Using Git and GitHub

EECS 348 Lab 1 — 1/30/2025

Harlan Williams

Electrical Engineering and Computer Science
University of Kansas

Info

- Email — hrw@ku.edu
- Office hours — Wednesdays 10:00 A.M.–12:00 P.M. Eaton 2011
- My Website people.eecs.ku.edu/˜h054w684/ — for slides
- Check Canvas for assignments and rubrics
- Some basic policy
  - Assignments are due 11:59 P.M. on Canvas one week after they are given in lab
  - You can make up a lab as long as I'm informed you will be missing it ahead of time
  - No late work will be accepted
  - Not taking attendance

Configuring Git

First time users will need to do some configuring

- `git config --global user.name <your name>`
- `git config --global user.email <your email>`

You can check your config with `git config -l`

For more help/info, see:
git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

Creating a Git repository and adding files

First step is creating an empty directory and running `git init -b main`

- Creates a new hidden directory called `.git/`
- Any existing files in the repository will be untracked
- Untracked files can be tracked by adding them with `git add <file>`
- From there, Git will track any modifications to those files

Basic workflow

- Modified files are added to a *stage* and then snapshotted in a *commit*
- Files can be either tracked or untracked
- Modifications to tracked files will be picked up by Git
- Untracked and modified files can be staged to be snapshotted with `git add <file>` or `git add .`
- Staged files can be snapshotted with `git commit -m <msg>` to save their state
- Commits are stored in a log which makes it easy to see a history of changes

Using Git to fearlessly make changes to your code

- Can go back to previous commits with `git checkout <commit hash>`
- Want to add a new feature without worrying about undoing it if it doesn't work out? Try `git branch`.
- If you want to keep the changes, use `git merge` to merge that branch and your main or master branch
- If it doesn't work out, delete the branch if you no longer want it
- Merging may introduce merge conflicts—areas where changes cannot be cleanly inserted
- Merge conflicts can be resolved with your editor of choice

Uploading code to GitHub

- You can save your Git repositories to GitHub to access them from anywhere—or share them with anyone
- GitHub offers a GUI and a CLI application to streamline using Git for GitHub
- After KU, most employers will want to see your GitHub with any projects you may have worked on

Problems connecting? See docs.github.com/en/authentication/connecting-to-github-with-ssh/about-ssh

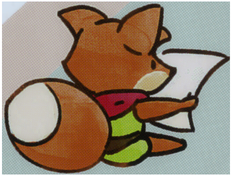
Lost? See...

The Git program comes with a lot of documentation. Use `git help` for top-level info. Additionally, `-h` can be appended to any Git command for more specific help

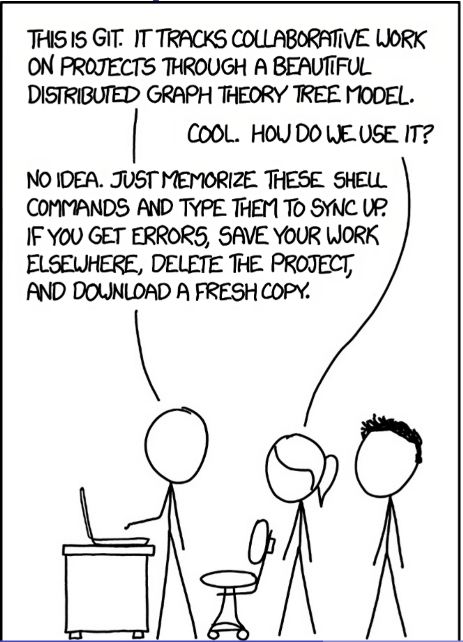
Git book git-scm.com/book/en/v2

More user-friendly tools for interacting with GitHub

- github.com/apps/desktop — GUI application
- cli.github.com/ — terminal utility



Summary


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Useful git commands to know

- `git config`
- `git init`
- `git add`
- `git commit`
- `git log`
- `git checkout`
- `git branch`
- `git merge`
- `git pull`
- `git push`
- `git help`