LAB #2

Adv. CiC
Spring 2026

Agenda

1. Housekeeping
2. Merge Conflicts
3. UI vs. Terminal
4. Review: Project Management
5. Lab 2 Exercise

Housekeeping

Due this past week

- Reading 1
- Lab 1

Late policy: up to 1 week for labs & projects; -5%/day

Upcoming

1. Project, pt. 1 - **2/5 @ 3pm**
2. Reading 2 - **2/5 @ 3pm**
3. Lab 2 - **2/6 @ 1pm**

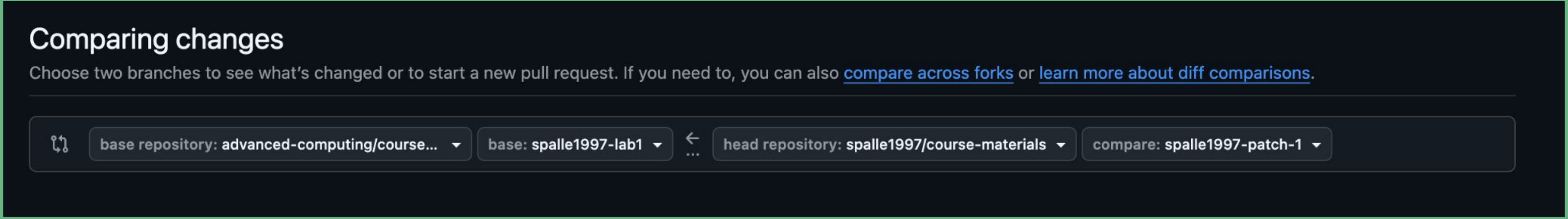
Additional Resources

- Online tools
  - Google, ChatGPT, etc.
  - ***Make sure to review AI policy!***
- **ED!!!**
- Office hours
  - 1 hr/week set time
  - Appts by email

Questions on Lab 1?

Common Issues

- Merging into the wrong branch (instead of main)
- Making a PR into the wrong repo (your fork vs. class repo)
- Syncing your fork but not your local repo


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 repository: advanced-computing/course... base: spalle1997-lab1 ← head repository: spalle1997/course-materials compare: spalle1997-patch-1

What's likely going wrong here? *(w.r.t. Lab 1)*

Merge Conflicts

What are merge conflicts?

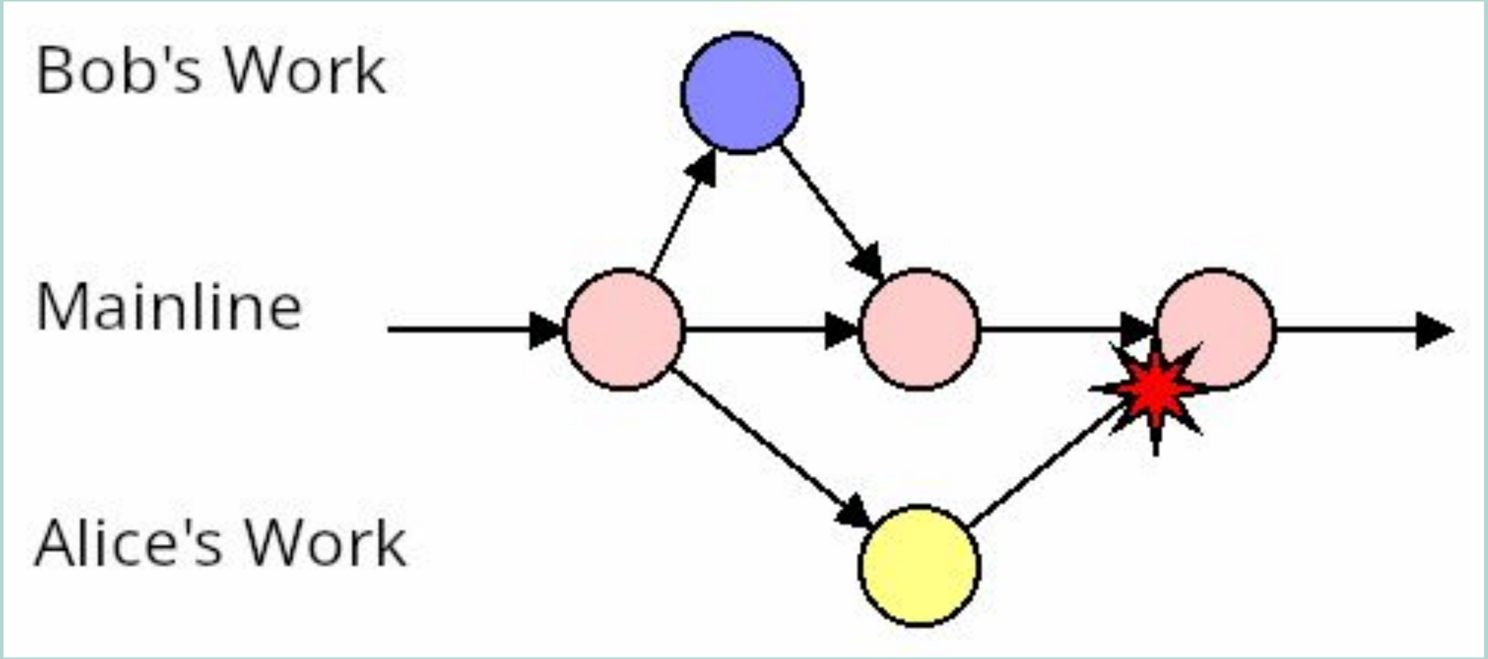
when a version control system can't automatically combine changes from different branches

