# Getting Started with Git and GitHub pt. 2

Coffee, Cookie and Coding (C3) Workshop supported by the Public Health Data Science and Data Equity team

Shelby Golden, M.S.

January 26th, 2026


xkcd Comic 1597 - Git Downloaded April 26th, 2025.

Yale SCHOOL OF PUBLIC HEALTH
*Data Science and Data Equity*

# Shelby Golden, M.S.

- Worked 7 years as a Molecular Biologist and Biochemist.
- Received a Masters in Applied Computational Mathematics from Johns Hopkins University in 2024.

# Today's Learning Objectives

01 Understand the purpose and value of Git and GitHub in managing coding projects. (~ 15 minutes)

02 Master essential Git and GitHub workflows for individual projects through hands-on practice. (~ 30 minutes)

03 Get introduced to collaborative workflows and teamwork strategies using guided GitHub examples. (~15 minutes)

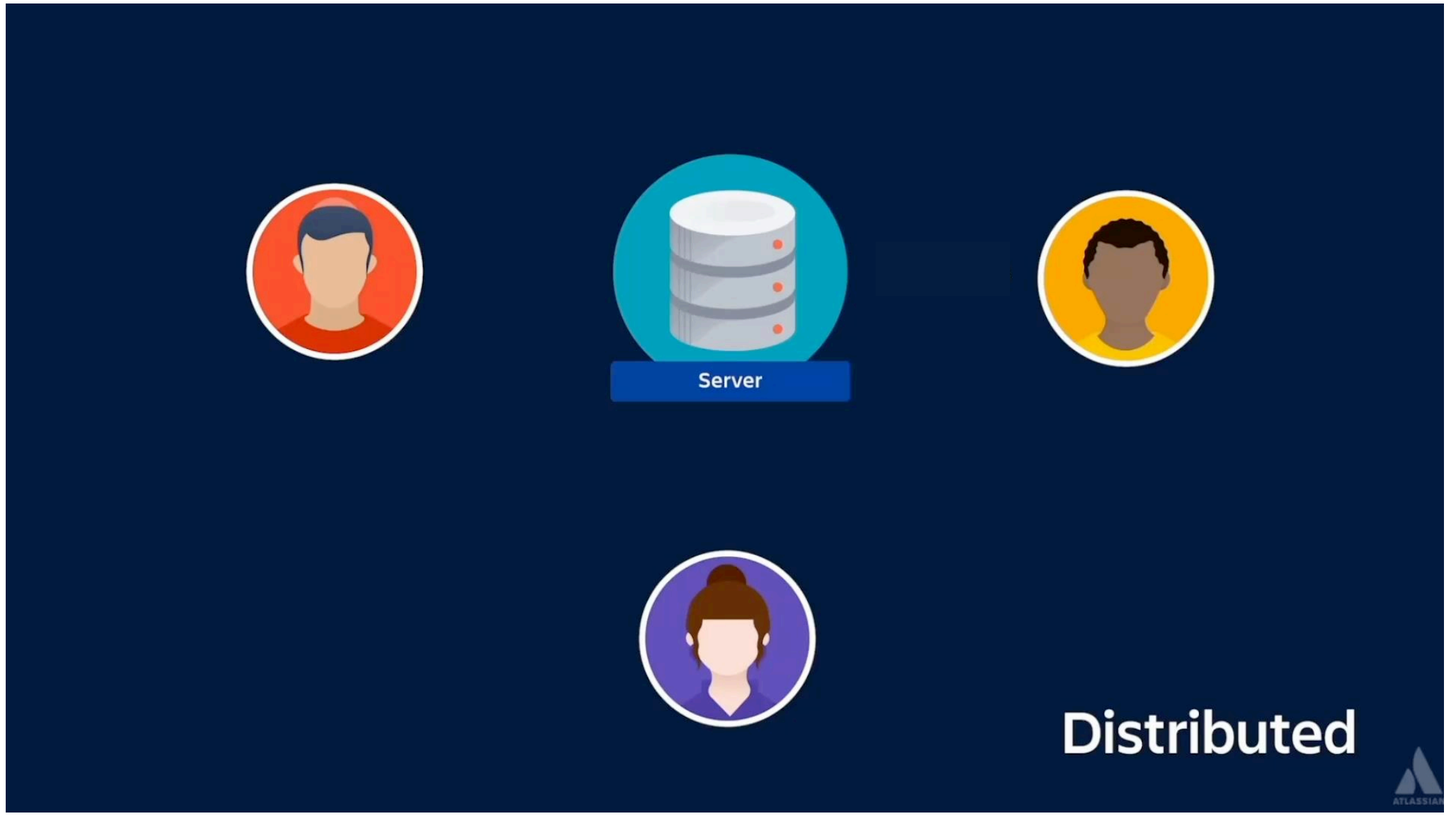
# Our Choice Resources

- Yale's Center for Research Computing workshop "Version Control by Git" by Kaylea Nelson, Ph.D.
- Yale's Harvey Cushing/John Hay Whitney Medical Library workshop "Git & GitHub: An Introduction To Version Control" by Justin DeMayo
- "Getting Git Right" by Atlassian
- Git and GitHub tutorials by W3Schools
- "Introduction to GitHub" by GitHub
- *Happy Git and GitHub for the useR* by Professor Jenny Bryan (and Yale alumn!) and Jim Hester

# Going Beyond Basic Git/GitHub

- Reviewing the developer documentation: git-scm.com/docs and docs.github.com
- "What is git commit, push, pull, log, aliases, fetch, config & clone" by Amit Prajapati
- "Git Guides" by various Graphite contributors
- "How to Write a Git Commit Message" by cbeams
- Git Graphical User Interface (GUI) clients by various contributors

# What is Git and GitHub?




"What is Version Control" by Atlassian. Updated February 23rd, 2020.
Git logo. Downloaded October 10th, 2024.
GitHub logo. Downloaded October 10th, 2024.

git

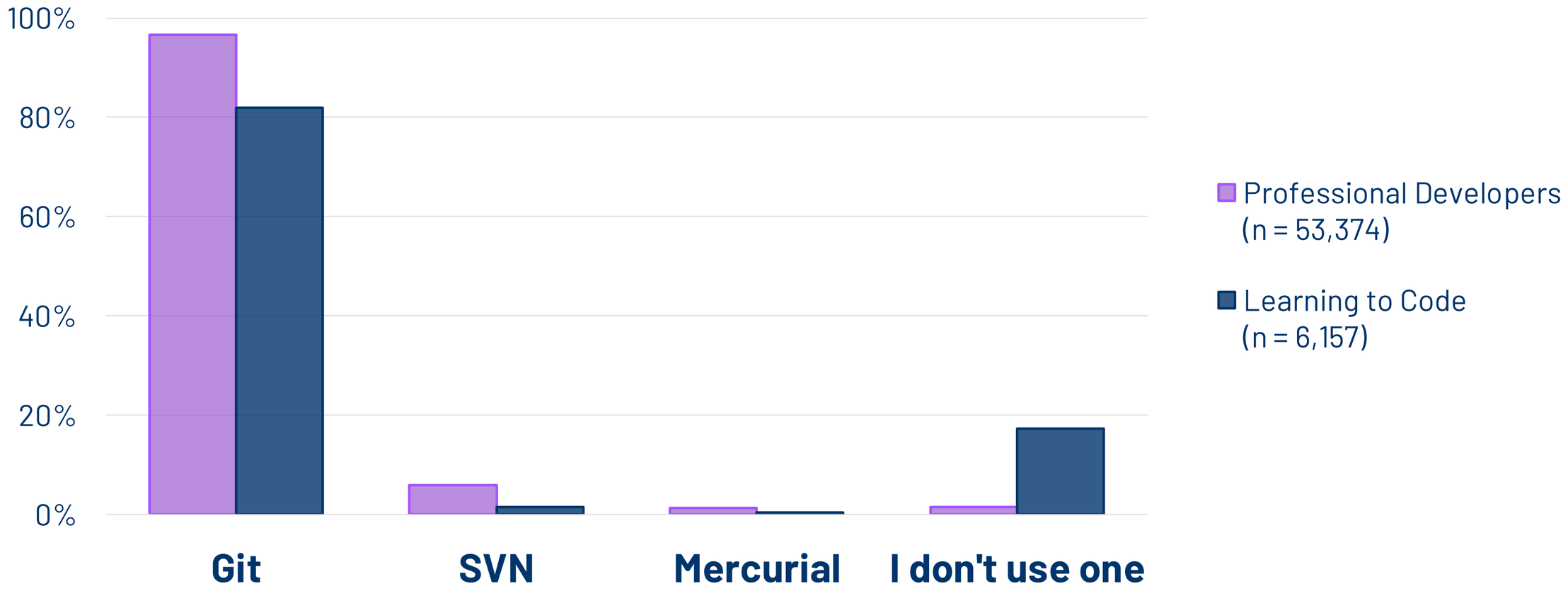
System for project management by distributive version control (DVCS).

GitHub

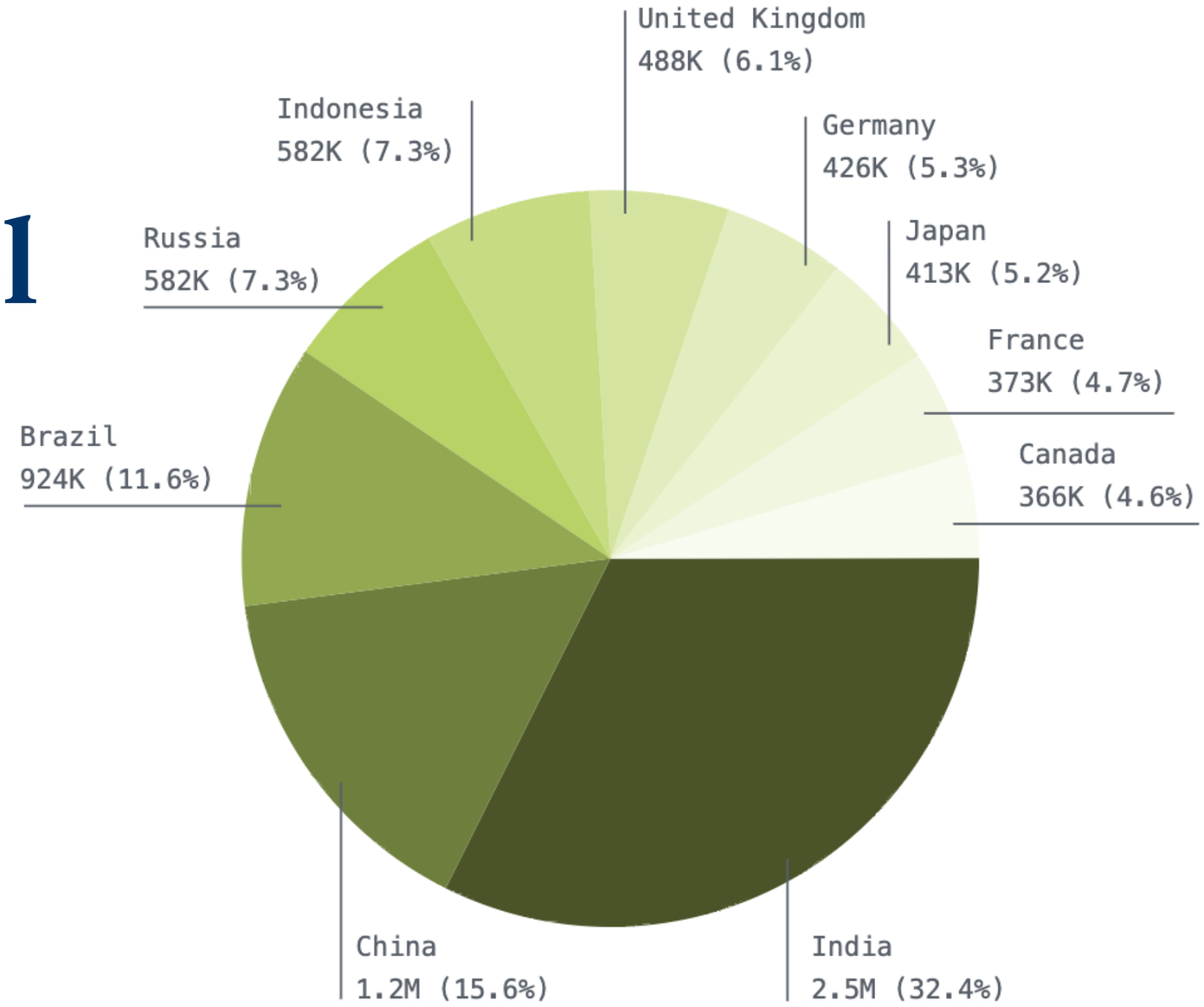
Developer platform for housing and managing projects and acts as the DVCS server.

