LAB #1

Adv. CiC
Spring 2026

Agenda

1. About me
2. Housekeeping
3. Review: Git
4. Workflow & Best Practices
5. Exercise

THE TWO STATES OF EVERY PROGRAMMER
I AM A GOD.
I HAVE NO IDEA WHAT I'M DOING.

About Me

- **Grew up in Michigan**
  - Originally from India
- **~4 years in NYC**
  - ~3.5 years in SF before
- **Pronouns: she/her**

- **Second-year MPA @ SIPA**
  - CEE, Tech Policy
- **Computer Science Engineering @ UMich**
  - Go Blue!
- **Work: ~7 years @ Meta**
  - Social Impact Org
  - Tech Lead / Senior Software Engineer

Some things I like: dancing, spicy food, tennis, political satire, new leaves on my plants

Housekeeping

Course Organization

See the Course Website!

This is the source-of-truth for basically everything.

Grading Components

1. Attendance
2. Labs
3. Reading responses
4. Projects

No final exam.

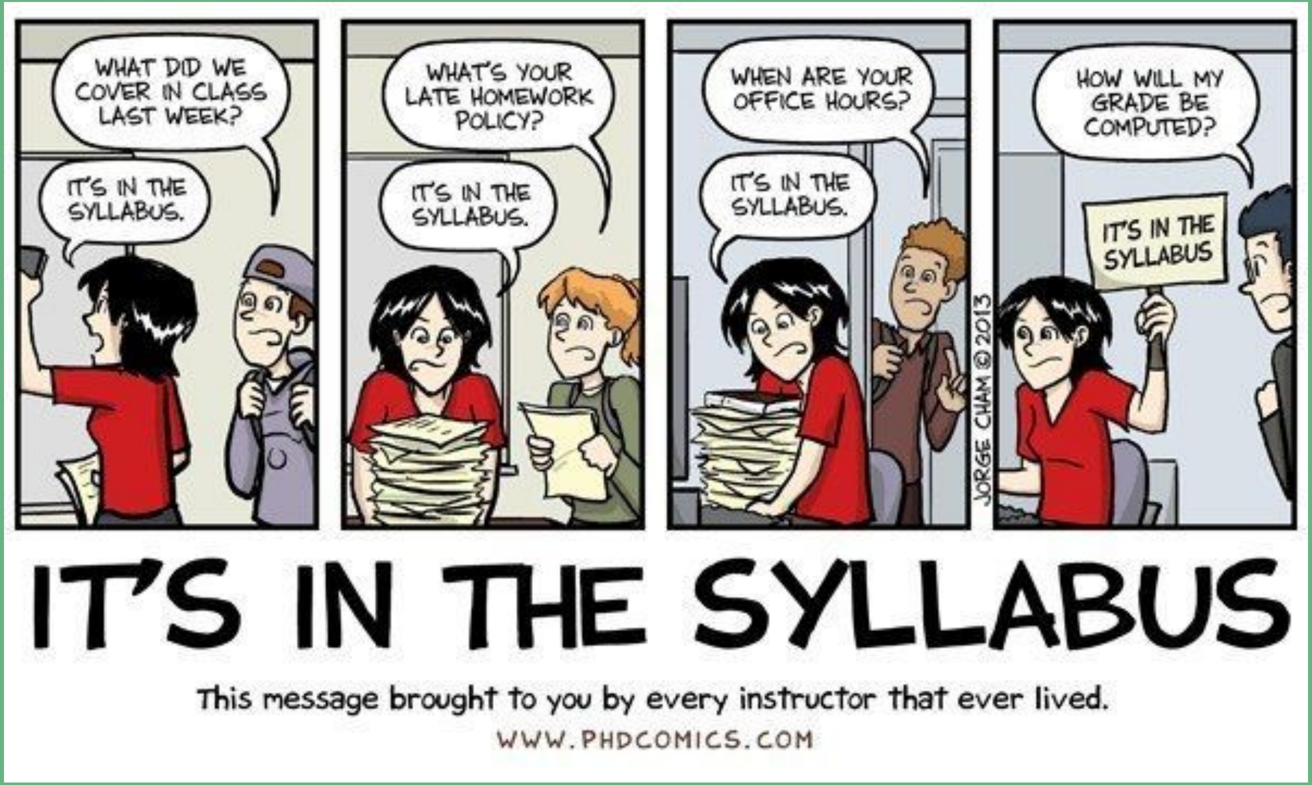
Adv. CiC Prerequisites

Computing in Context *(or equivalent placement test)*

VSCode & Git/GitHub should be set up.

Additional Resources

1. Online tools (Google, ChatGPT, etc.)
2. **ED!!!**
3. Office hours


WHAT DID WE COVER IN CLASS LAST WEEK?
IT'S IN THE SYLLABUS.
WHAT'S YOUR LATE HOMEWORK POLICY?
IT'S IN THE SYLLABUS.
WHEN ARE YOUR OFFICE HOURS?
IT'S IN THE SYLLABUS.
HOW WILL MY GRADE BE COMPUTED?
IT'S IN THE SYLLABUS
JORGE CHAM © 2013
IT'S IN THE SYLLABUS
This message brought to you by every instructor that ever lived.
WWW.PHDCOMICS.COM

Questions?