Merge Conflicts: WHY?

Why do they occur?

- **Simultaneous edits:** People change the same lines on different branches.
- **Delete vs. edit:** One branch deletes a file that another branch modifies.
- **Overlapping changes:** Both branches alter the same code section differently.

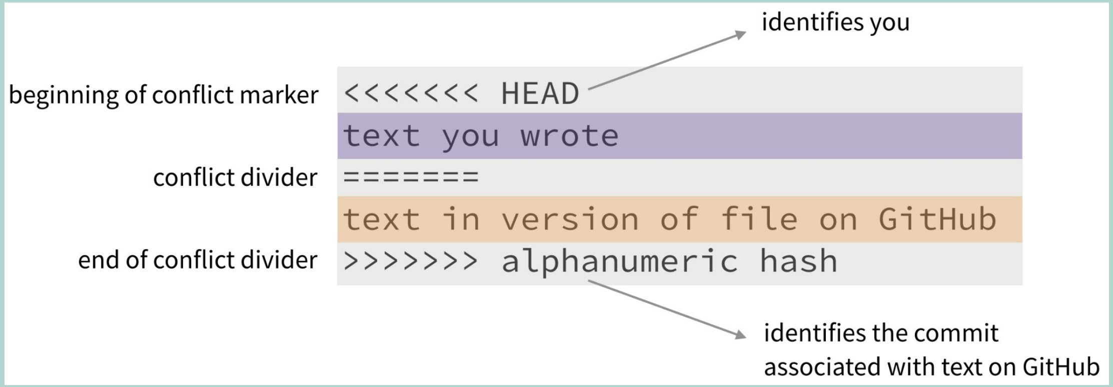
he is fellow collaborator, he is pain in the ass
i add new file to repository, he adds new file to repository
i commit new changes, he commits new changes
i push my commits, he has to resolve merge conflicts, great success


Bob's Work
Mainline
Alice's Work

Merge Conflicts: WHAT?

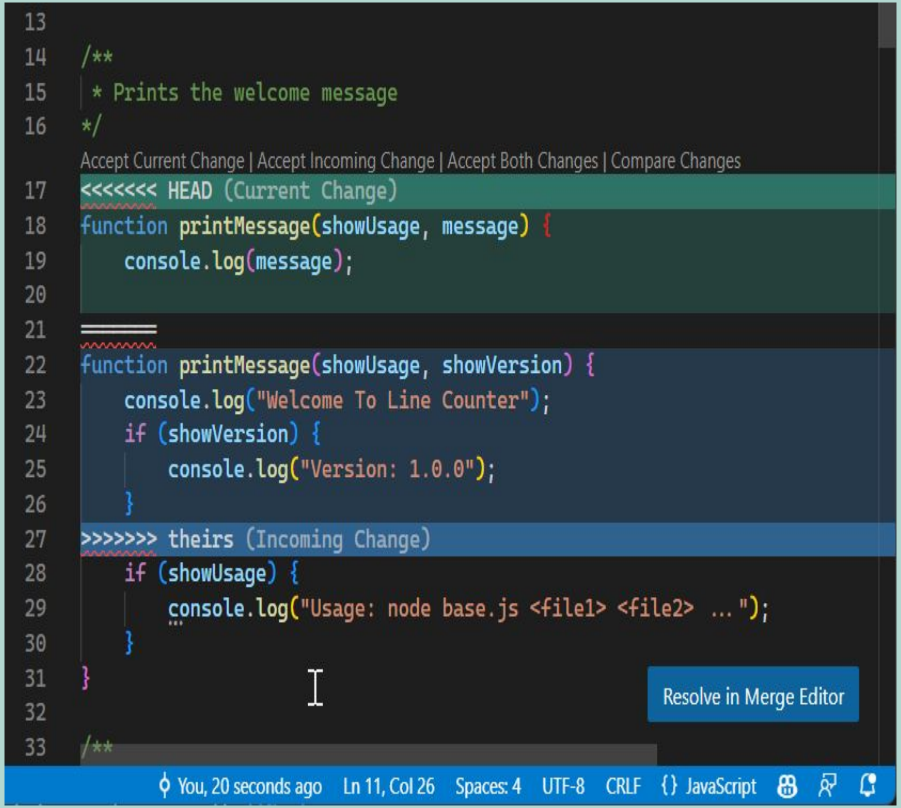
What do they look like?

