Introduction to

Git & GitHub

Regina-Mae Dominguez
January 10, 2026

Git/Github

Before we start,

Please make sure you have an account on **github.com**, does *not* have to be an enterprise account.

If you are on a windows computer, please download **git** for windows.

Before we start,

Please make sure you have an account on **github.com**, does *not* have to be an enterprise account.

If you are on a windows computer, please download **git** for windows.

Survey Question

On a scale of 1-5, how familiar are you with linux commands?

Git/Github

LINUX

MIGHT BE USEFUL TODAY

Linux Command Line Cheat Sheet

Directory navigation

cd dir – change directory

cd or cd ~ – change to home

pwd – show current directory

hpc – data processing cluster terminal

List files and directories

ls -l – list files and directories

more file – list content of file

tree – list directory hierarchy

cat file – list content of file

nano -w file – edit file with nano

File and folder manipulations

touch file – create or update file

cat > file – create file

mkdir dir – create a directory

cp file1 file2 – copy file1 to file2

cp -r dir1 dir2 – copy dir1 to dir2;

rm file – delete file

rm -r dir – delete directory

rm -f file – force remove file

rm -rf dir – force remove directory

mv file1 file2 – rename/move file1 to file2

Search files and folders

find - ./data -name report.txt – search folder "data" for file with name report.txt

find - ./ -iname "*report*" – search current folder for a file name containing "report"

find ./ -empty – find empty files and folders

grep -r "text" – recursively search files for word "text"

grep "completeness" report.txt – search for word "completeness" in file

Misc. commands

clear – remove all terminal entries

reset – resets current terminal

du -sh * – list size of files and folders

df -h – list disk usage statistics

apropos – shell version of "Google"

Bash Terminal Shortcuts

Ctrl+C – stop current command

Ctrl+R – search history

Ctrl+Shift+C – Copy

Ctrl+Shift+V – Paste

TAB - autocomplete terminal entry

Disclaimer

Today, we will **not** go over

- forks,
- merge conflicts & resolutions
- stashing
- rebase
- GitHub Desktop (which may be the easier solution for what you need, but would still need to knowledge of foundations of Git)

Goals&/orObjectives

- Make a project folder on your computer, then **push** it to GitHub
- Probably need to make a **key** to allow **push/pull??!**
- Create a new **branch**, open a **pull request**, and merge it into your main branch
- Save your changes as a **commit**
- Undo a change by reverting it
- **Clone** a GitHub repository onto your computer

And learn basic linux commands :)

What is **Git?**

GIt is a distributed **version control system**.

- Runs locally on your computer
- Tracks changes and versions of files for collaboration
- Operate through the command line or a different GUI

What is **GitHub?**

GitHub is the site or service that **hosts** your git files. (remote location)