Review: GIT

What is GIT?

a distributed, version control system

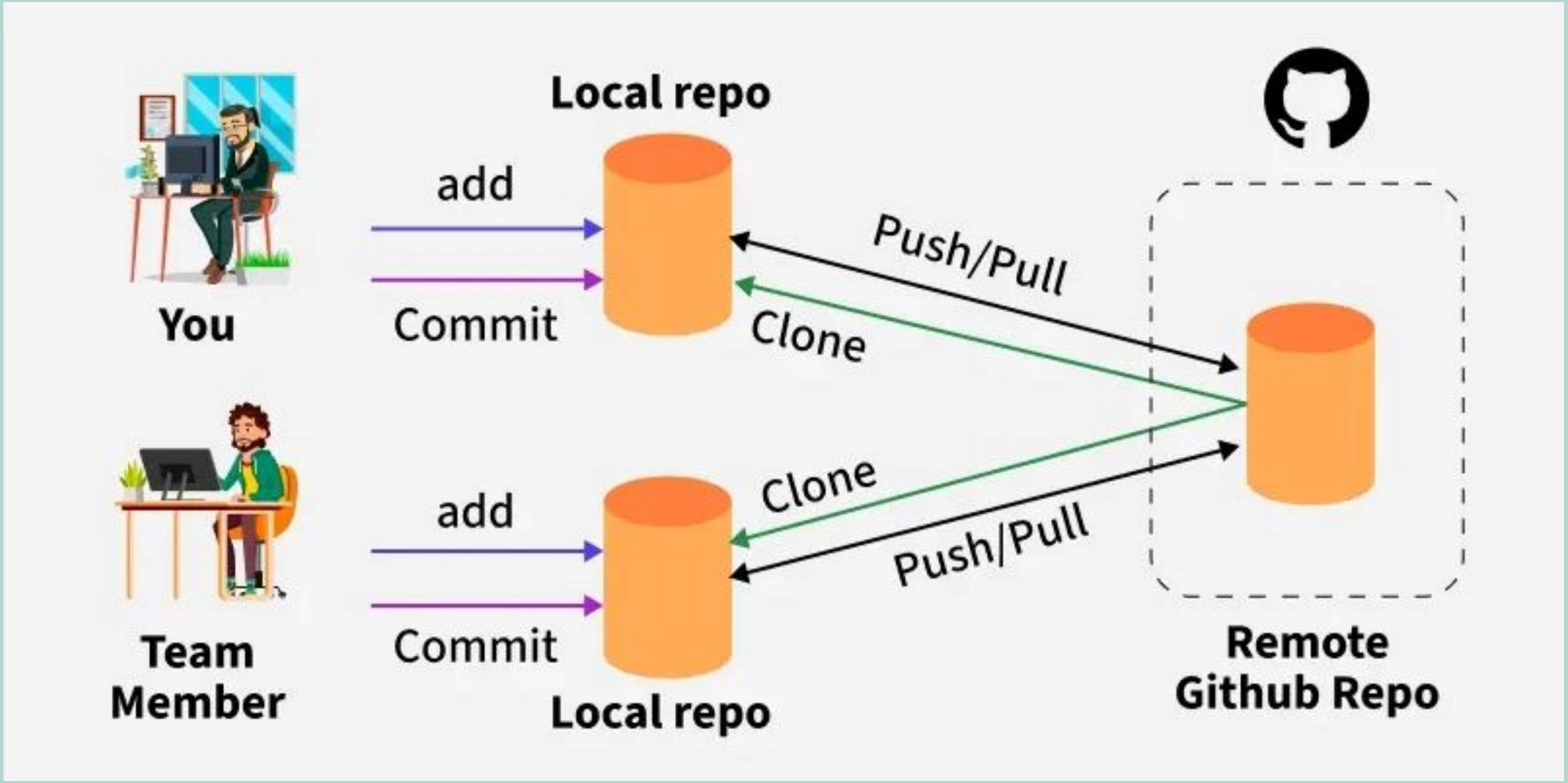
What is GIT?

“Distributed”

- Every contributor has a full copy of the project’s code & history on their own machine
- No single central server is required to work or save progress
- You can sync your repo with a remote repo (on the server) as needed

Why does this matter?

- If you move to a different computer, you can still easily pick up the code
- If your partner makes changes, you can easily pick them up
- **Resilience:** work continues even if one computer fails
- **Collaboration:** easier & more flexible


Local repo
add
Commit
You
Push/Pull
Clone
Team Member
add
Commit
Local repo
Clone
Push/Pull
Remote Github Repo

