

# Introduction to **Git & GitHub**

Regina-Mae Dominguez  
January 10, 2026

# 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?



# Linux Command Line Cheat Sheet

# LINUX

MIGHT BE USEFUL TODAY

Git/Github

Directory navigation	File and folder manipulations
<b>cd dir</b> – change directory	<b>touch file</b> – create or update file
<b>cd</b> or <b>cd ~</b> – change to home	<b>cat &gt; file</b> – create file
<b>pwd</b> – show current directory	<b>mkdir dir</b> – create a directory
<b>hpc</b> – data processing cluster terminal	<b>cp file1 file2</b> – copy file1 to file2
<b>cp -r dir1 dir2</b> – copy dir1 to dir2;	
<b>rm file</b> – delete file	
<b>more file</b> – list content of file	<b>rm -r dir</b> – delete directory
<b>tree</b> – list directory hierarchy	<b>rm -f file</b> – force remove file
<b>cat file</b> – list content of file	<b>rm -rf dir</b> – force remove directory
<b>nano -w file</b> – edit file with nano	<b>mv file1 file2</b> – rename/move file1 to file2
List files and directories	Search files and folders
<b>ls -l</b> – list files and directories	<b>find ./data -name report.txt</b> – search folder "data" for file with name report.txt
<b>more file</b> – list content of file	<b>find ./ -iname "*report*</b> " – search current folder for a file name containing "report"
<b>tree</b> – list directory hierarchy	<b>find ./ -empty</b> – find empty files and folders
<b>cat file</b> – list content of file	<b>grep -r "text"</b> – recursively search files for word "text"
<b>nano -w file</b> – edit file with nano	<b>grep "completeness" report.txt</b> – search for word "completeness" in file
Search files and folders	
Misc. commands	Bash Terminal Shortcuts
<b>clear</b> – remove all terminal entries	<b>Ctrl+C</b> – stop current command
<b>reset</b> – resets current terminal	<b>Ctrl+R</b> – search history
<b>du -sh *</b> – list size of files and folders	<b>Ctrl+Shift+C</b> – Copy
<b>df -h</b> – list disk usage statistics	<b>Ctrl+Shift+V</b> – Paste
<b>apropos</b> – shell version of "Google"	<b>TAB</b> – 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 knowledege 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.