# Git is the most widely used VCS but is underutilized by those learning to code.

2022 Developer Survey by StackOverflow. Published 2022.



# GitHub is an invaluable tool for coders worldwide.

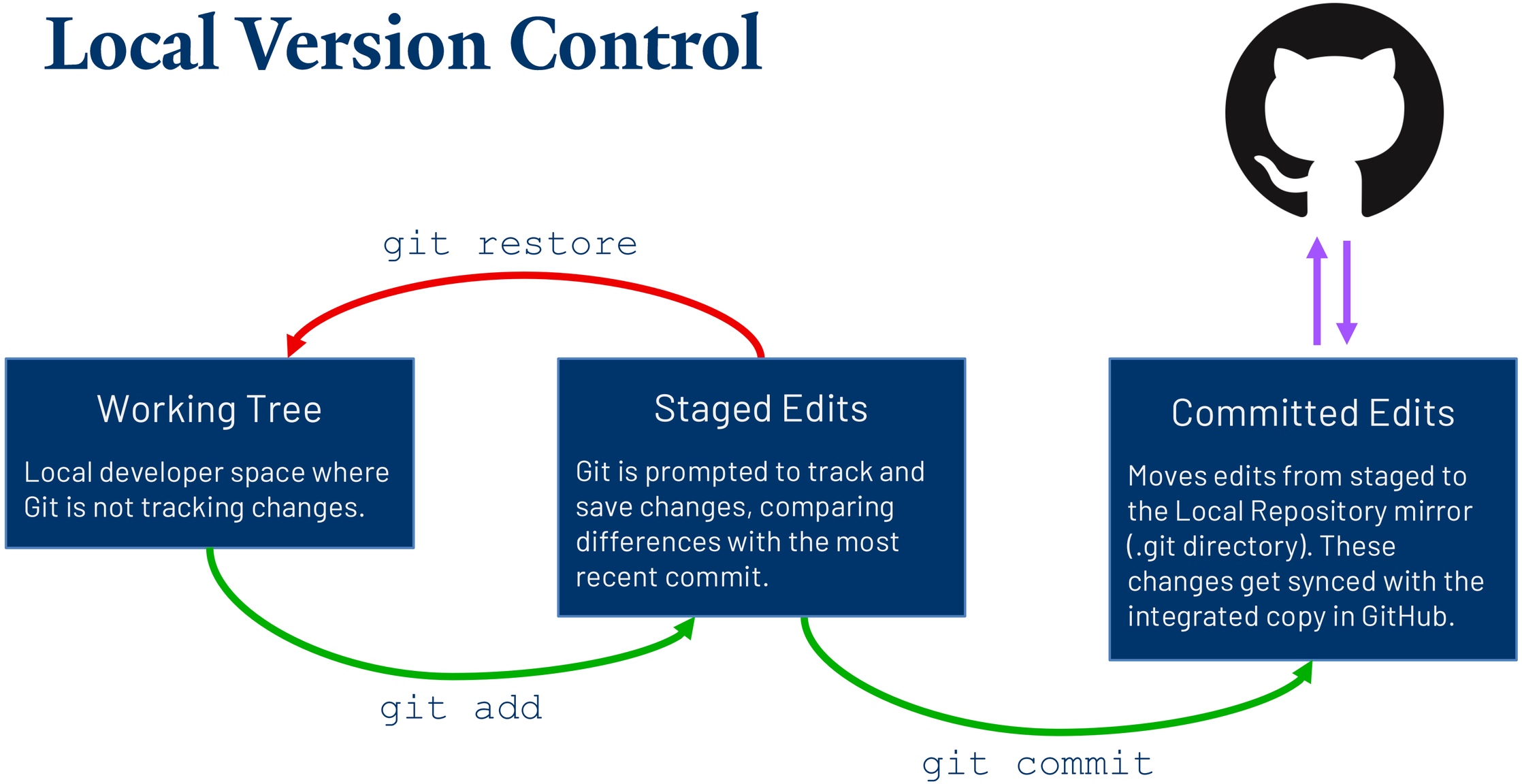




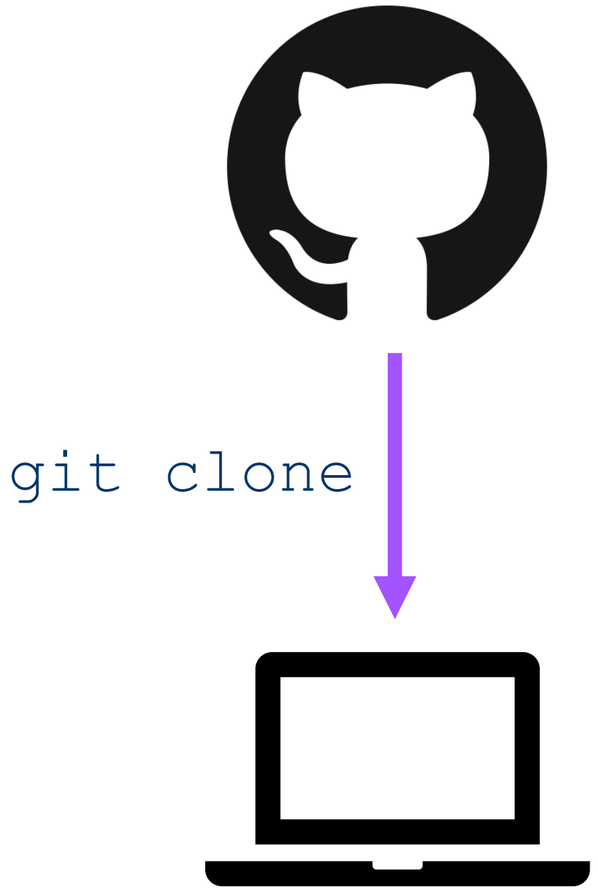

"A global community of developers." by GitHub, Octoverse 2022 report.

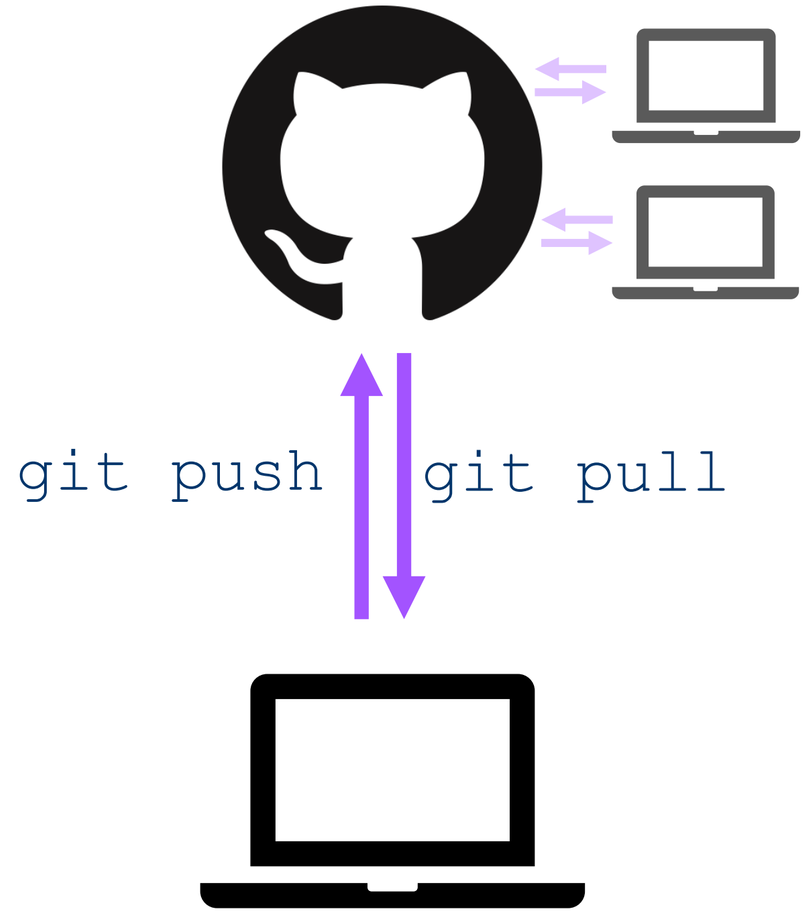
# How do Git and GitHub relate to one another?