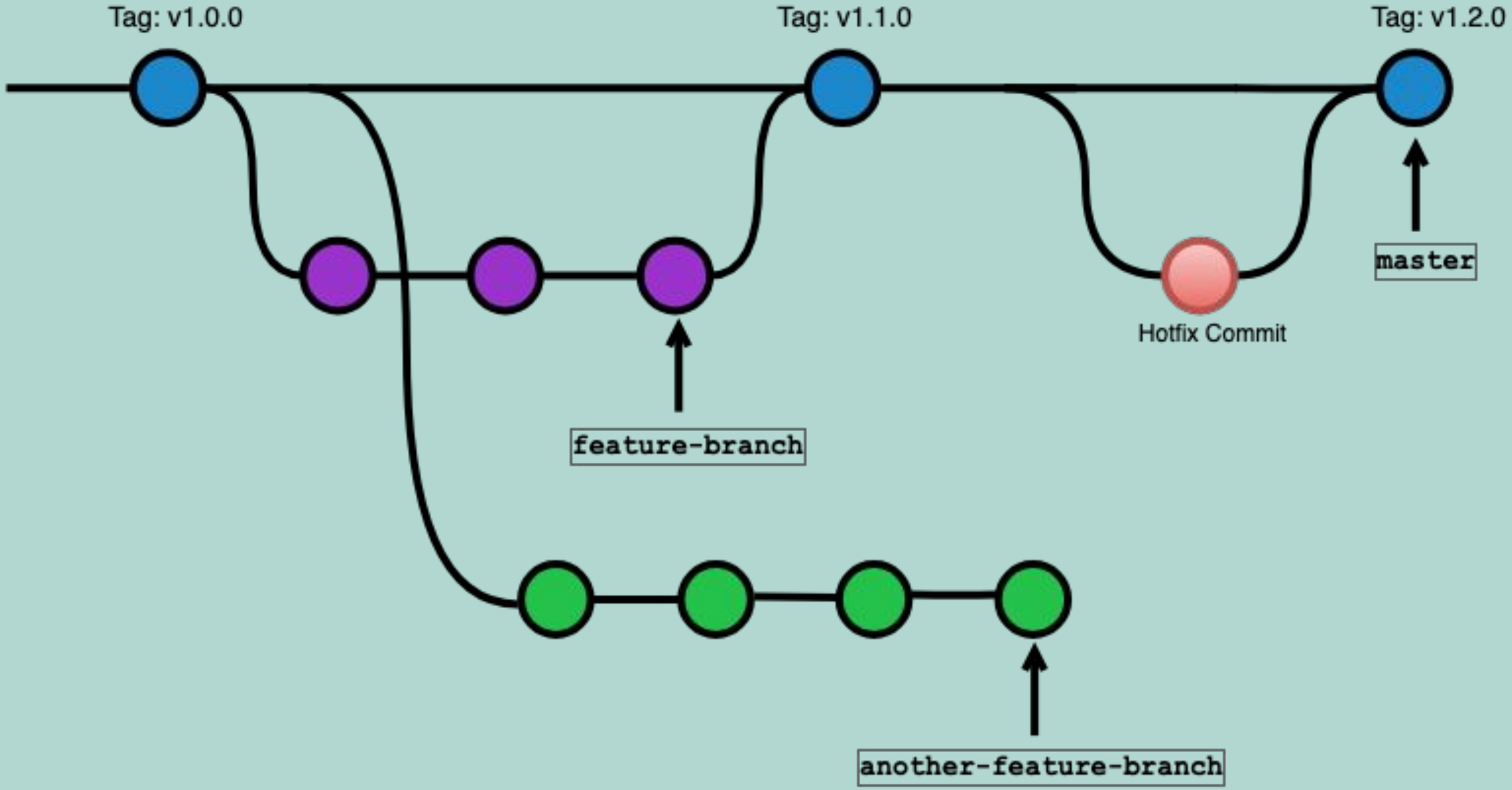
What is GIT?

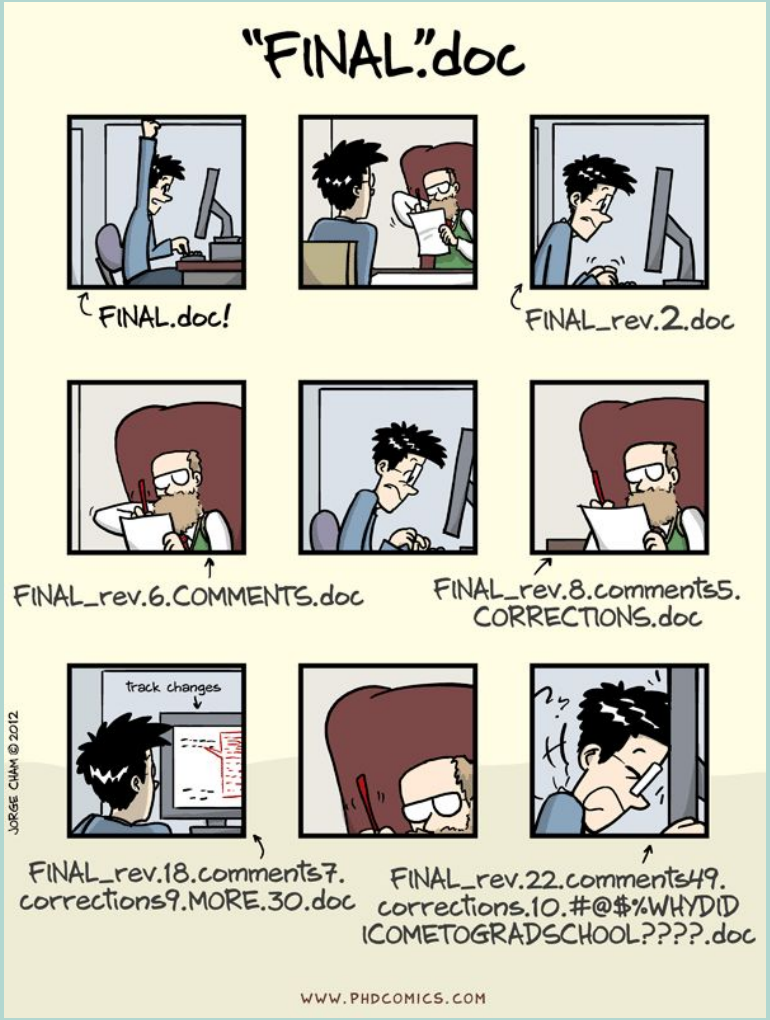
“Version Control”