Why know **Git?**

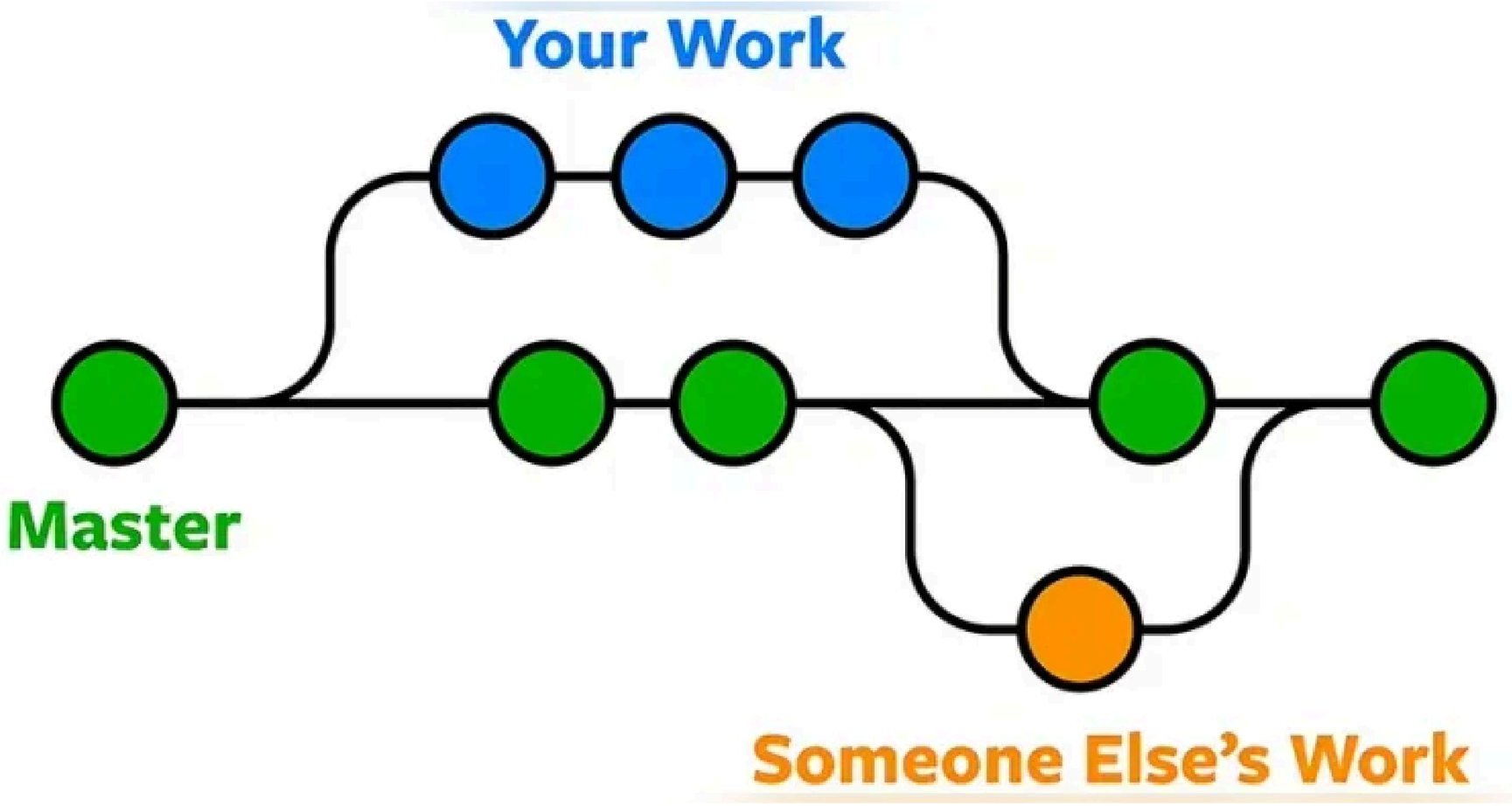
Code should be readable, reliable, and reusable/reproducible.

1. Makes **collaboration** with teammates easier with a standard workflow (workplace standard in most places).
2. Allows you to **build new features safely using branches**, so you don’t accidentally break working code.
3. Git **protects** your project with clear “save points” (commits), so you can **track changes and recover** if something goes wrong.

Quick concepts:

- **Repository (repo):** your project directory (files + history)
- **Commit:** a saved snapshot of your changes
- **Staging:** the “ready to commit” area
- **Branch:** a separate line of work
- **Merge:** combine changes from branches
- **Push / Pull:** upload commits to the remote / download updates to your computer
- **Clone:** copy a remote repo onto your computer

Branches



Exercise 1: Create a SSH key on GitHub.

Mac

1. Open **Terminal** and **run this script replacing with your email:**

```
ssh-keygen -t ed25519 -C "INSERT_EMAIL_HERE"
```

2. Run this script to copy the contents and **paste** it onto a notes/text app (will need for the next step)

```
pbcopy < ~/.ssh/id_ed25519.pub
```

Windows

1. Open **Command Prompt** and **run this script replacing with your email:**

```
type %USERPROFILE%\.ssh\id_ed25519.pub
```