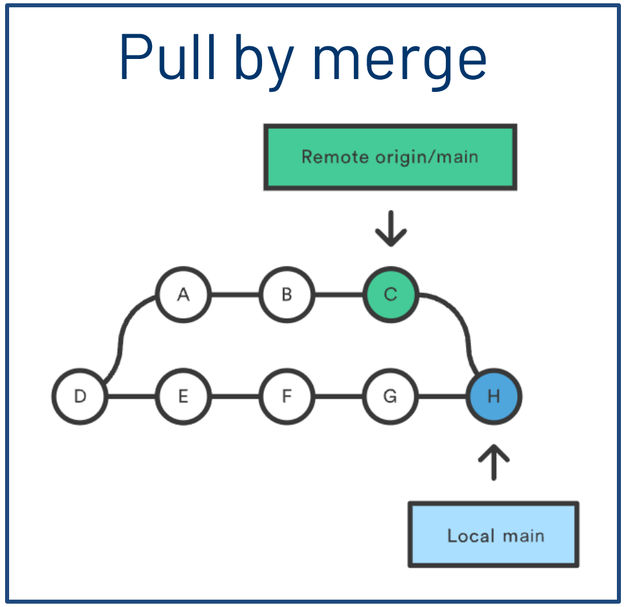
# Local Version Control

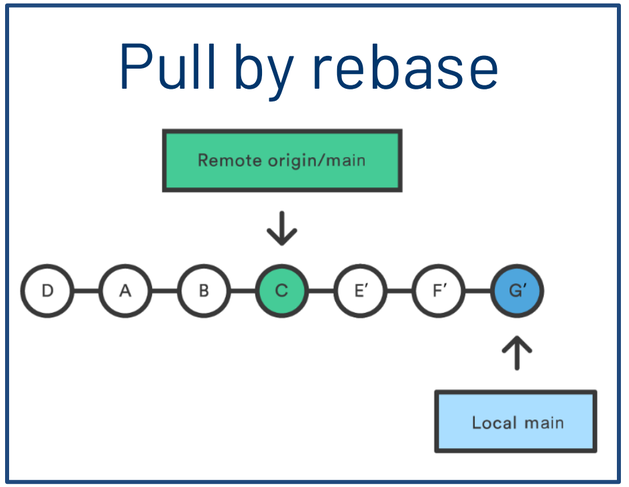


# Remote Storage and Distribution



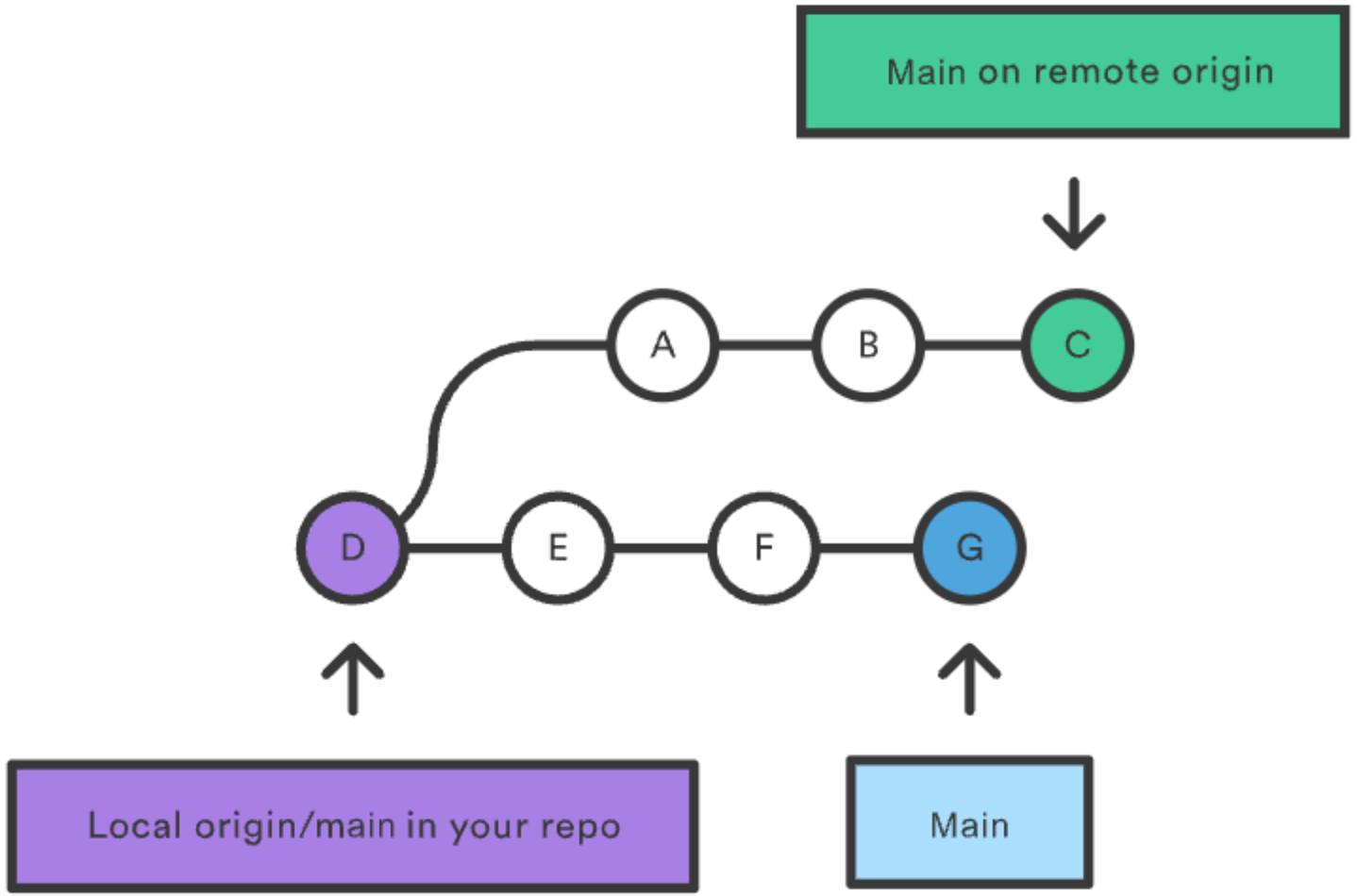





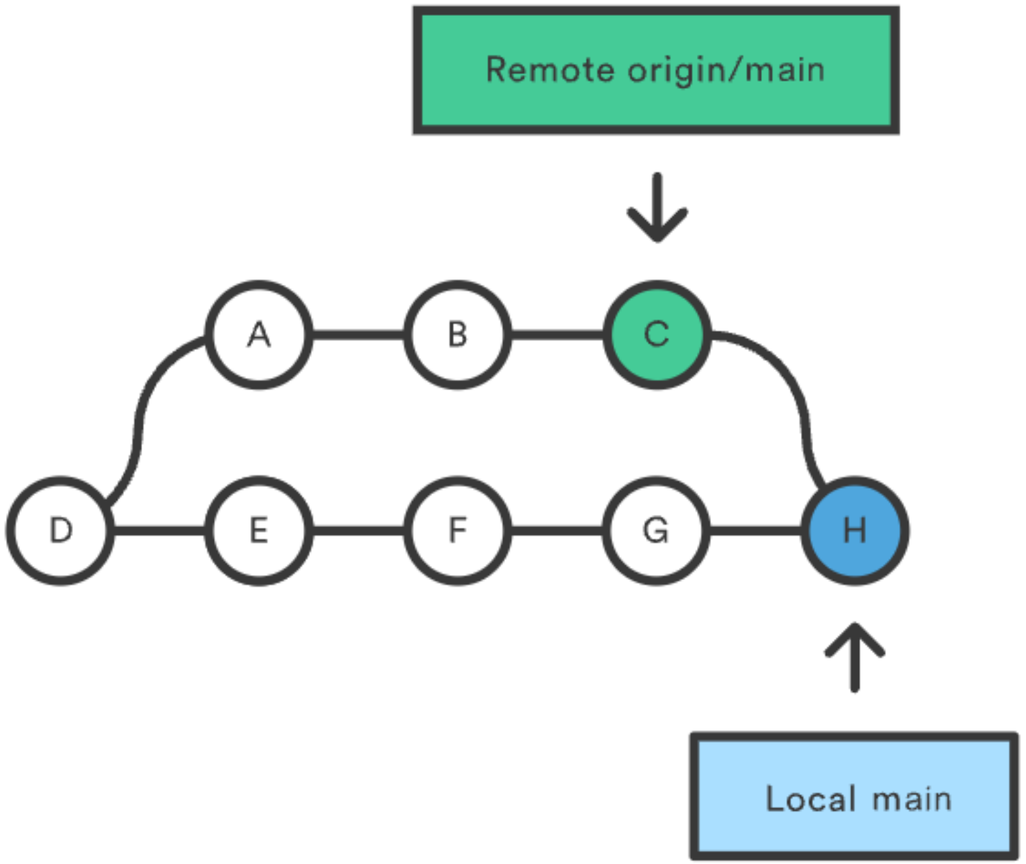

git pull from Atlassian's Git tutorials, Downloaded April 24th, 2025.

# Consider the example…

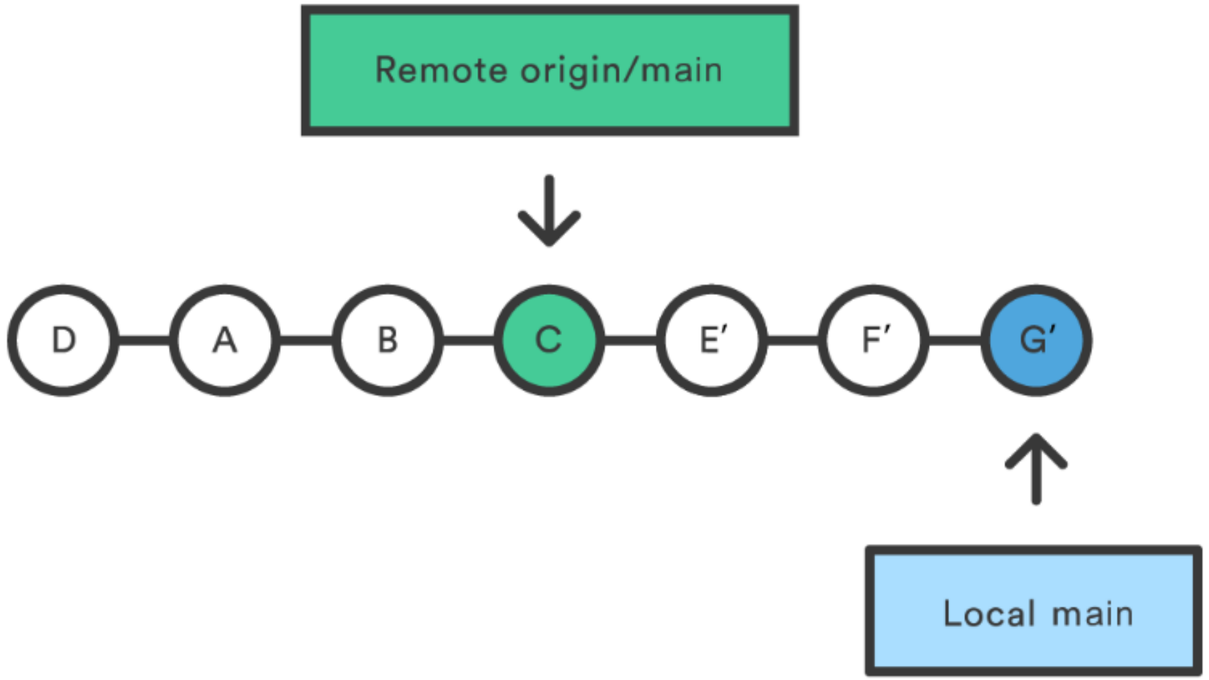



git pull from Atlassian's Git tutorials, Downloaded April 24th, 2025.