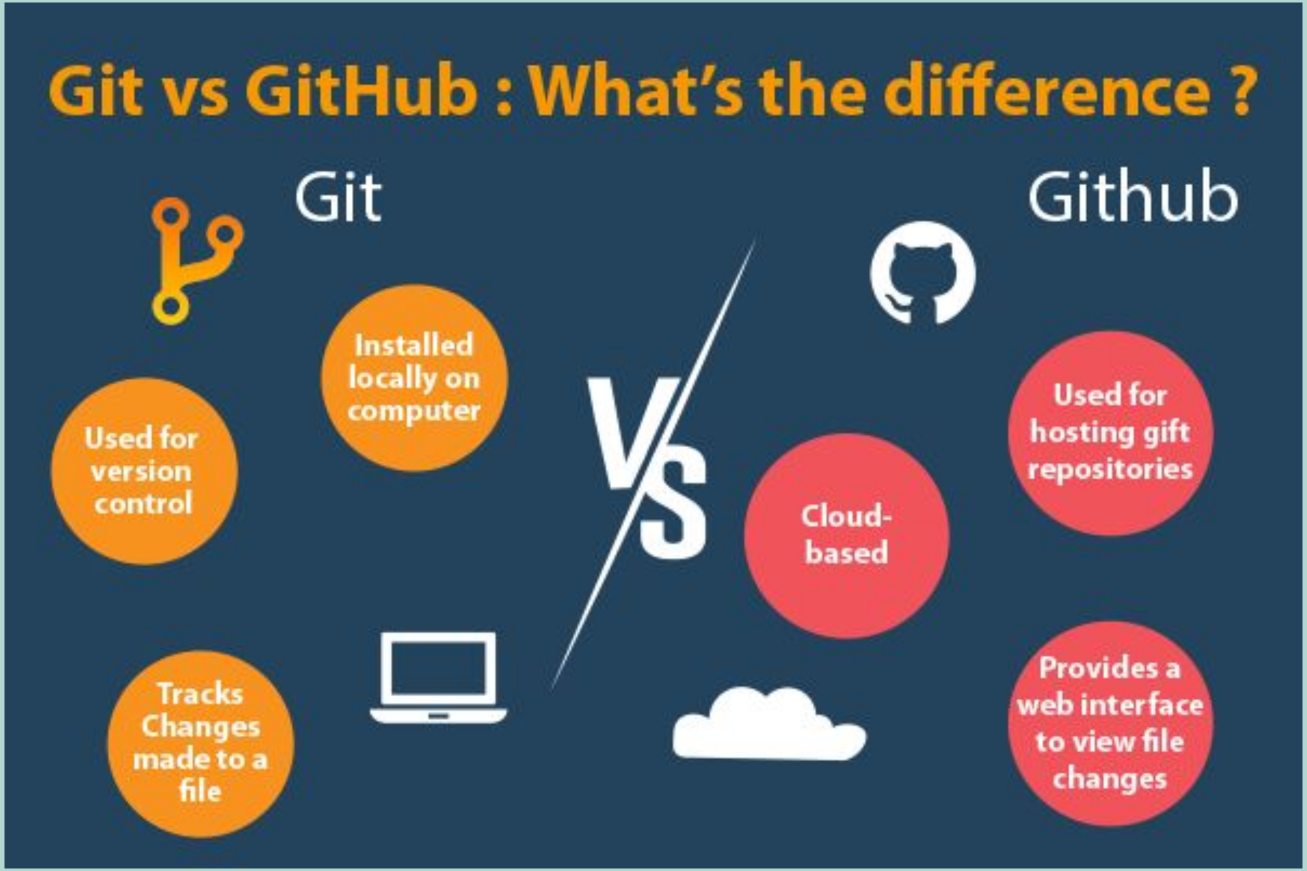
- Tracks changes to files over time (history)
- Allows you to compare, revert, or merge different versions of code
- Makes collaboration safer by managing conflicts between contributors

Why does this matter?

- You can undo mistakes or return to earlier versions
- Changes are easier to review and manage
  - Multiple people can work on the same project safely
- **Safety**: mistakes are reversible
- **Coordination**: people don’t overwrite each other’s work


Tag: v1.0.0
Tag: v1.1.0
Tag: v1.2.0
master
Hotfix Commit
feature-branch
another-feature-branch


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM



Git vs GitHub : What's the difference ?