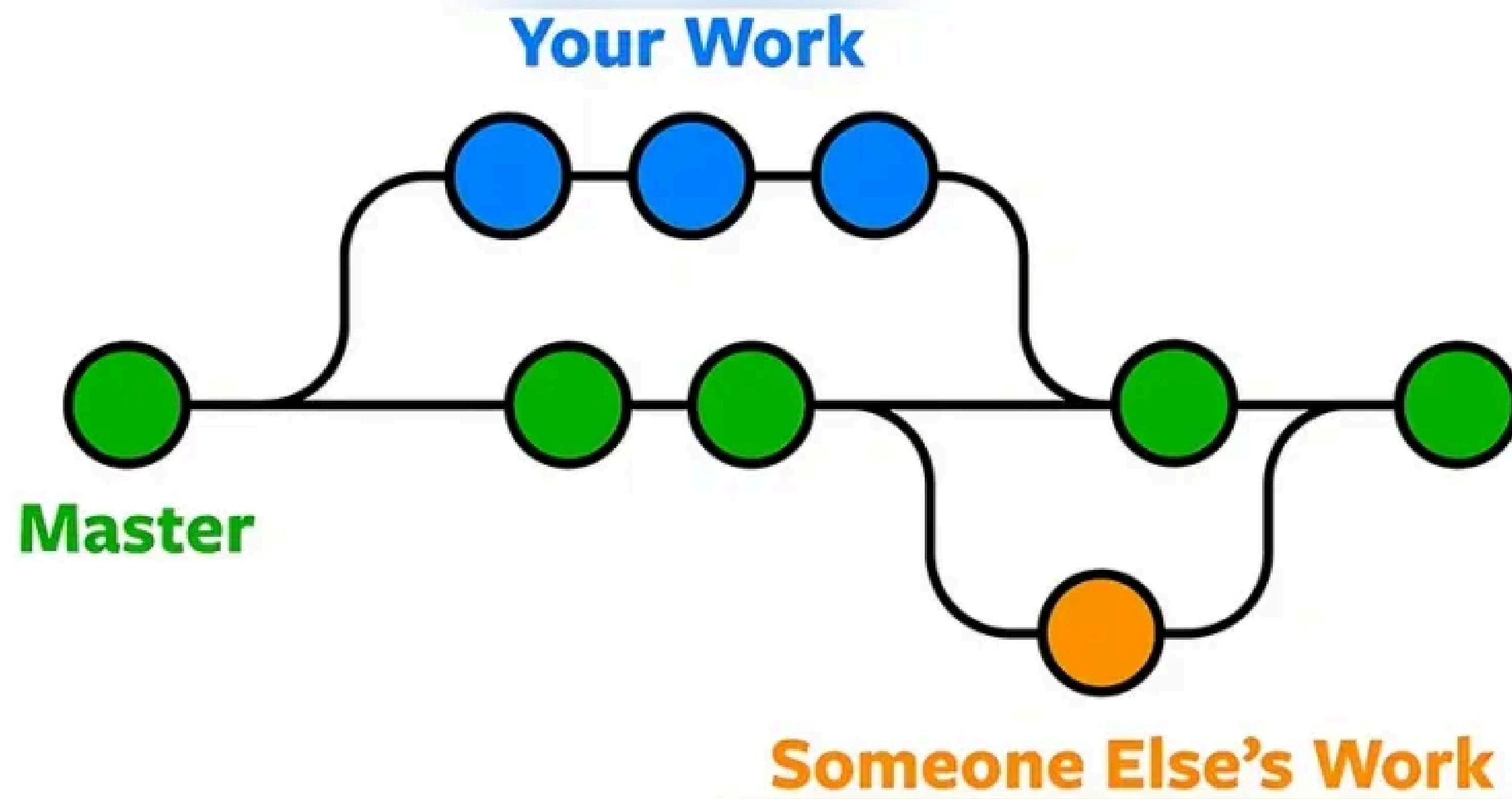
1. Makes **collaboration** with teammates easier with a standard workflow (workplace standard in most places).
2. Allows you to **build new features safely using branches**, so you don't accidentally break working code.
3. Git **protects** your project with clear "save points" (commits), so you can **track 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

Git/Github



# 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

## Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

1

### General

Owner \*



Repository name \*

github\_test\_2

**github\_test\_2 is available.**

Great repository names are short and memorable. How about [scaling-guide](#)?