2. Run this script to copy the contents and **paste** it onto a notes/text app (will need for the next step)

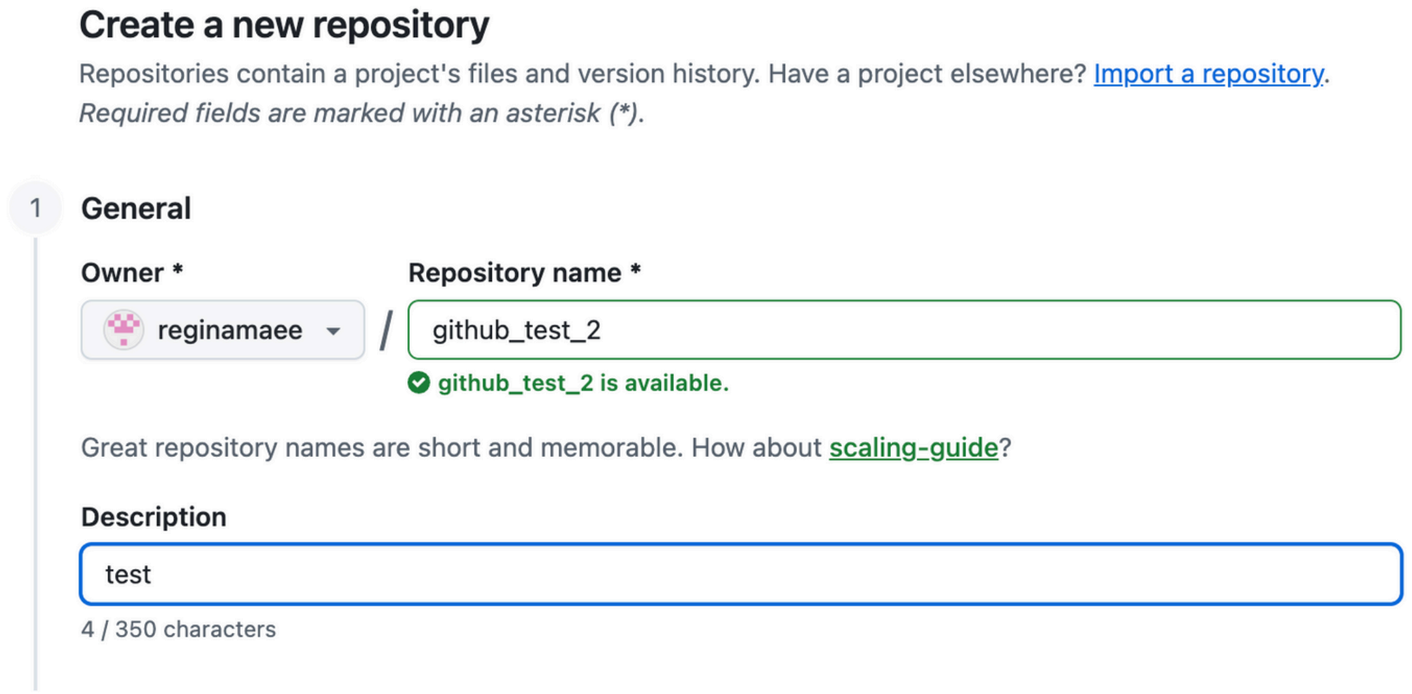
```
type %USERPROFILE%\.ssh\id_ed25519.pub | clip
```

Exercise 1: Create a SSH key on GitHub.

1. Log in to **GitHub**
2. Click your **profile picture** → **Settings** → **SSH and GPG keys**
3. **Click New SSH key** (or Add SSH key)
4. Title: name it something like “**MacBook**” or “**PC**”
5. **Key type**: leave as **Authentication** (most common)
6. **Key: paste your public key (the contents of id_ed25519.pub)**
7. Click **Add SSH key**
8. If prompted, confirm your password / 2FA

Exercise 2. Create a new repo on GitHub

1. Create a new repo on your Github account.
    a. Optional visibility, readme, .gitignore, or license settings



Exercise 2.

2. Create a new repository on command line or prompt

Mac

1. Navigate to the directory in **Terminal** where you want to save this project
2. Run these commands in **Terminal**:

```
git remote add origin *GITHUB LINK HERE*
echo "# github_test_2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git push -u origin main
```