Git

Installed locally on computer

Used for version control

Tracks Changes made to a file

VS

Github

Used for hosting gift repositories

Cloud-based

Provides a web interface to view file changes

Workflow & Best Practices

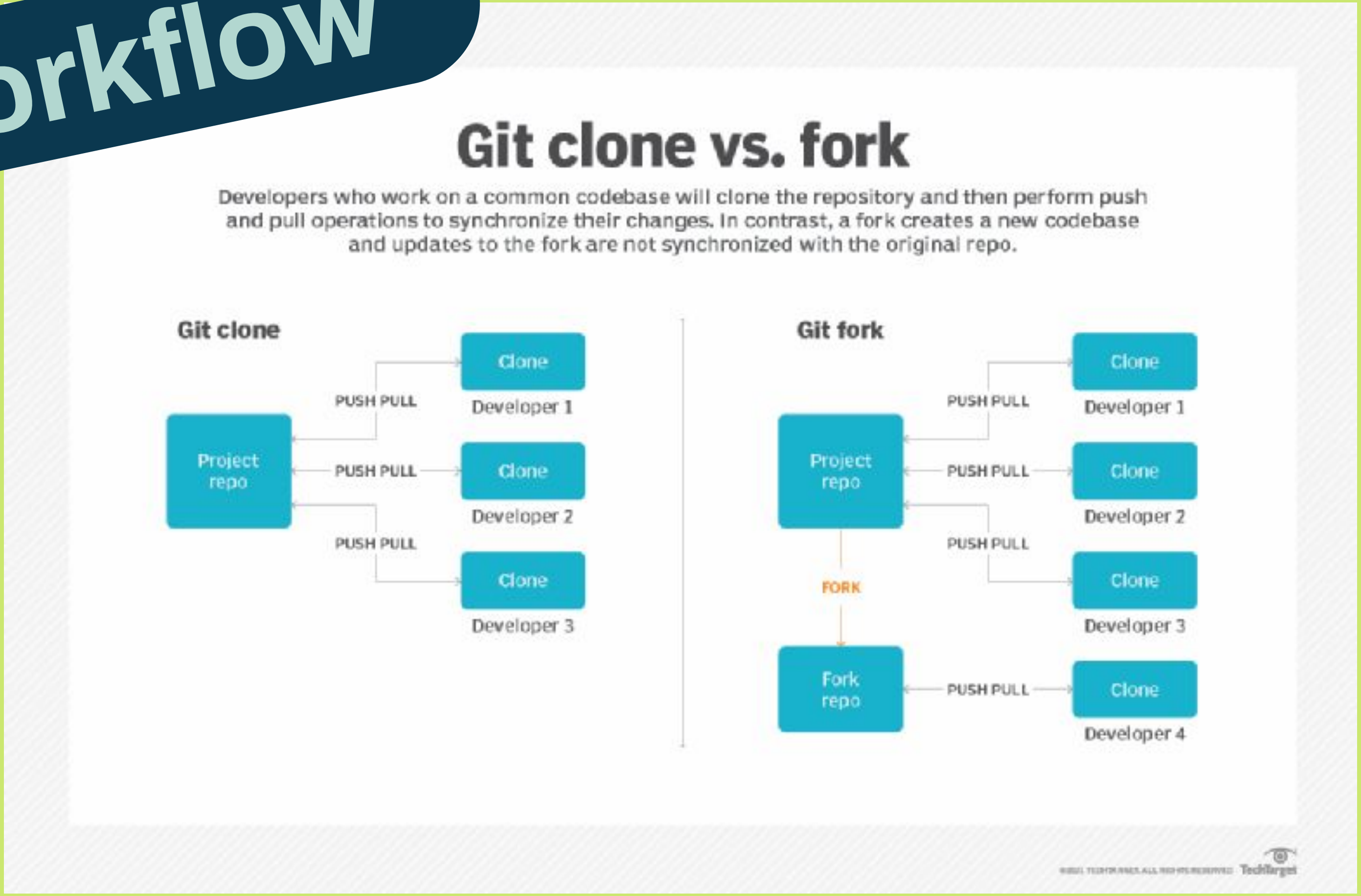
Basic Terminolog y

- **Repository (repo):** A project folder that stores the code and its history
  - **Local** vs. **remote (server)**
- **Clone:** Make your own copy of a repository on your computer
- **Fork:** Create a separate, independent copy of someone else’s project
- **Commit:** Save a snapshot of your changes (with a short note)
- **History:** A timeline of all changes/commits

Basic Terminolog y (cont.)

- **Pull & push:** Pull gets updates from the server; push sends your changes to it
- **Branch:** A separate line of work to try changes without affecting the main version
- **Pull Request (PR):** Ask to add your changes to the main project repo
- **Merge:** Combine changes into one version