# git merge



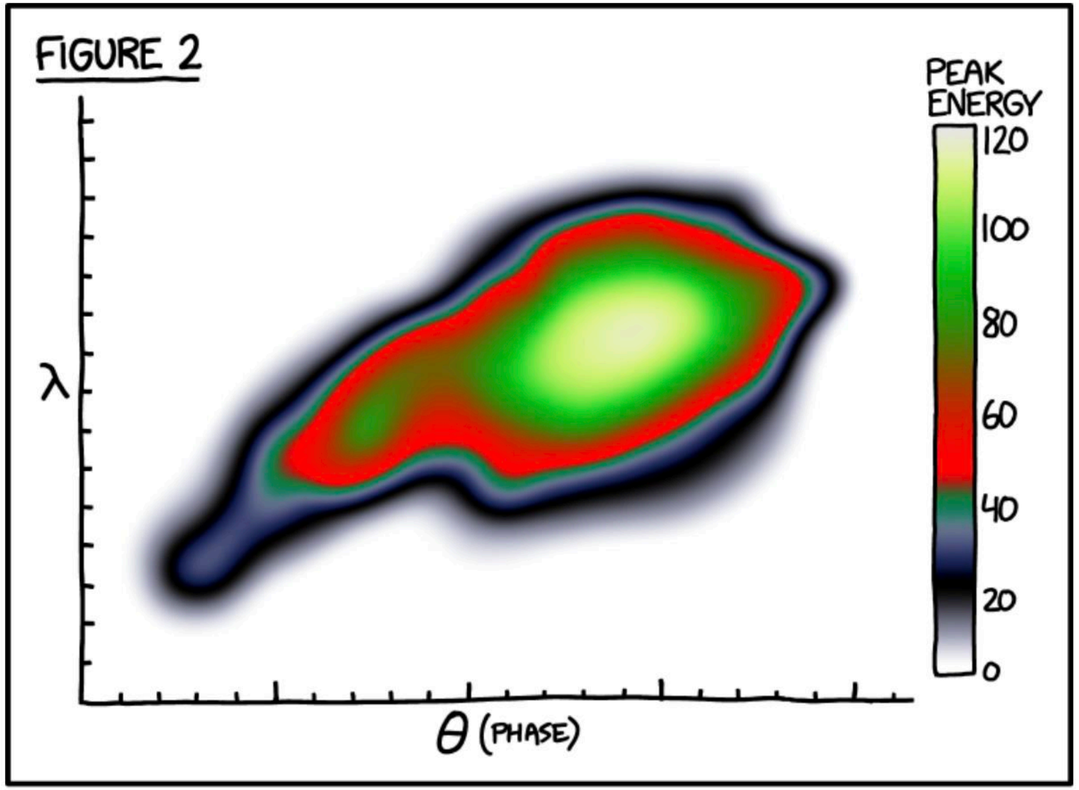

# git rebase




git pull from Atlassian's Git tutorials, Downloaded April 24th, 2025.

# Detailed Walk-Through

# Painbow color scale




xkcd Comic 2537 - Git Downloaded from, Flowing Data April 26th, 2025.

Command-Line Application

```
git status
```

Command-Line Output

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

Changes not staged for commit:
    (use "git add/rm ..." to update what will be committed)
    (use "git restore ..." to discard changes in working directory)

       modified:    making_cool_plots.R

Untracked files:
    (use "git add ..." to include in what will be committed)

       earth_shattering_color_scheme.png
```

Command-Line Application

```
# OPTION #1: List each file
git add "making_cool_plots.R" "earth_shattering_color_scheme.png"

# OPTION #2: Use the wild card "." to add all files
git add .

# View the results of git add.
git status
```

Command-Line Output

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

Changes to be committed:
    (use "git restore --staged ..." to unstage)

        modified:    making_cool_plots.R
        new file:    earth_shattering_color_scheme.png
```

```
git commit -m "Revelatory message elucidating the hidden secrets of git."
```

Command-Line Application

```
git commit
```

Command-Line Output

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit."
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#     (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:    making_cool_plots.R
#       new file:    earth_shattering_color_scheme.png
```

**Command-Line Output**

```
Outstanding progress on color schemes for density plot fill scaling.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit."
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#     (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:    making_cool_plots.R
#       new file:    earth_shattering_color_scheme.png
```

```
:wq
```

**Command-Line Output**

```
[main f9b4cf2] Outstanding progress on color schemes for density plot
fill scaling.
   2 files changed, 11 insertions (+), 2 deletions (-)
   create mode 100644 earth_shattering_color_scheme.png
```

Command-Line Application

```
# -----------
# Pull in One Command

# OPTION #1: Integrate the fetched copy of "origin/main" into "main" with merge.
git pull                          # Assuming the default protocol is a merge

# OPTION #2: Rebases "main" with the new parent history reflected in "origin/main".
git pull --rebase                 # Override the default merge to do a rebase
```

Command-Line Application

```
# -----------
# Two-Step Pull

# Download branch main from the remote repository, origin.
git fetch origin main

# Inspect changes.
git branch -a    # Lists all local branches, including remote copies
git status       # Summary of status differences between the two versions
git diff main    # Summary of code differences between the two versions

# OPTION #1: Integrate the fetched copy of "origin/main" into "main" with merge.
git merge FETCH_HEAD

# OPTION #2: Rebases "main" with the new parent history reflected in "origin/main".
git rebase FETCH_HEAD
```

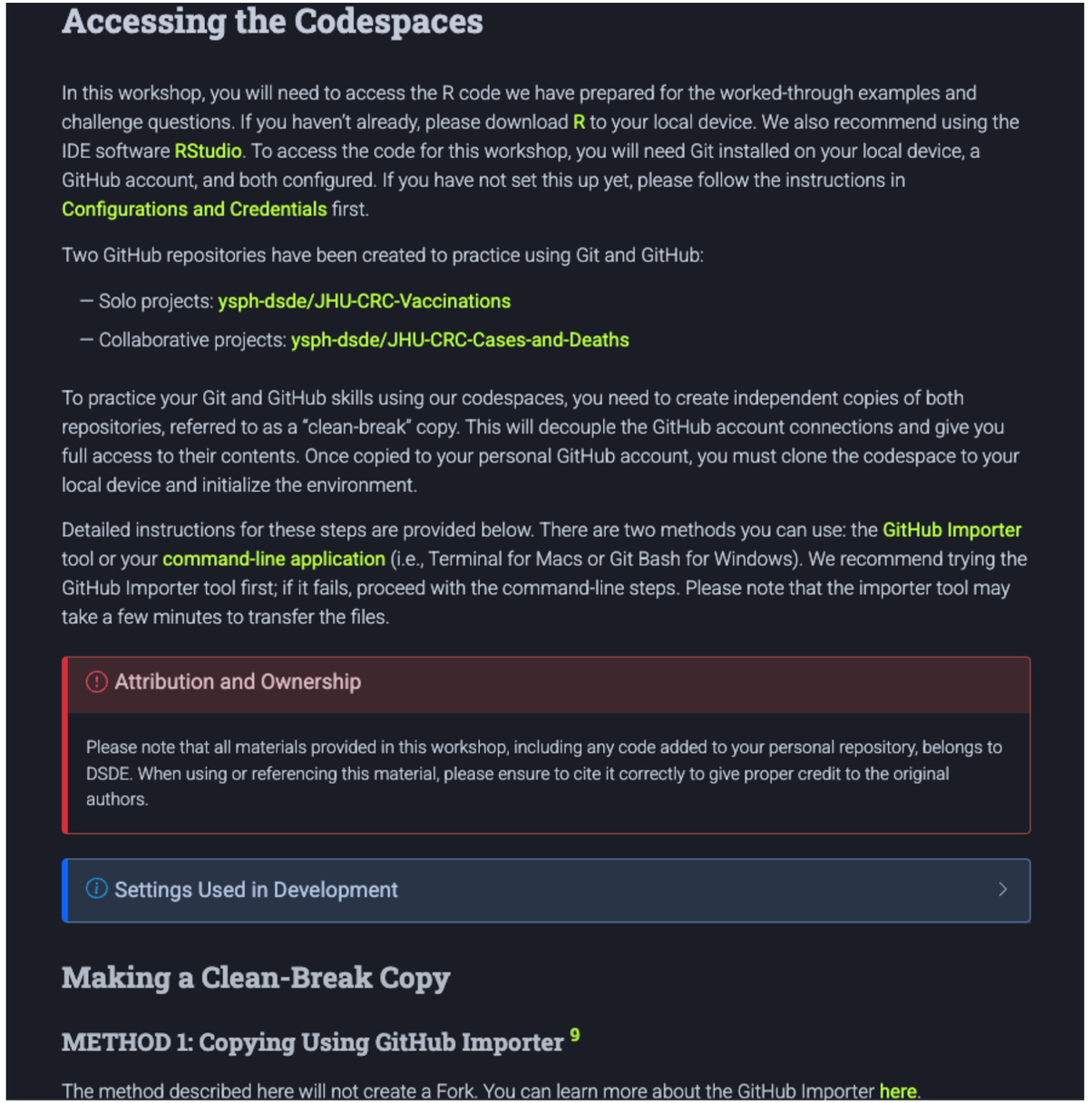
Command-Line Application

