

Git It: Intro to Git and GitHub

Nicole Cote, 2024

What We'll Cover

- Introductions
- A conceptual intro to Git and GitHub
- What is version control?
- Related command line basics and vocab
- A basic Git/GitHub workflow
- Markdown basics
- Trying it out: Getting started by creating a new repository, making commits, & etc.
- Additional resources and where to go next
- Questions

Important Reminder:

If you wish to follow along with the applied portion of this workshop, please be sure you have completed the below, as needed.

- Create a free account on GitHub (and please be logged in/have login details handy!)
- Download Git for Mac or Windows
- Download a free text editor (i.e. Visual Studio Code)

NOTE: It is also perfectly fine to just hang out, follow-along with what's on my screen, see if you are interested, and take any notes you need—these applied concepts can always be completed after, if preferred!

Introductions

Let's Get to Know Each Other.... Please share:

- Preferred name
- Pronouns
- Program
- A “boring” fact about you (or a “fun” fact if you prefer)
- Anything else you'd like (background, interests, what you hope to get out of the session, or anything else)

Git *and/vs* GitHub

What is Version Control?

Why learn these tools?

Example Projects

GitHub Vocab and Command Line Basics

Some Basic GitHub Related Vocab

- Repository (or “repo”)
- Commit
- Push/Pull
- Clone
- Markdown
- README

Some Additional GitHub Vocab

- Fork
- Branch
- Pull Request
- Merge Conflict
- Issue

Let's See An Example Repo

Common Terminal and Git Commands

- `cd` (change directory)
- `clear` (clear what's visible in Terminal)
- `ls` (list contents of folder)
- `git init` (creates git directory)
- `git add` ("stages" your changes)
- `git commit -m 'Type Message'` (commits your changes with a message < 50 characters)
- `git status` (shows current status of your repo)
- `git help` (shows the help feature)
- `git diff` (shows changes)
- `git push origin main` (pushes local changes to the main branch of your GitHub repo)
- `pwd` (print working directory, i.e. the folder you are in currently)

Common Terminal and Git Commands

- **cd (change directory)**
- clear (clear what's visible in Terminal)
- ls (list contents of folder)
- git init (creates git directory)
- **git add ("stages" your changes)**
- **git commit -m 'Type Message' (commits your changes with a message < 50 characters)**
- **git status (shows current status of your repo)**
- git help (shows the help feature)
- git diff (shows changes)
- **git push origin main (pushes local changes to the main branch of your GitHub repo)**
- pwd (print working directory, i.e. the folder you are in currently)

A Basic Git/GitHub Workflow

Basic Git/GitHub Process

1. code/write something
2. open the Terminal
3. cd to your directory
4. git status
5. git add <filename> (alternatively, to add everything you've changed do: git add --all)
6. git commit - m 'write message'
7. optional: do another git status (just to double check it)
8. git push origin main

Let's See An Example Workflow

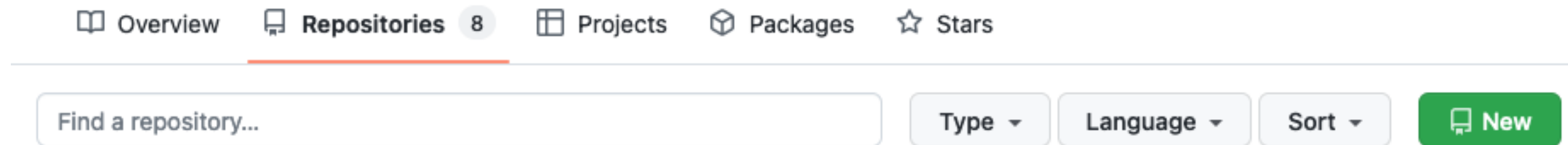
Markdown Basics

Any Qs?

Trying it Out!

Making a New GitHub Repository

Making a New Repo in GitHub Pt 1



- In GitHub, navigate to Repositories (on the top bar)
- Click New

Making a New Repo in GitHub Pt 2

- Fill in details (“Repository Name,” “Description,” and choose public vs private)
- Click “Add a README file” (optional but recommended)
- Click “Create repository”

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

Repository name *

— /

git-introduced



Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-robot](#)?