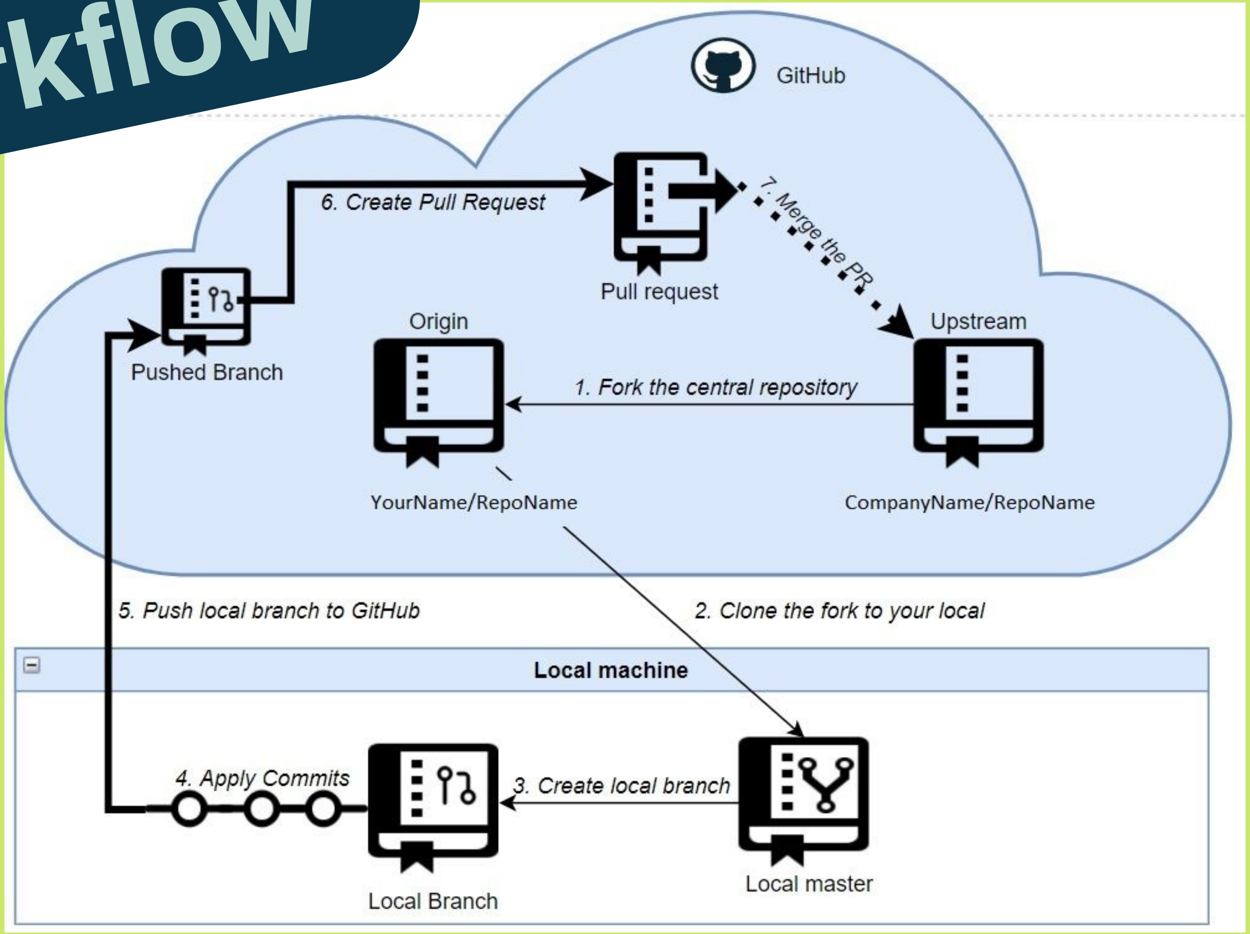
Workflow



Git clone vs. fork

Developers who work on a common codebase will clone the repository and then perform push and pull operations to synchronize their changes. In contrast, a fork creates a new codebase and updates to the fork are not synchronized with the original repo.

Workflow


GitHub
6. Create Pull Request
Pull request
7. Merge the PR
Pushed Branch
Origin
Upstream
1. Fork the central repository
YourName/RepoName
CompanyName/RepoName
5. Push local branch to GitHub
2. Clone the fork to your local
Local machine
4. Apply Commits
3. Create local branch
Local Branch
Local master

Best Practices

1. Write clear commit messages

- Explain *what* you changed and *why* in a short, clear message
- Make sure descriptions are easily understandable

2. Keep commits small & focused

- Each commit should do one clear thing
- Makes changes easier to understand, review, and undo

3. Review PRs carefully

- Read the code, not just the summary
- Check that changes are correct, clear, and don't break other parts

4. Pull often before you push

- Get the latest changes before adding yours
- Reduces conflicts and surprises when collaborating