```
git push origin main
```

Command-Line Output

```
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 90.13 KiB | 22.53 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To github.com:ysph-dsde/PROJECT-REPOSITORY.git
   7d1b339..f9b4cf2 main -> main
```

# Starting Individual Projects



# Accessing the Codespaces

In this workshop, you will need to access the R code we have prepared for the worked-through examples and challenge questions. If you haven't already, please download **R** to your local device. We also recommend using the IDE software **RStudio**. To access the code for this workshop, you will need Git installed on your local device, a GitHub account, and both configured. If you have not set this up yet, please follow the instructions in **Configurations and Credentials** first.

Two GitHub repositories have been created to practice using Git and GitHub:

- Solo projects: **ysph-dsde/JHU-CRC-Vaccinations**
- Collaborative projects: **ysph-dsde/JHU-CRC-Cases-and-Deaths**

To practice your Git and GitHub skills using our codespaces, you need to create independent copies of both repositories, referred to as a "clean-break" copy. This will decouple the GitHub account connections and give you full access to their contents. Once copied to your personal GitHub account, you must clone the codespace to your local device and initialize the environment.

Detailed instructions for these steps are provided below. There are two methods you can use: the **GitHub Importer** tool or your **command-line application** (i.e., Terminal for Macs or Git Bash for Windows). We recommend trying the GitHub Importer tool first; if it fails, proceed with the command-line steps. Please note that the importer tool may take a few minutes to transfer the files.

**Attribution and Ownership**

Please note that all materials provided in this workshop, including any code added to your personal repository, belongs to DSDE. When using or referencing this material, please ensure to cite it correctly to give proper credit to the original authors.

**Settings Used in Development**

## Making a Clean-Break Copy

### METHOD 1: Copying Using GitHub Importer [9]

The method described here will not create a Fork. You can learn more about the GitHub Importer **here**.