Description (optional)

This is a repo for an introductory workshop on Git and Github



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 ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

License: None ▾

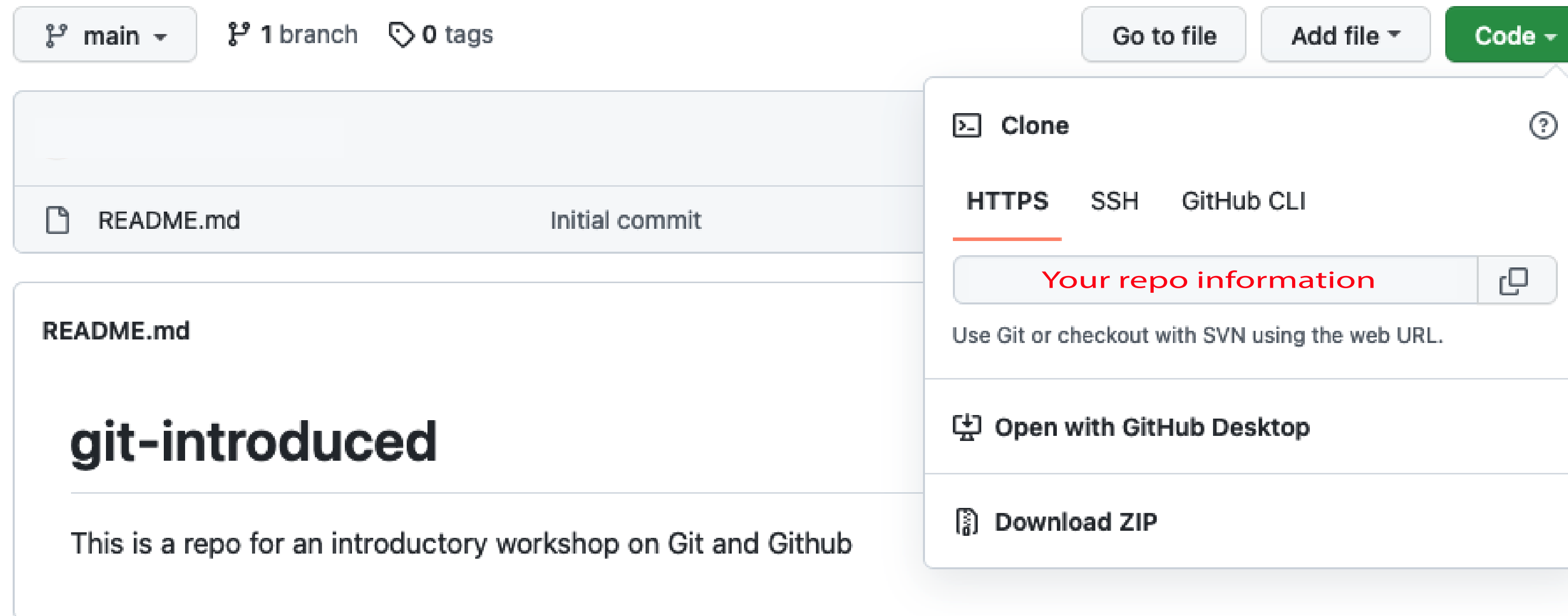
This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Making a New Repo in GitHub Pt 3

- In the Repo click Code and copy the url:



Making a New Repo in GitHub Pt 4

- This last step is to make a local version of the repository you created on GitHub.
- In the Command Line (i.e. the Terminal) do the following:

\$ cd <directory where you want the repo>

FOR EXAMPLE: if you make a folder called Github in your Documents folder, and put it in there it would look something like: \$ cd /Users/username/Documents/Github

NOTE: on a Mac you can ctrl-click on a folder and then hold down option and you get a choice to copy the folder path, which makes this much easier! You can also drag and drop a folder into the Terminal to get the file path!

\$ git clone <paste the url from Github>

\$ cd <name of folder, i.e. repository name>

\$ open .

NOTE: The last two steps are optional—not a required part of making a repo. Alternatively, just navigate to, and open, the folder however is preferable.

**Friendly Reminder:
Don't Actually Type the \$**

Recommended Next Steps - Option 1

1. In the local repo: edit the README.md and/or create a new Markdown file
2. open the Terminal
3. cd to your directory
4. git status
5. git add <filename> (alternatively, to add everything you've changed do: git add --all)
6. git commit - m 'write message'
7. optional: do another git status
8. git push origin main
9. view your edits on GitHub!