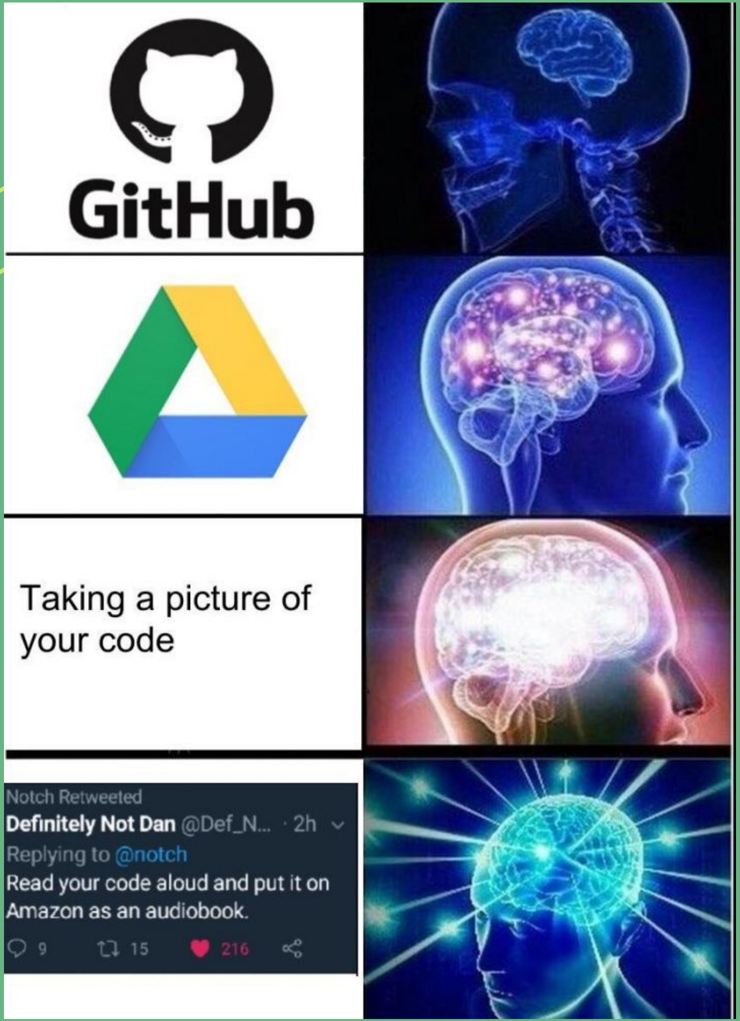
What Makes a Good Pull Request

Relevant to the Project

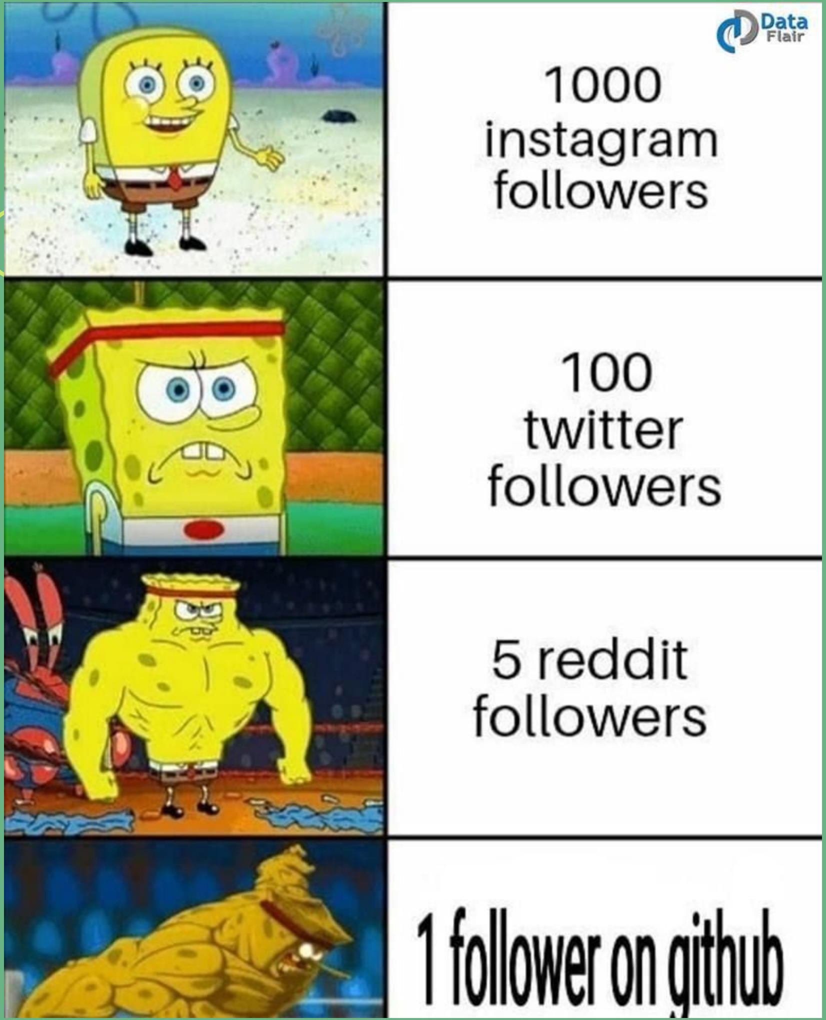
Well Documented

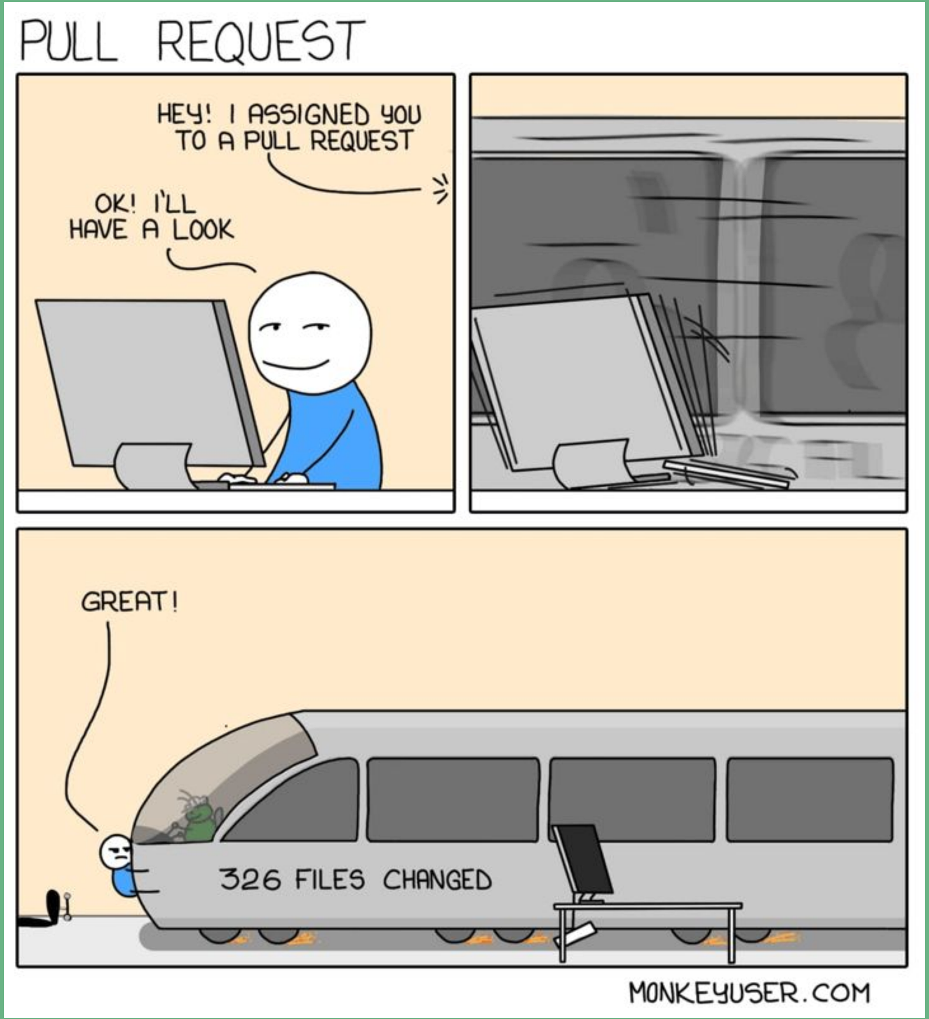
Easy to Review

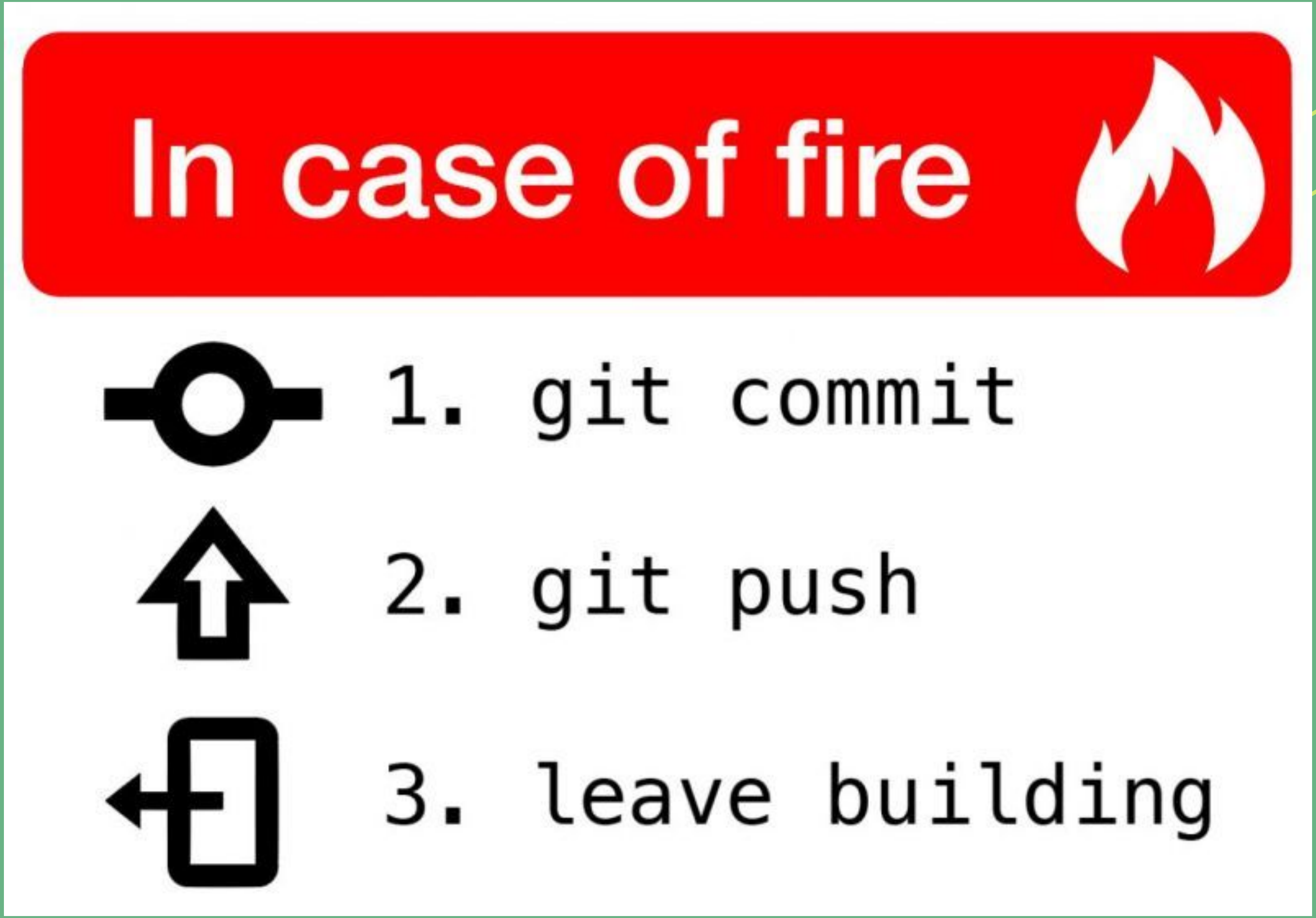
Adhere to Code Standards



Notch Retweeted
Definitely Not Dan @Def_N... · 2h
Replying to @notch
Read your code aloud and put it on Amazon as an audiobook.
9 15 216


Data Flair
1000 instagram followers
100 twitter followers
5 reddit followers
1 follower on github


PULL REQUEST
HEY! I ASSIGNED YOU TO A PULL REQUEST
OK! I'LL HAVE A LOOK
GREAT!
326 FILES CHANGED
MONKEYUSER.COM



In case of fire

1. git commit

2. git push

3. leave building

EXERCIS
E

Lfg