# Handout: Solo Projects

Yale SCHOOL OF PUBLIC HEALTH
*Data Science and Data Equity*

## START LOCALLY

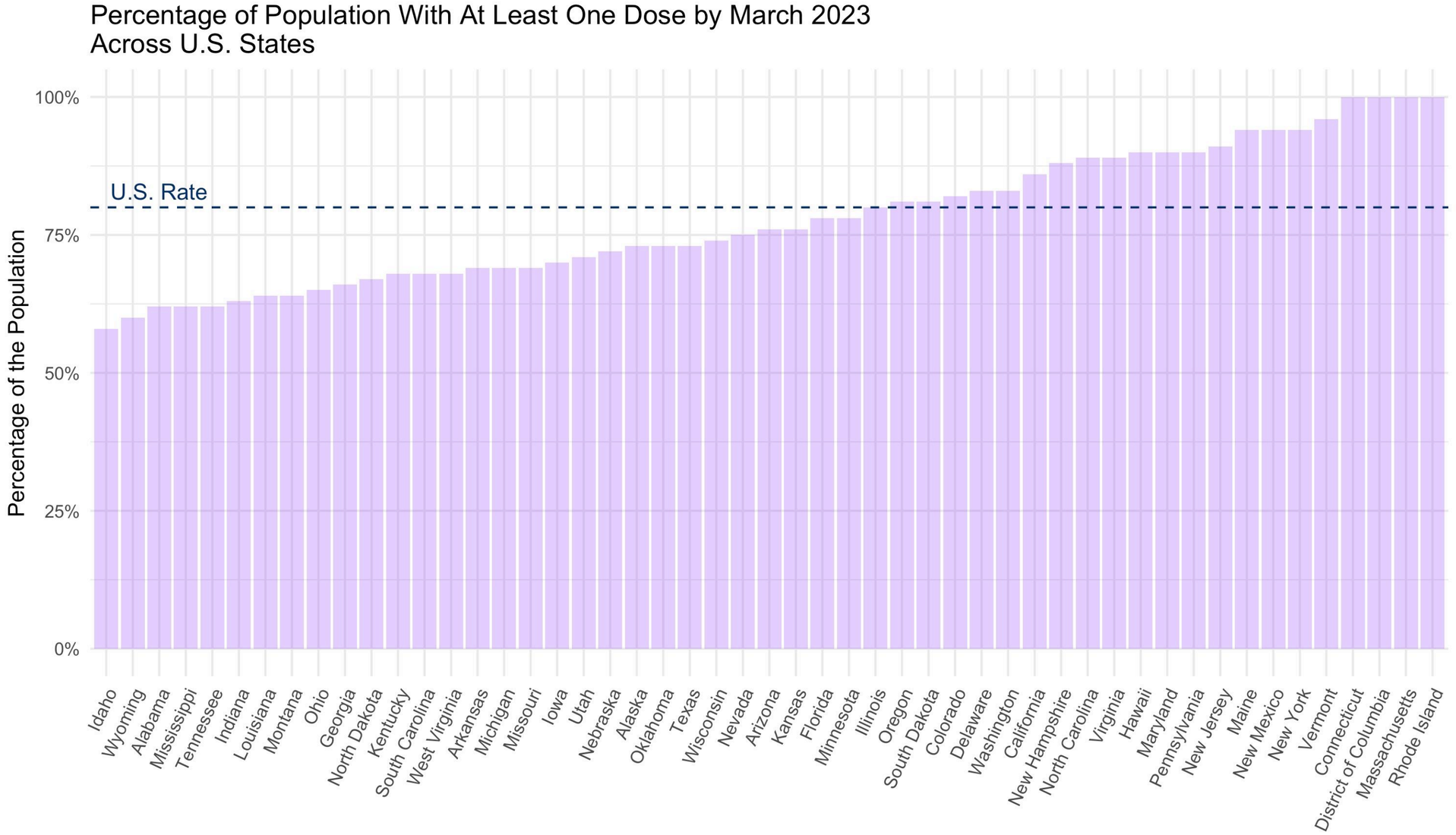
1. Initialize Git locally, create an empty README.md file, and push to an empty GitHub repository.
2. Add "Hello World" to README.md on GitHub and pull the changes locally.

## START REMOTELY

1. Clone the clean-break copy of the JHU-CRC-Vaccinations repository (Method #1 or Method #2).
2. Create and export a bar plot (customize as desired).
3. Commit and push the changes to GitHub.

# Discussion:

How would this process change if I had modified the local copy (e.g., adding a new R script) before pulling the `README.md` changes?


Percentage of Population With At Least One Dose by March 2023
Across U.S. States
Percentage of the Population
100%
75%
50%
25%
0%
U.S. Rate
Idaho
Wyoming
Alabama
Mississippi
Tennessee
Indiana
Louisiana
Montana
Ohio
Georgia
North Dakota
Kentucky
South Carolina
West Virginia
Arkansas
Michigan
Missouri
Iowa
Utah
Nebraska
Alaska
Oklahoma
Texas
Wisconsin
Nevada
Arizona
Kansas
Florida
Minnesota
Illinois
Oregon
South Dakota
Colorado
Delaware
Washington
California
New Hampshire
North Carolina
Virginia
Hawaii
Maryland
Pennsylvania
New Jersey
Maine
New Mexico
New York
Vermont
Connecticut
District of Columbia
Massachusetts
Rhode Island

# Handout: Solo Projects

Yale SCHOOL OF PUBLIC HEALTH
*Data Science and Data Equity*

## START LOCALLY

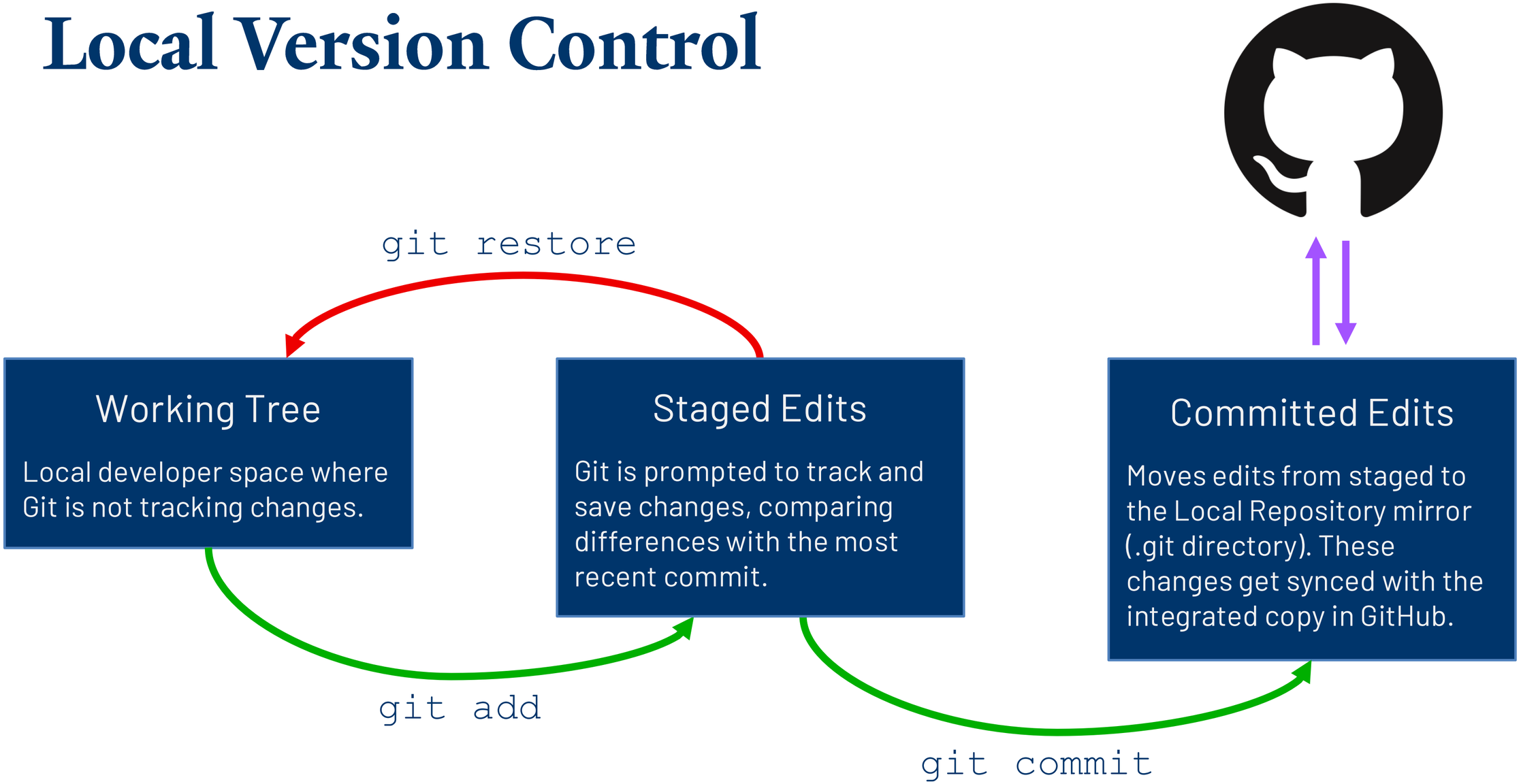
1. Initialize Git locally, create an empty README.md file, and push to an empty GitHub repository.
2. Add "Hello World" to README.md on GitHub and pull the changes locally.

## START REMOTELY

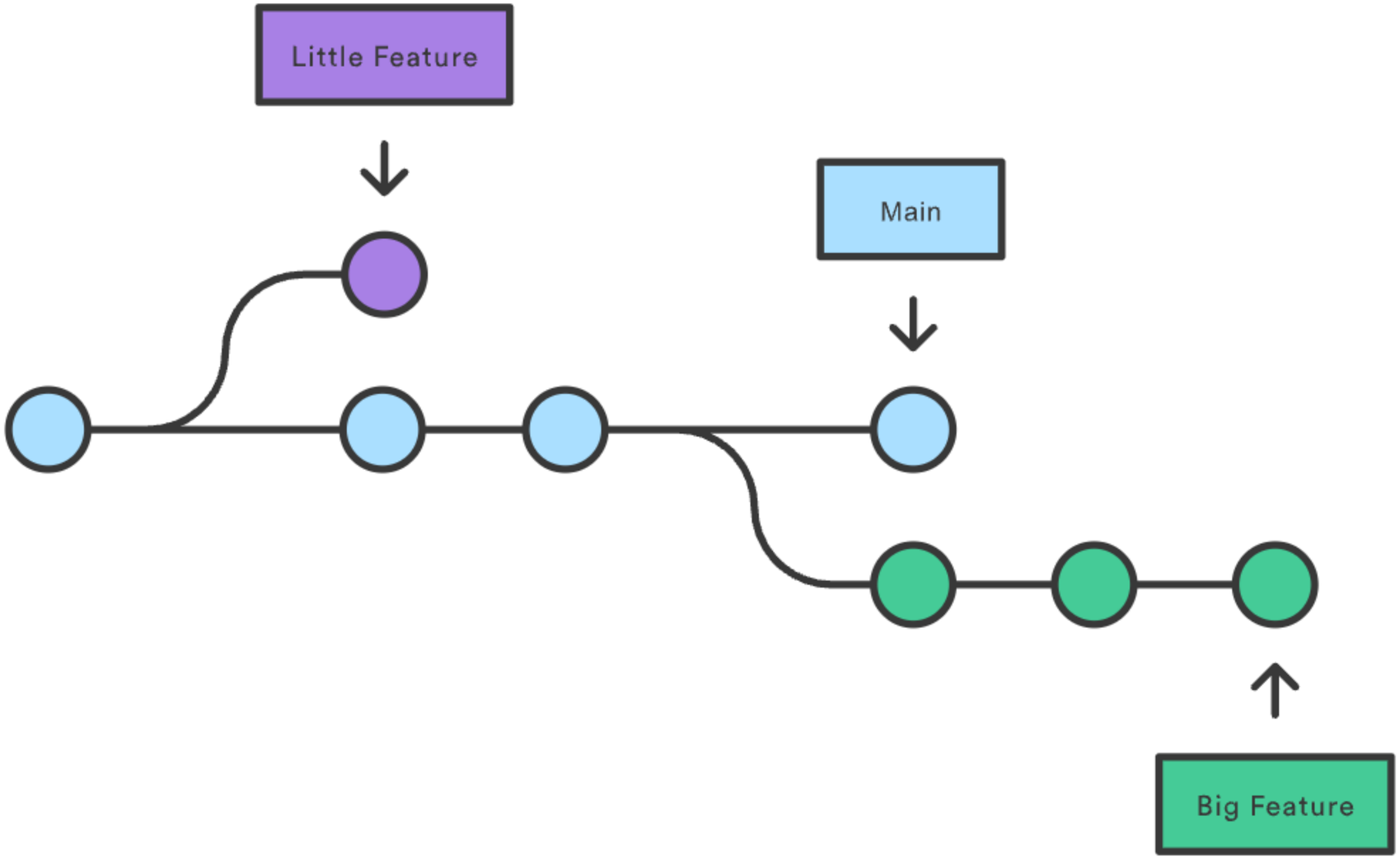
1. Clone the clean-break copy of the JHU-CRC-Vaccinations repository (Method #1 or Method #2).
2. Create and export a bar plot (customize as desired).
3. Commit and push the changes to GitHub.

# Leveraging Branches for Collaboration

# Local Version Control



# Consider the example…



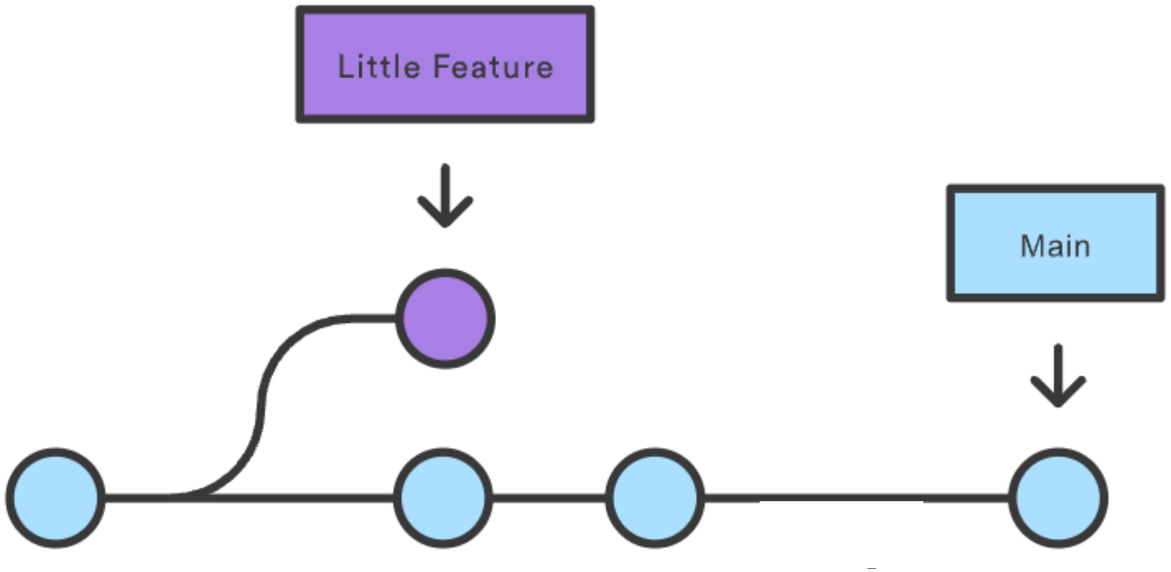


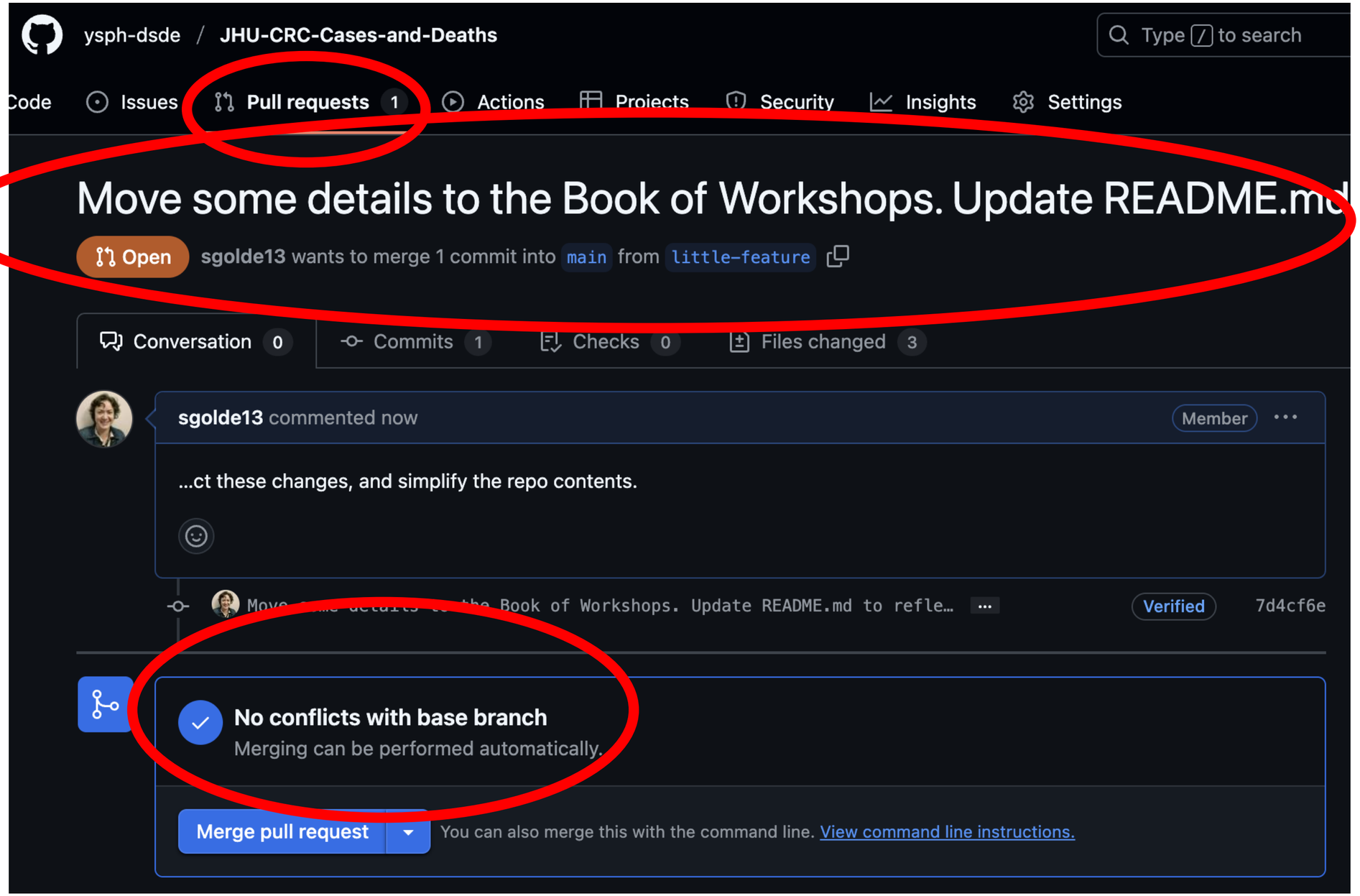

git branch from Atlassian's Git tutorials, Downloaded April 27th, 2025.

Command-Line Application