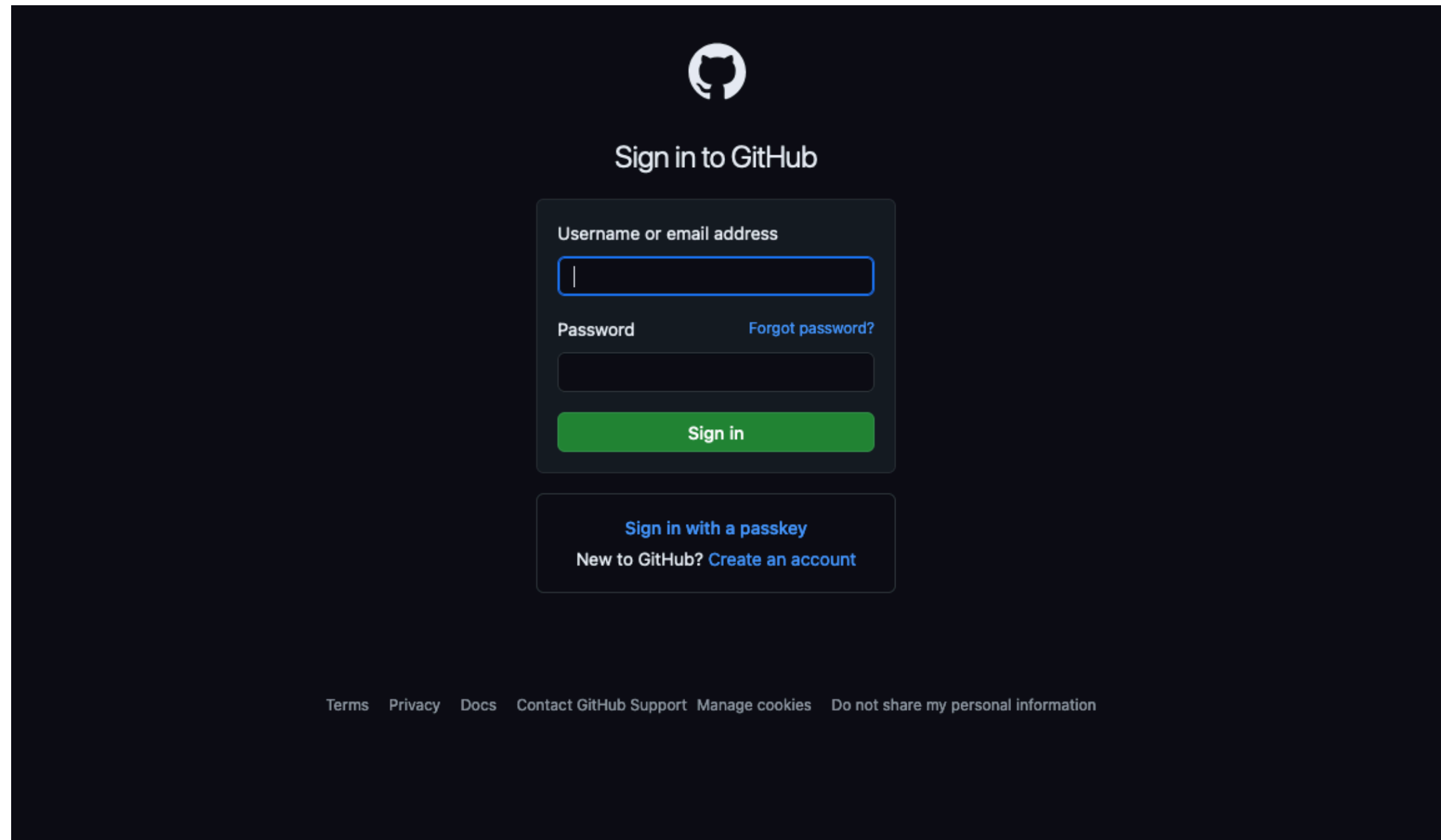
Recommended Next Steps - Option 2

1. On GitHub: edit the README.md and/or create a new Markdown file
2. Commit the changes and add a commit message on Github
3. View your edits and the version differences GitHub!