- When you try to merge, Git will **stop the process and tell you** which files have conflicts
- Conflicted files contain **markers** like **<<<<<<<, =======, >>>>>>>**



```
<<<<<<< HEAD
text you wrote
=======
text in version of file on GitHub
>>>>>>> alphanumeric hash
```

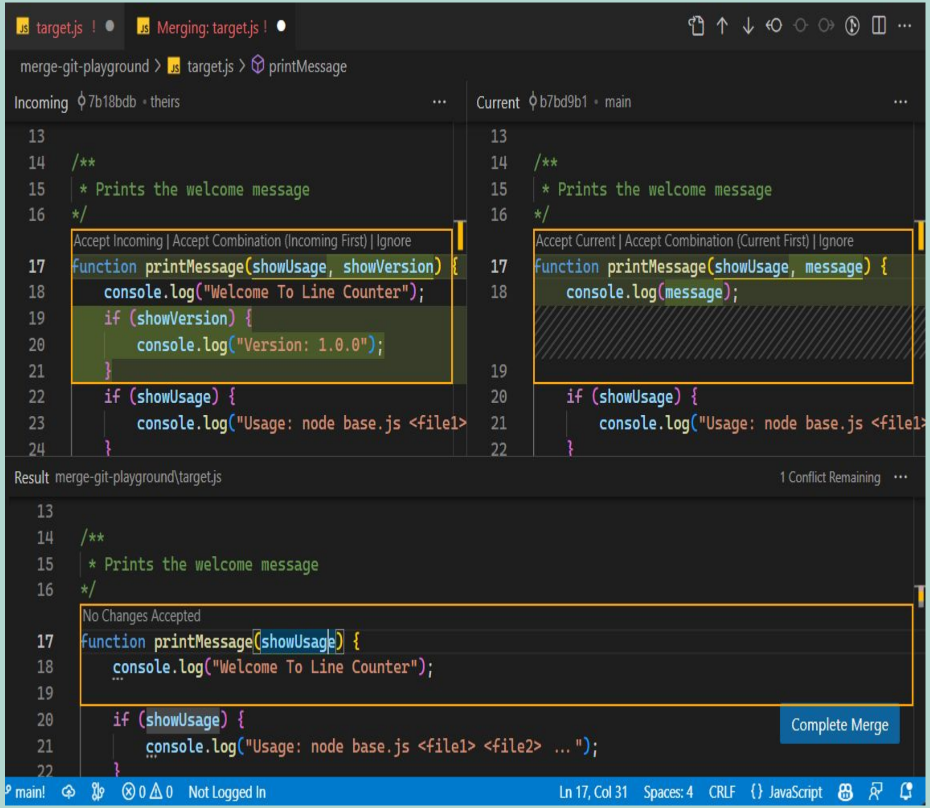
identifies you
beginning of conflict marker
conflict divider
end of conflict divider
identifies the commit associated with text on GitHub



```
/**
 * Prints the welcome message
*/
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
function printMessage(showUsage, message) {
    console.log(message);

=======
function printMessage(showUsage, showVersion) {
    console.log("Welcome To Line Counter");
    if (showVersion) {
        console.log("Version: 1.0.0");
    }
>>>>>>> theirs (Incoming Change)
    if (showUsage) {
        console.log("Usage: node base.js <file1> <file2> ...");
    }
}

/**
```



Incoming 7b18bdb • theirs

```
/**
 * Prints the welcome message
*/
Accept Incoming | Accept Combination (Incoming First) | Ignore
function printMessage(showUsage, showVersion) {
    console.log("Welcome To Line Counter");
    if (showVersion) {
        console.log("Version: 1.0.0");
    }
    if (showUsage) {
        console.log("Usage: node base.js <file1>
    }
```