```
# Create a new branch called "little-feature".
git branch little-feature

# Move from the main branch into the new one.
git checkout little-feature
```




ysph-dsde / JHU-CRC-Cases-and-Deaths
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Security
Insights
Settings
Move some details to the Book of Workshops. Update README.md
Open
sgolde13 wants to merge 1 commit into main from little-feature
Conversation 0
Commits 1
Checks 0
Files changed 3
sgolde13 commented now
Member
...ct these changes, and simplify the repo contents.
Book of Workshops. Update README.md to refle…
Verified
7d4cf6e
No conflicts with base branch
Merging can be performed automatically.
Merge pull request
You can also merge this with the command line. View command line instructions.

# Handout: Collaborations

Yale SCHOOL OF PUBLIC HEALTH
*Data Science and Data Equity*

### DevOps

Introduction to this widely-used collaborative philosophy and structure for technical projects.

### PULL REQUESTS

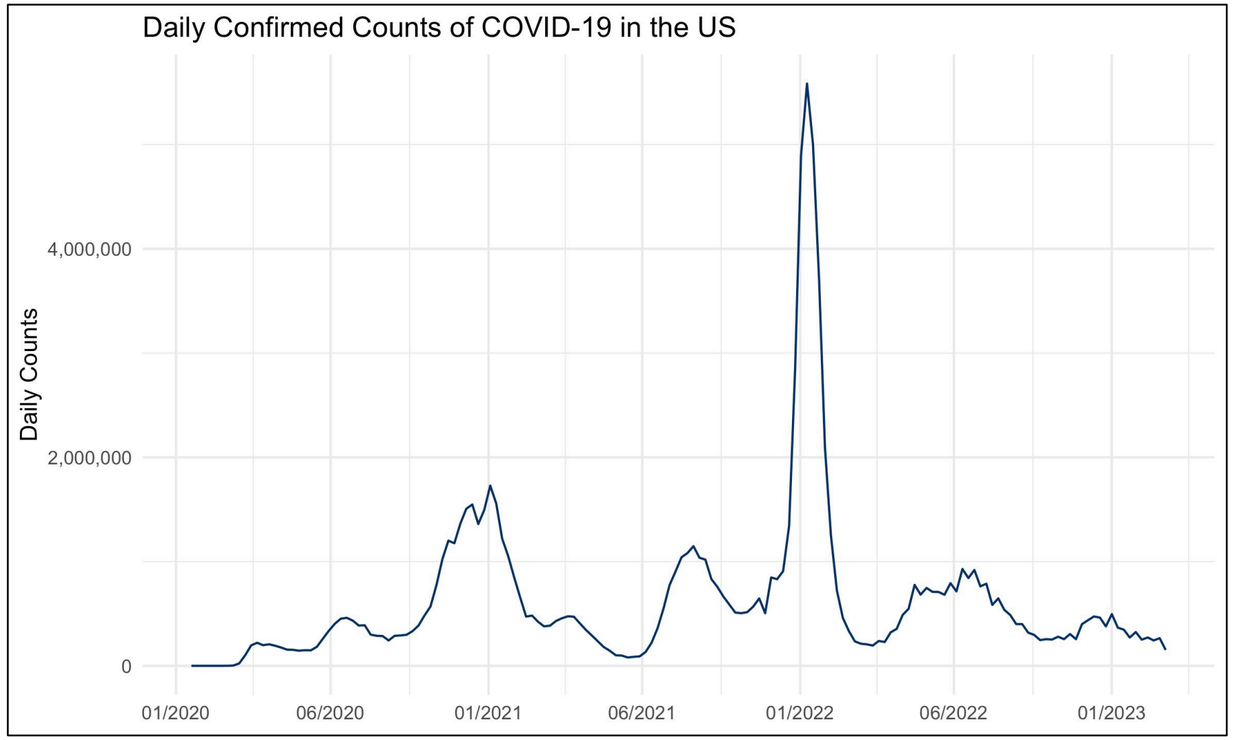
Mechanics of creating a Pull Request for team review before merging into `origin/main`.

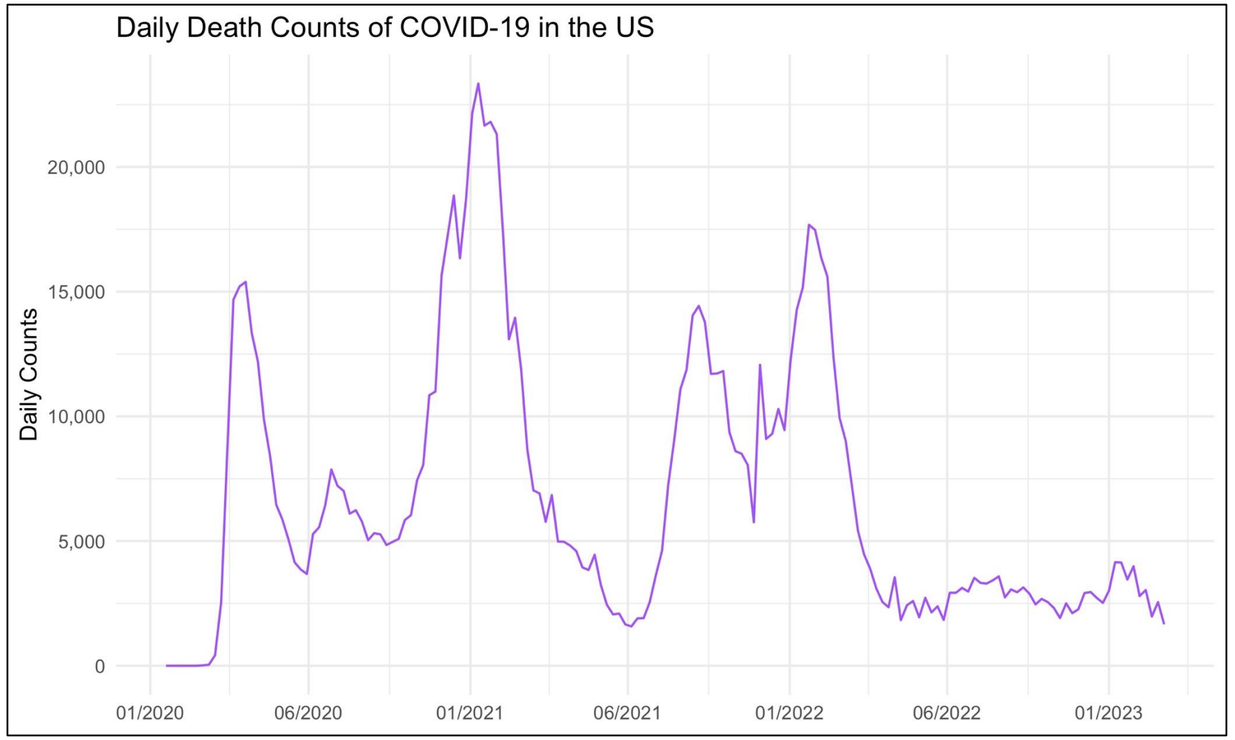
### MERGE CONFLICTS

Identifying merge conflicts and resolving them in GitHub and locally through the CLI.

### RULESETS

Applying branch protections and observing their impact on workflow.


Daily Confirmed Counts of COVID-19 in the US
Daily Counts
4,000,000
2,000,000
0
01/2020
06/2020
01/2021
06/2021
01/2022
06/2022
01/2023


Daily Death Counts of COVID-19 in the US
Daily Counts
20,000
15,000
10,000
5,000
0
01/2020
06/2020
01/2021
06/2021
01/2022
06/2022
01/2023

# Thank you!

**ysph.yale.edu**
**sph.yale.edu/dsde**

**@YaleSPH**

---

**Public Health Data Science and Data Equity**
**Yale School of Public Health**
**60 College Street, New Haven, CT 06510**

Yale SCHOOL OF PUBLIC HEALTH

# Appendix

# Glossary

**Command-Line Interface (CLI)** A texted-base application that directly interacts with the computer's operating system, manages files, and can run programs. It typically lacks a graphical user interface (GUI).

**Distributive Version Control System** The project codebase is copied as a mirror to each contributor's local computer. Local changes get synched via patches sent "peer-to-peer" through the server.

**`git add`** Prompt Git to track changes made to specified files and transition them from the **Working Tree** to the **Staged Edits** domain. Git compares them to the `.git` directory.

# Glossary

**`git branch`** Create or list different “versions” or “paths” of a project that you can work on separately.

**`git checkout`** Command to switch between the different branches.

**`git clone`** Copies an existing repository stored in a remote server to your own device, including all files, history, and branches.

**`git commit`** Promote the staged version of specified files to become the latest copy reflected in the `.git` directory. The **Committed Edits** version is synced to the remote server as a “peer-to-peer” patch.

# Glossary

**`git fetch`** Downloads the remote branch contents into a separate local branch for review before integration. Part of `pull`.

**`git merge`** One method to reconcile different committing histories in divergent branches. Creates a new version integrating the head of the two branches in a three-way commit.

**`git mv`** Moves or renames a specified file, directory, or symlink that is already tracked by Git.

**`git pull`** Downloads a copy of the specified remote repository's branch and integrates it with the local copy. Combines the actions of `fetch` followed by `merge` or `rebase`.

# Glossary

**`git push`** Upload your recent **Committed Edits** to a remote repository, synchronizing changes for others to see.

**`git rebase`** An alternative to `merge`. The branch commit histories are realigned so that the leading one defines the commit parent history of the following branch, thus rebasing its commits.

**`git reset`** Reverts content in the **Working Tree** or **Staged Edits** to their last committed state. Adding the `--staged` option will move files from **Staged Edits** back to the **Working Tree**, without overwriting any new changes there.

# Glossary

| | |
|---|---|
| `git status` | Summarizes the state of files in the **Working Tree** and **Staged Edits**, comparing changes to the latest, committed version in the `.git` directory. |
| **Graphical User Interface (GUI)** | An interface that allows users to interact with computers through visual elements like buttons and menus. |
| **Hypertext Transfer Protocol Secure (HTTPS)** | Facilitates secure transmission of information over an encrypted internet connection. Decryption occurs when the correct username and Personal Access Token (PAT) are provided. |

# Glossary

**Integrated Development Environment (IDE)** A software application that combines tools for editing, building, testing, and debugging code into a single, user-friendly interface.

**Mirror** An exact copy of a project from a server, including al files, the version history, and branches.

**Patch** Snippets of code or data used to update existing software.

**Peer-to-peer** Participants in a network act as both client and server by trading resources and services with one another.

# Glossary

**Root directory** The top-most directory in a branched hierarchy, containing all other files and directories. Your project's root directory holds all the relevant files and code used your project.

**Server and Client** Servers are computers or systems that provides resources (i.e. data or programs) to other computers, known as clients, over a network.

**Shell** A program used by the CLI to mediate communication between the user and computer by interpreting commands and outputs. Examples: Bash, PowerShell, etc.

# Glossary

**Secure Shell (SSH)** Uses a pair of keys—a private key and a public key. Information is encrypted when sent between servers, and only the matching key pair can decrypt it upon receipt.

**Version Control** Manage, organize, and track different versions of files. Identify differences between versions and allows reverting to older versions.

# Glossary

## Helpful Cheat Sheets:

1. Git Cheat Sheet by Atlassian
2. Vim Cheat Sheet by Richard Torruellas
3. Bash Shortcuts by Mohankumar Balasundaram
4. Command Line Cheat Sheet by Tobias Günther

# References


**Slide 1**

1. xkcd, “Git.” Accessed: Apr. 27, 2025. [Online]. Available: https://xkcd.com/1597/

**Slide 4**

1. K. DuMez, “How to rename a file in Git,” Graphite. Accessed: Apr. 21, 2025. [Online]. Available:
   Atlassian, “Learn Git,” Atlassian Tutorials. Accessed: Oct. 09, 2024. [Online]. Available: https://www.atlassian.com/git
2. J. Bryan Ph.D. and J. Hester, “Let’s Git started,” Happy Git and GitHub for the useR. Accessed: Oct. 09, 2024. [Online]. Available: https://happygitwithr.com/
3. J. DeMayo, “gitdemo,” Harvey Cushing/John Hay Whitney Medical Library. Accessed: Nov. 01, 2024. [Online]. Available: https://github.com/CWML/gitdemo/tree/main
4. GitHub, “Introduction to GitHub,” GitHub Skills. Accessed: Nov. 01, 2024. [Online]. Available: https://github.com/skills/introduction-to-github

# References


**Slide 4 continued**

5. K. Nelson Ph.D., “Version Control with Git,” Yale Center for Research Computing (YCRC). Accessed: Oct. 09, 2024. [Online]. Available: https://research.computing.yale.edu/training/version-control-git

6. W3Schools, “Git Tutorial,” W3Schools. Accessed: Oct. 09, 2024. [Online]. Available: https://www.w3schools.com/git/default.asp?remote=github

**Slide 5**

1. A. Prajapati, “What is git commit, push, pull, log, aliases, fetch, config & clone,” Medium. Accessed: Oct. 28, 2024. [Online]. Available: https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c

2. Various contributors, “Git guides,” Graphite. Accessed: Oct. 28, 2024. [Online]. Available: https://graphite.dev/guides/topic/git

# References


**Slide 5 continued**

3. cbeams, “How to Write a Git Commit Message,” cbeams. Accessed: Oct. 28, 2024. [Online]. Available: https://cbea.ms/git-commit/
4. Git-SCM, “Git - GUI Clients,” Git-SCM. Accessed: Oct. 28, 2024. [Online]. Available: https://git-scm.com/downloads/guis
5. GitHub Developers, “GitHub Docs,” GitHub. Accessed: Apr. 24, 2025. [Online]. Available: https://docs.github.com/en
6. Git Developers, “Git - Reference,” Git. Accessed: Apr. 24, 2025. [Online]. Available: https://git-scm.com/docs

**Slide 7**

1. Atlassian, “What is version control,” Atlassian Tutorials. Accessed: Oct. 14, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-version-control

# References

**Slide 7 continued**


2. S. Chaudhary, “How GitHub became the largest source code host?,” Medium. Accessed: Oct. 14, 2024. [Online]. Available: https://srishtichy.medium.com/how-github-became-the-largest-source-code-host-a1c37ea5e5f

3. K. Finley, “The Problem With Putting All the World’s Code in GitHub,” WIRED Magazine. Accessed: Oct. 14, 2024. [Online]. Available: https://www.wired.com/2015/06/problem-putting-worlds-code-github/

4. “Distributed version control,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Distributed_version_control

5. “GitHub,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/GitHub

6. “Git,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Git

# References


**Slide 8**

1. StackOverflow, “Version control systems results,” StackOverflow Developer Survey. Accessed: Oct. 14, 2024. [Online]. Available: https://survey.stackoverflow.co/2022/#version-control-version-control-system-prof