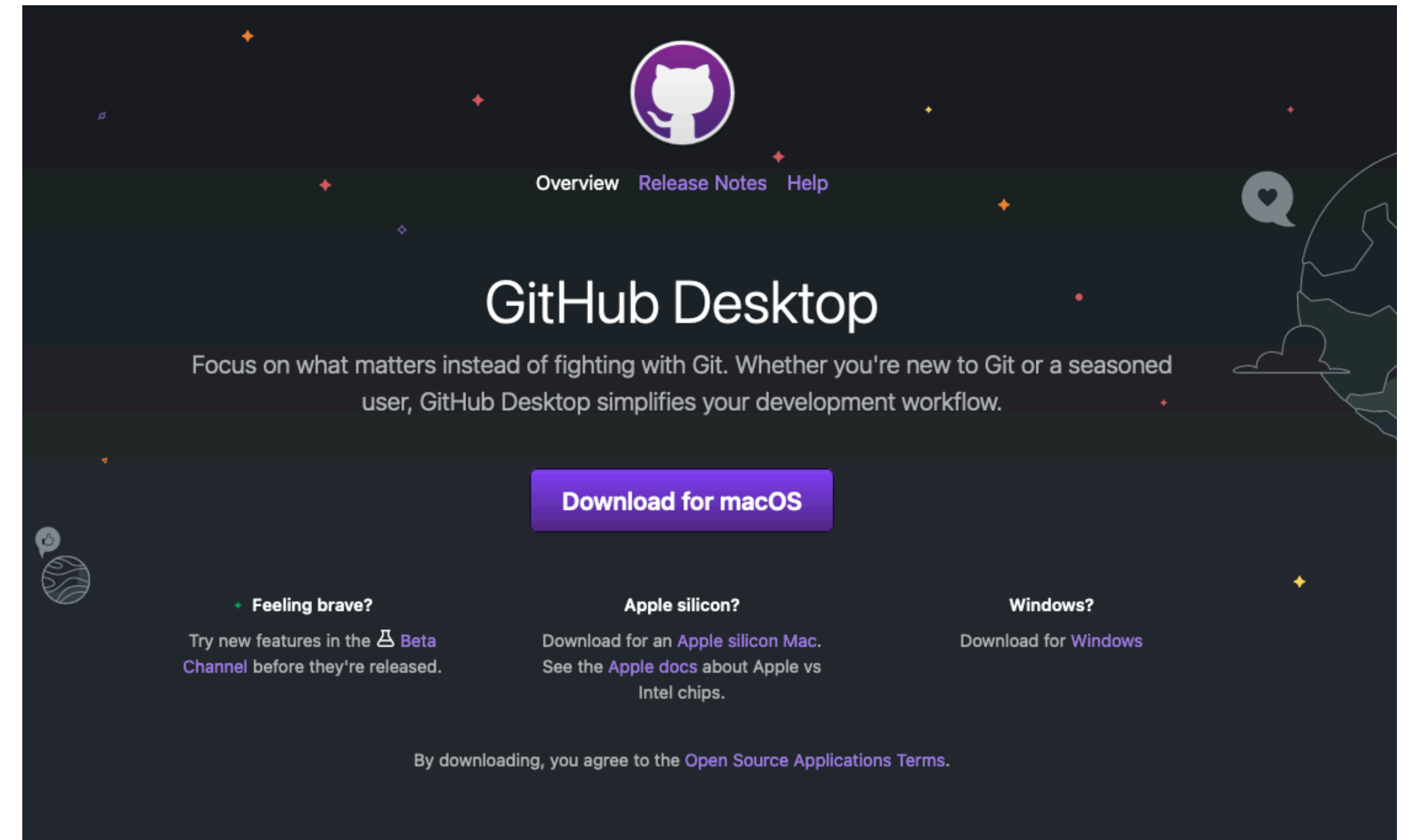
Any Questions?

Additional Resources

Don't want to use the Command Line?



GitHub Website: <https://github.com/>



GitHub Desktop: <https://desktop.github.com/>

Git, GitHub, and Markdown Guides

Git/GitHub

- “Mini-Videos” provided by Git: <https://git-scm.com/videos>
- “Git Cheat Sheet” by GitHub: <https://github.github.com/training-kit/downloads/github-git-cheat-sheet.pdf>
- “Learning Resources” by GitHub: <https://docs.github.com/en/get-started/quickstart/git-and-github-learning-resources>
- “GitHub Glossary” by GitHub: <https://docs.github.com/en/get-started/quickstart/github-glossary>
- “Set up Git” by GitHub: <https://docs.github.com/en/get-started/quickstart/set-up-git>

Markdown

- John Gruber, “Markdown” resource <https://daringfireball.net/projects/markdown/>
- Markdown Guide, “Cheat Sheet” <https://www.markdownguide.org/cheat-sheet/>

Where to Go Next?

With these tools:

- Download everything and try this exercise on your own if you haven't already!
- Check out the additional resources!
- Attend office hours with an MA/MS Program Advising Fellow (Nicole and Sam).
- Request an individual consultation with the GC Digital Fellows: <https://gcdi.commons.gc.cuny.edu/participate/>

At the Grad Center, generally:

- Check out events and workshops by the Digital Fellows (<https://gcdi.commons.gc.cuny.edu/participate/>), Mina Rees Library (<https://gc-cuny.libcal.com/calendar>), ITP (<https://itpcp.commons.gc.cuny.edu/>), and QRCC (<https://qrcc.commons.gc.cuny.edu/>). Keep an eye on your GC email/the Commons for these opportunities!
- Sign up for the GCDI's January Digital Research Institute/bootcamp: <https://gcdri.commons.gc.cuny.edu/>
- Utilize LinkedIn Learning training videos with your GC and NYPL accounts: <https://www.gc.cuny.edu/news/full-access-linkedin-learning-training-videos-free>
- Review the Library's Digital Tools and Techniques Guide: https://libguides.gc.cuny.edu/digital_tools_consult

Any Final Questions?