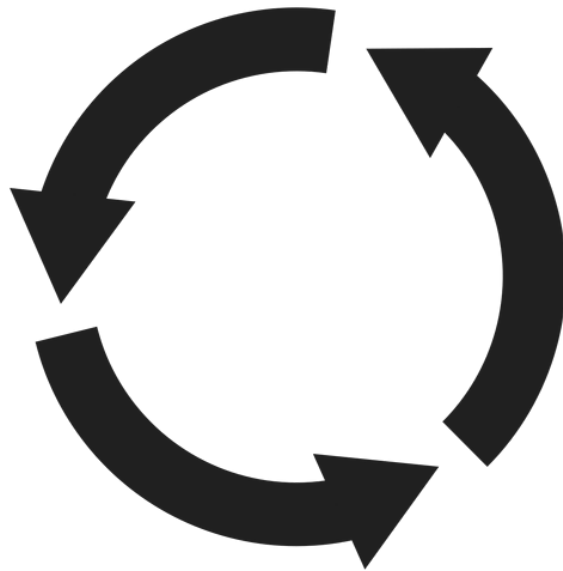
Questions / Roadblocks?

We have:

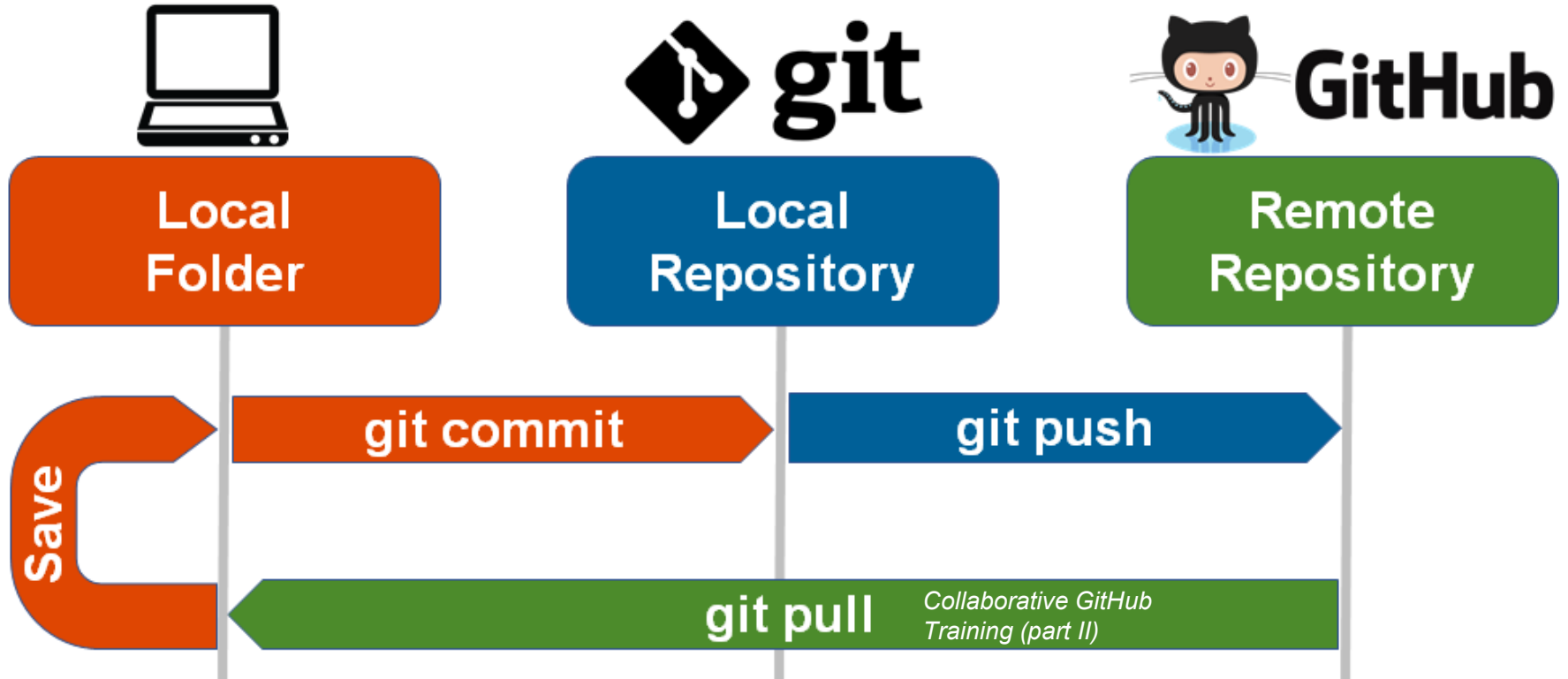
- Created a repository
- Copied that repository from GitHub to our workstation
- Made a change to the local version of the repository (on our workstations)
- Pushed those changes to the remote version of the repository

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?