Current b7bd9b1 • main

```
/**
 * Prints the welcome message
*/
Accept Current | Accept Combination (Current First) | Ignore
function printMessage(showUsage, message) {
    console.log(message);

    if (showUsage) {
        console.log("Usage: node base.js <file1>
    }
```

Result merge-git-playground\target.js — 1 Conflict Remaining

```
/**
 * Prints the welcome message
*/
No Changes Accepted
function printMessage(showUsage) {
    console.log("Welcome To Line Counter");

    if (showUsage) {
        console.log("Usage: node base.js <file1> <file2> ...");
    }
```

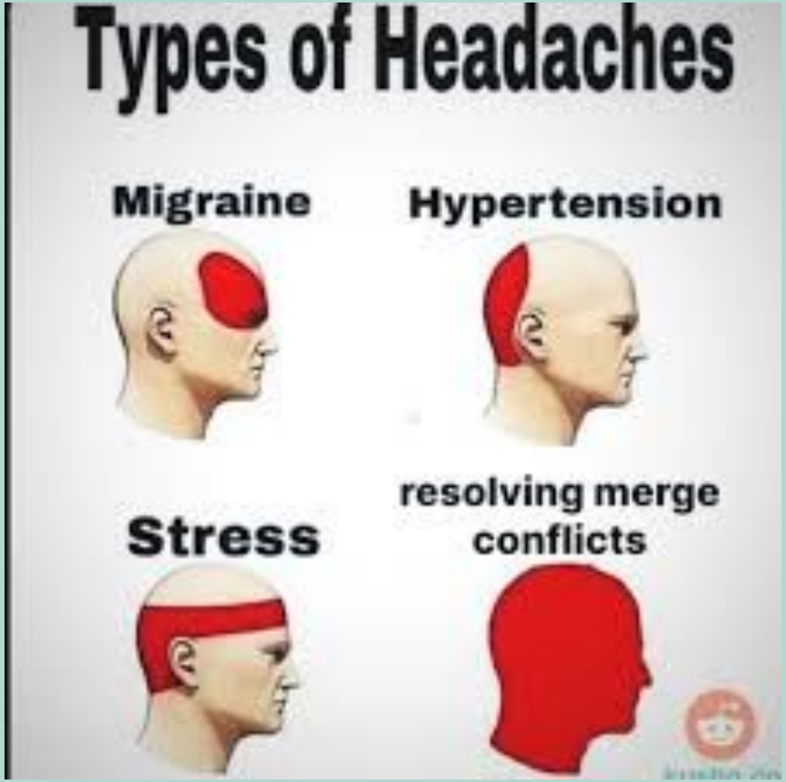
Merge Conflicts: HOW?

How do we resolve them?

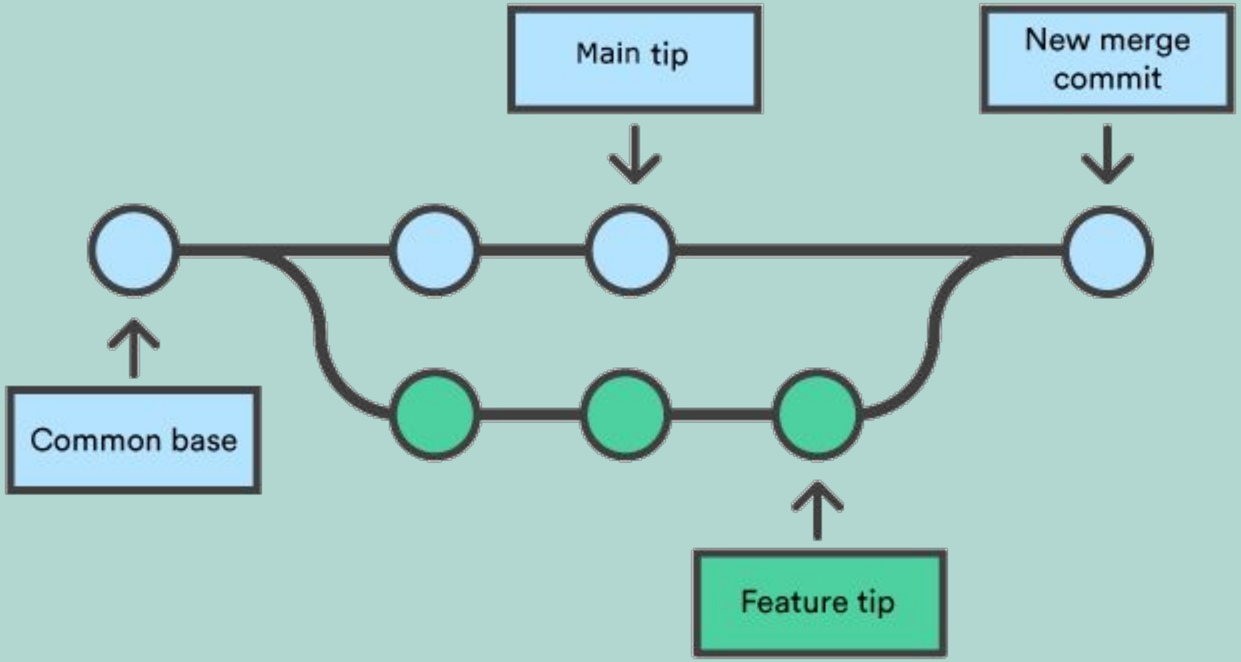
1. Open the conflicted file and review the marked sections.