**Slide 9**

1. GitHub, “Global distribution of developers,” GitHub Octoverse. Accessed: Oct. 14, 2024. [Online]. Available: https://octoverse.github.com/2022/global-tech-talent

**Slide 11**

1. K. DuMez, “How to rename a file in Git,” Graphite. Accessed: Apr. 21, 2025. [Online]. Available: https://graphite.dev/guides/rename-file-in-git
2. J. Houghton, “Boost your Git confidence with the new file rename notification.” Accessed: Apr. 21, 2025. [Online]. Available: https://devblogs.microsoft.com/visualstudio/boost-your-git-confidence-with-the-new-file-rename-notification/

# References


**Slide 11 continued**

3. A. Prajapati, “What is git commit, push, pull, log, aliases, fetch, config & clone,” Medium. Accessed: Oct. 28, 2024. [Online]. Available: https://medium.com/mindorks/what-is-git-commit-push-pull-log-aliases-fetch-config-clone-56bc52a3601c

4. M. Lorenz, “Git Restore and Git Reset,” Academind GmbH. Accessed: Oct. 05, 2025. [Online]. Available: https://academind.com/tutorials/git-restore-and-git-reset

5. G. Foster, “Understanding and using the `git restore` command,” Graphite. Accessed: Jan. 24, 2026. [Online]. Available: https://graphite.com/guides/git-restore

**Slide 12**

1. Atlasian, “git clone,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone

2. Atlassian, “git rebase,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase

# References


**Slide 12 continued**

3. Atlassian, “git merge,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-merge

4. Atlassian, “git pull,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-pull

5. Atlassian, “git fetch,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-fetch

**Slides 13-14**

1. Atlassian, “git fetch,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-fetch

2. Atlassian, “git pull,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-pull

# References


**Slides 13-14 continued**

3. Atlassian, “git merge,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-merge

**Slide 16**

1. xkcd, “Painbow color scale.” Accessed: Apr. 27, 2025. [Online]. Available: https://flowingdata.com/2021/11/08/painbow-color-scale/

**Slides 17-20**

1. G. Foster, “Git add, commit, and push,” Graphite. Accessed: Nov. 02, 2024. [Online]. Available: https://graphite.dev/guides/git-add-commit-push

# References


**Slides 19-20**

1. Answered by Go Dan, “File Permissions - How to read the mode field of git-ls-tree’s output,” StackExchange. Accessed: Apr. 24, 2025. [Online]. Available: https://stackoverflow.com/questions/737673/how-to-read-the-mode-field-of-git-ls-trees-output/8347325#8347325
2. K. Hess, “An introduction to the vi editor.” Accessed: Apr. 22, 2025. [Online]. Available: https://www.redhat.com/en/blog/introduction-vi-editor
3. R. Torruellas, “Vim Cheat Sheet,” Vim. Accessed: Nov. 02, 2024. [Online]. Available: https://vim.rtorr.com/

**Slide 21**

1. Atlassian, “git pull,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-pull

# References


**Slide 22**

1. GitKraken, “Git Tutorial #32: What Is Git Fetch and How to Use It | Learn Git,” YouTube. Accessed: Jan. 24, 2026. [Online]. Available: https://www.youtube.com/watch?v=uEEcw1s_wWk

**Slides 31-32**

1. Atlassian, “git branch,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches

**Slides 32-33**

1. Atlassian, “Creating and merging branches in Git - Git Guides (2020) - YouTube,” YouTube. Accessed: Jan. 28, 2026. [Online]. Available: https://www.youtube.com/watch?v=S2TUommS300
2. Atlassian, “Is GitOps the next big thing in DevOps?,” Atlassian Tutorial. Accessed: Jan. 28, 2026. [Online]. Available: https://www.atlassian.com/git/tutorials/gitops

# References

**Slides 32-33 continued**


3. Atlassian, “What is DevOps?,” Atlassian Tutorial. Accessed: Jan. 28, 2026. [Online]. Available: https://www.atlassian.com/devops
4. Atlassian, “git checkout,” Atlassian Tutorial. Accessed: Jan. 28, 2026. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-checkout
5. Atlassian, “Merging vs. Rebasing,” Atlassian Tutorial. Accessed: Jan. 29, 2026. [Online]. Available: https://www.atlassian.com/git/tutorials/merging-vs-rebasing
6. GitHub, “What is DevOps?” Accessed: Jan. 28, 2026. [Online]. Available: https://github.com/resources/articles/what-is-devops
7. GitHub, “About branches - GitHub Docs.” Accessed: Jan. 28, 2026. [Online]. Available: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches

# References


**Slide 34**

1. GitHub, “About pull requests,” GitHub Docs. Accessed: Jan. 28, 2026. [Online]. Available: https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests

**Glossary**

1. Inc. Amazon Web Services, “What is an IDE (Integrated Development Environment)?” Accessed: Nov. 09, 2025. [Online]. Available: https://aws.amazon.com/what-is/ide/
2. anukruti16, “What is P2P (Peer-to-Peer Process)?,” GeeksforGeeks. Accessed: Nov. 02, 2024. [Online]. Available: https://www.geeksforgeeks.org/what-is-p2p-peer-to-peer-process/
3. Apple, “Keyboard shortcuts in Terminal on Mac,” Terminal User Guide. Accessed: Nov. 02, 2024. [Online]. Available: https://support.apple.com/guide/terminal/keyboard-shortcuts-trmlshtcts/mac

# References

## Glossary continued


4. Atlasian, “git clone,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone

5. Atlassian, “git checkout,” Atlassian Tutorial. Accessed: Jan. 28, 2026. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-checkout

6. Atlassian, “What is version control,” Atlassian Tutorials. Accessed: Oct. 14, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-version-control

7. Atlassian, “git fetch,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-fetch

8. Atlassian, “git pull,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/syncing/git-pull

# References

## Glossary continued


9. Atlassian, “git merge,” Atlassian Tutorial. Accessed: Oct. 28, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches/git-merge

10. Atlassian, “Git cheat sheet,” Atlassian Tutorial. Accessed: Nov. 02, 2024. [Online]. Available: https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet

11. Atlassian, “git branch,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches

12. Atlassian, “git branch,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/using-branches

13. Atlassian, “git rebase,” Atlassian Tutorial. Accessed: Apr. 26, 2025. [Online]. Available: https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase

14. M. Balasundaram, “Bash keyboard shortcuts,” GitHub Gist. Accessed: Nov. 02, 2024. [Online]. Available: https://gist.github.com/tuxfight3r/60051ac67c5f0445efee

# References

## Glossary continued


15. M. Buenning, “What is a Computer Patch?,” ninjaOne. Accessed: Oct. 05, 2025. [Online]. Available: https://www.ninjaone.com/blog/what-is-a-computer-patch/

16. G. Foster, “Git add, commit, and push,” Graphite. Accessed: Nov. 02, 2024. [Online]. Available: https://graphite.dev/guides/git-add-commit-push

17. G. Foster, “Understanding and using the `git restore` command,” Graphite. Accessed: Jan. 24, 2026. [Online]. Available: https://graphite.com/guides/git-restore

17. GeeksforGeeks Contributor, “Difference between Terminal, Console, Shell, and Command Line.” Accessed: Apr. 25, 2025. [Online]. Available: https://www.geeksforgeeks.org/difference-between-terminal-console-shell-and-command-line/

18. Git-SCM, “git mv,” Git Documentation. Accessed: Oct. 05, 2025. [Online]. Available: https://git-scm.com/docs/git-mv

# References

## Glossary continued


19. Git-SCM, “git reset,” Git Documentation. Accessed: Oct. 05, 2025. [Online]. Available: https://git-scm.com/docs/git-reset

20. GitHub, “About authentication to GitHub,” GitHub Docs. Accessed: Oct. 19, 2024. [Online]. Available: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/about-authentication-to-github

21. GitHub, “Set up Git - Authenticating with GitHub from Git,” GitHub Docs. Accessed: Oct. 19, 2024. [Online]. Available: https://docs.github.com/en/get-started/getting-started-with-git/set-up-git#authenticating-with-github-from-git

22. GitHub, “Generating a new SSH key and adding it to the ssh-agent - Generating a new SSH key,” GitHub Docs. Accessed: Oct. 19, 2024. [Online]. Available: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key

# References

## Glossary continued


23. GitHub, “Managing your personal access tokens,” GitHub Docs. Accessed: Oct. 28, 2024. [Online]. Available: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens

24. T. Gunther, “Command Line Cheat Sheet | Tower Blog.” Accessed: Apr. 25, 2025. [Online]. Available: https://www.git-tower.com/blog/command-line-cheat-sheet

25. Lenova, “What is a GUI?” Accessed: Nov. 09, 2025. [Online]. Available: https://www.lenovo.com/us/en/glossary/what-is-a-gui/

26. Paessler GmbH, “Server - Definition and details,” IT Explained. Accessed: Nov. 02, 2024. [Online]. Available: https://www.paessler.com/it-explained/server

27. Y. Singh, “Difference between SSH and HTTPS in GitHub,” Medium. Accessed: Oct. 19, 2024. [Online]. Available: https://ysrazsingh.medium.com/difference-between-ssh-and-https-in-github-95ea7cf38f4c

# References

## Glossary continued


28. SNIA, “What is Mirroring?,” SNIA. Accessed: Oct. 05, 2025. [Online]. Available: https://www.snia.org/education/what-is-mirroring

29. R. Torruellas, “Vim Cheat Sheet,” Vim. Accessed: Nov. 02, 2024. [Online]. Available: https://vim.rtorr.com/

30. “Distributed version control,” Wikipedia. Accessed: Oct. 14, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Distributed_version_control