Windows

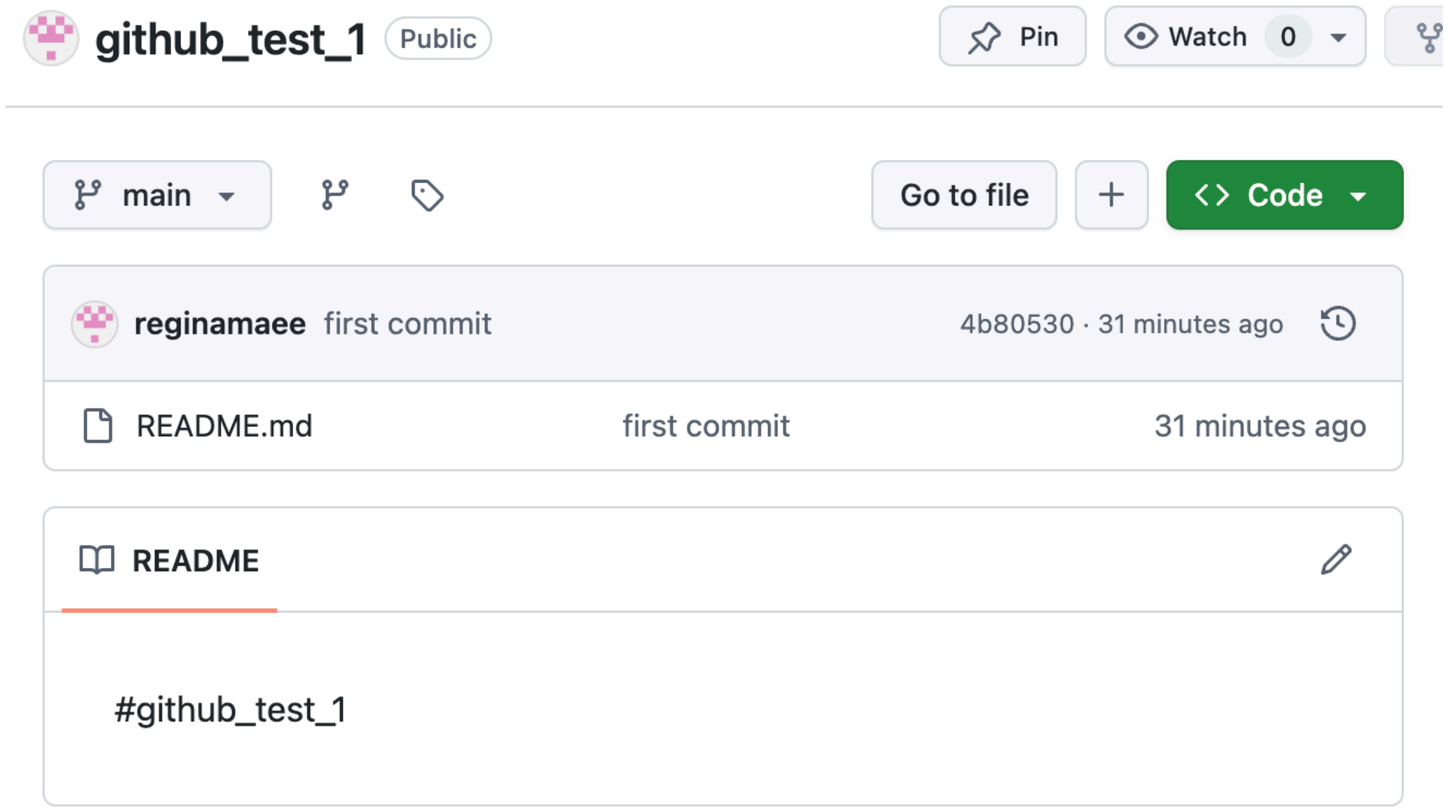
1. Navigate to the directory in **Powershell** where you want to save this project
2. Run these commands in **Powershell.**

```
git remote add origin *GITHUB LINK HERE*
echo "# github_test_2" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git push -u origin main
```

Exercise 2

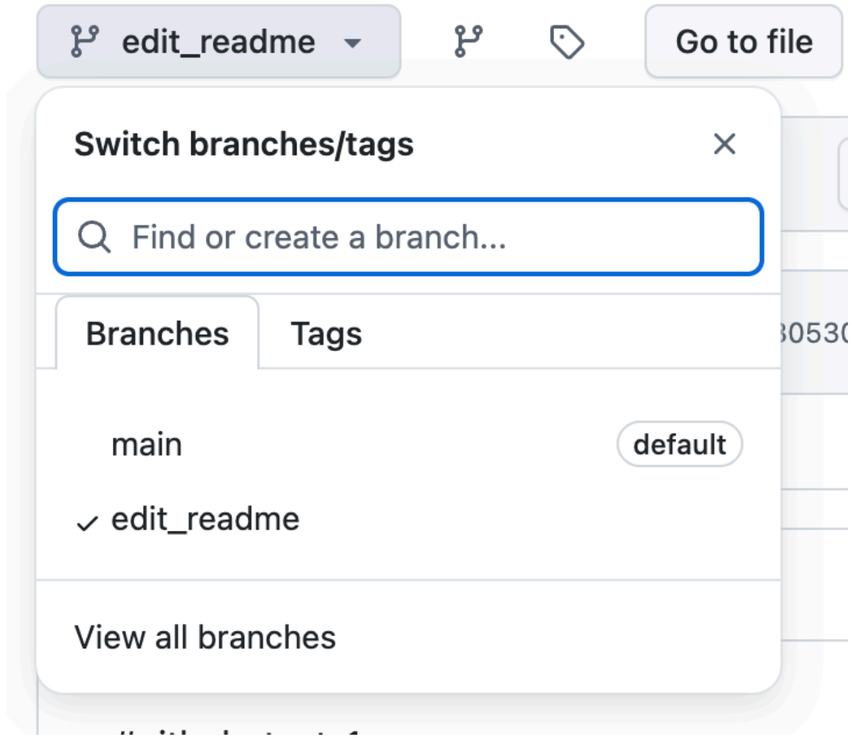
Create a repo on Github

Exercise 2.