### Description

test

4 / 350 characters

# 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
```

Create a repo on  
Github

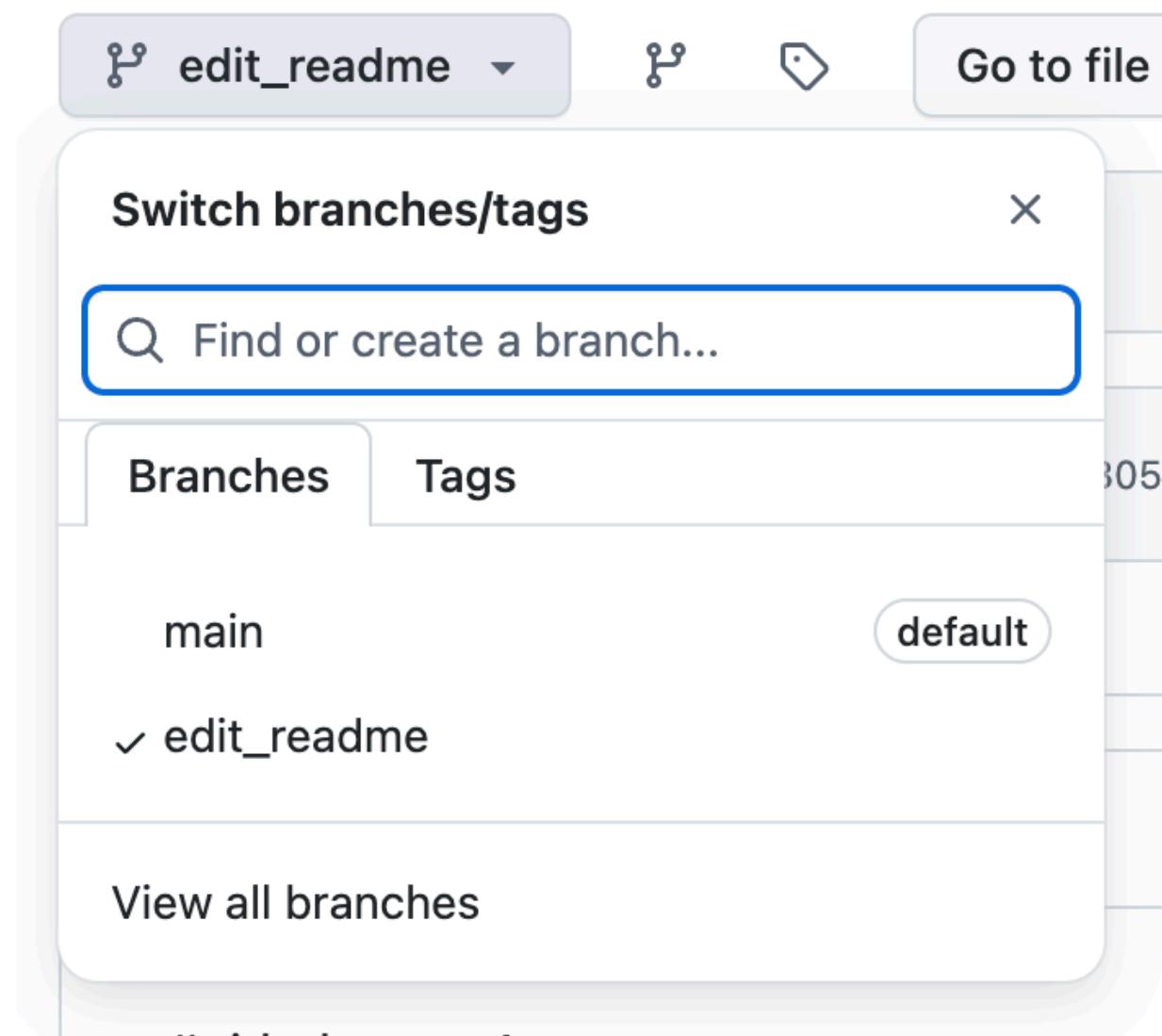
Exercise 2

# Exercise 2.

The screenshot shows a GitHub repository page for 'github\_test\_1'. The repository is public. At the top right, there are buttons for 'Pin', 'Watch' (with 0 notifications), and a gear icon. Below the repository name, there are buttons for 'main' (with a dropdown arrow), a branch icon, and a tag icon. To the right are buttons for 'Go to file', a plus sign, and a green 'Code' button with a dropdown arrow. The main area displays a commit from 'reginamaee' with the message 'first commit' and timestamp '4b80530 · 31 minutes ago'. Below it is a file entry for 'README.md' with the message 'first commit' and timestamp '31 minutes ago'. The 'README' file content is shown as '#github\_test\_1' with an edit icon.

# 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

1. Open the README.md in a text editor  
**and add your name in Line 2.** Save  
these edits.

2. Back in **Terminal/Powershell**,  
navigate 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`

## Edit README.md

## Exercise 4

# Exercise 4. Edit README

The screenshot shows a GitHub repository page for a public repository named `github_test_1`. The repository has 2 branches and 0 tags. A recent push was made 20 seconds ago by `edit_readme`. The main branch is 1 commit ahead of the `main` branch. The `README.md` file contains the text `#github_test_1 Regina`.

github\_test\_1 Public

edit\_readme had recent pushes 20 seconds ago

Compare & pull request

edit\_readme 2 Branches 0 Tags Go to file t + Code

This branch is 1 commit ahead of main.

Contribute

reginamaee Add name to README 6e619e9 · 2 minutes ago 2 Commits

README.md Add name to README 2 minutes ago

README

#github\_test\_1 Regina

# 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**

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: main ▾ ← compare: edit\_readme ✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) Create pull request

-o 1 commit    1 file changed    1 contributor

-o Commits on Jan 9, 2026

Add name to README  
reginamaee committed 9 minutes ago

Showing 1 changed file with 1 addition and 0 deletions.

Split Unified

1 README.md @@ -1 +1,2 @@ #github\_test\_1 + Regina

# 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](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*
```