Exercise 3. Create a new branch

2. Click **main** to open branch options and create branch “edit_readme”



Exercise 3. Create a new branch & change branches locally

Mac

1. Navigate to the directory where you saved this project locally and run these commands in **Terminal**:

```
git pull
git checkout edit_readme
```

Windows

1. Navigate to the directory where you saved this project locally and run these commands in **Powershell**:

```
git pull
git checkout edit_readme
```

Exercise 4

Edit README.md

Exercise 4. Edit README

1. Open the README.md in a text editor **and add your name in Line 2.** Save these edits.
2. Back in **Terminal/Powershell,** naviggate to the directory where your README is saved and run `git status`

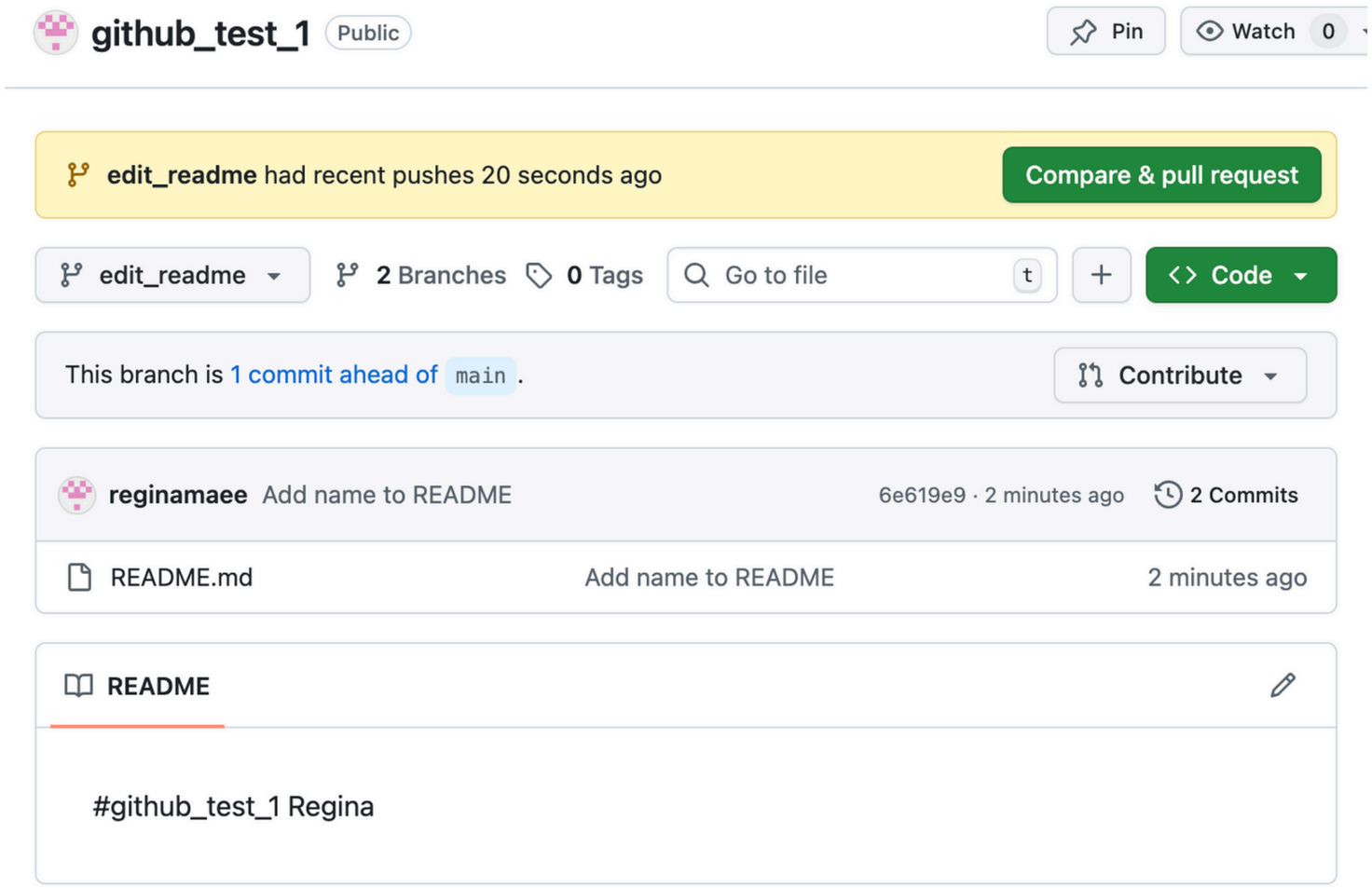
```
On branch edit_readme
Your branch is up to date with 'origin/edit_readme'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
```

3. Run `git commit -m “Add name to README”`
4. Push the edits to GitHub by running: `git push`

Exercise 4

Edit README.md

Exercise 4. Edit README

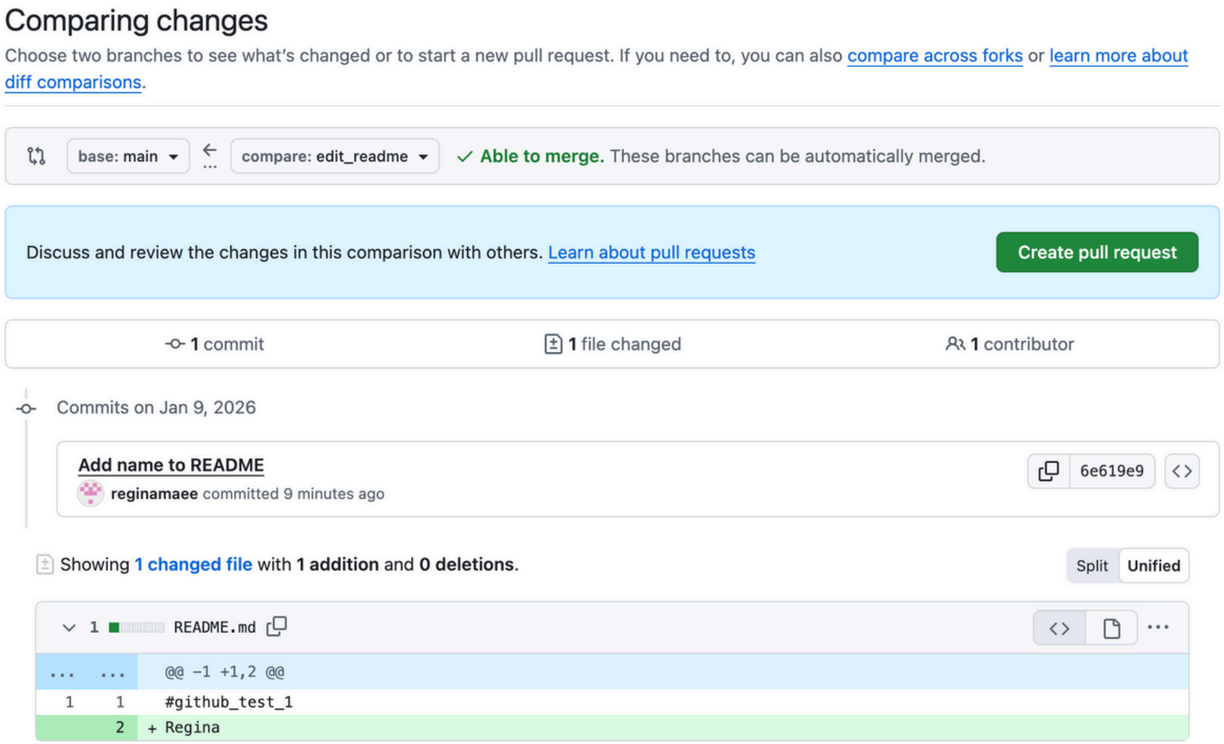


Exercise 5

Review Changes

Exercise 5. Review changes & create pull request

1. On **GitHub,** navigate to the **Pull requests tab**
2. Click **new pull request**
3. Change **compare** branch to **edit_readme**



Exercise 5. Review changes & create pull request

4. Review changes and **create pull request**
5. **Merge pull request**
6. **Changes should now appear in the main branch!!**

Exercise 6: Clone an existing repo

1. Navigate to: https://github.com/reginamaee/intro_to_git
2. Click the **green Code button** and copy the **https** link
3. Navigate to a directory on **Terminal/Powershell** to where **you want to save this project locally and run:**

```
git clone *THE COPIED HTTPS LINK*
```