2. Choose one version or combine changes.

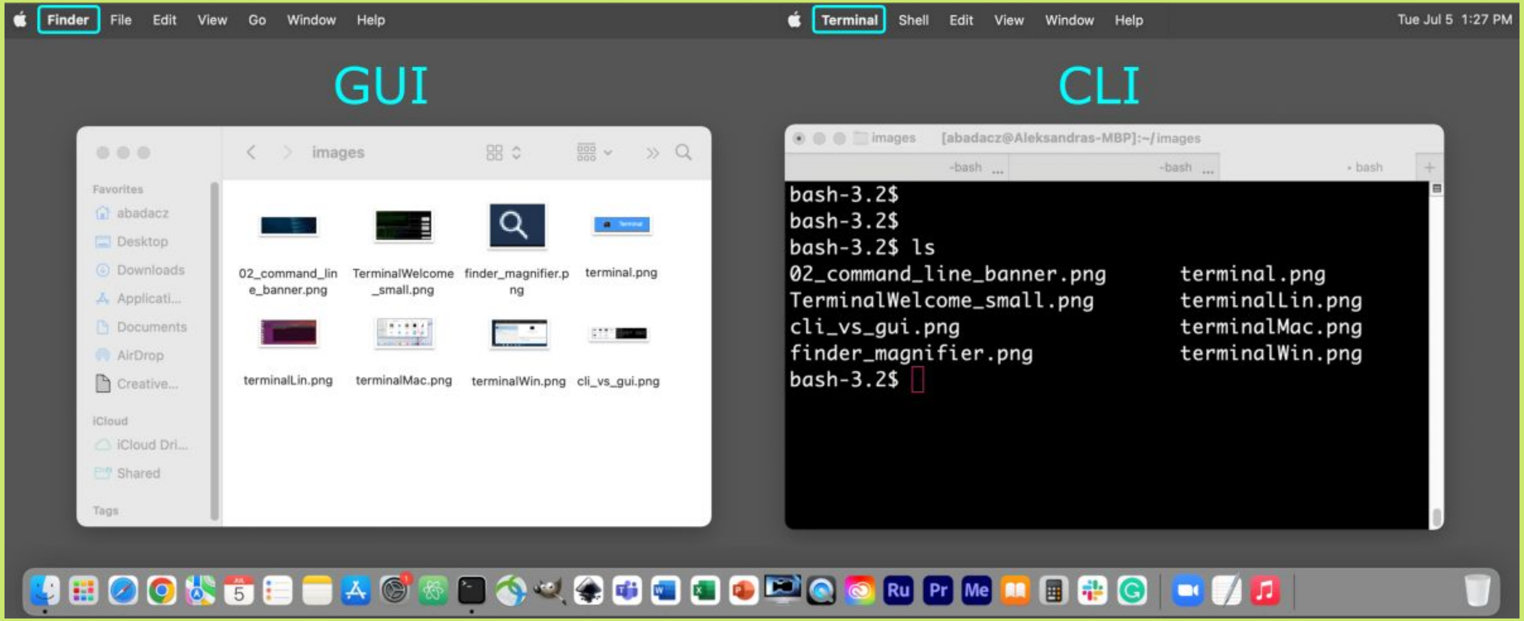
3. Remove markers, (stage the file), and complete the merge.



Accept Incoming
Accept Current
GIT MERGE CONFLICTS
JAKE-CLARK.TUMBLR
imgflip.com


Main tip
New merge commit
Common base
Feature tip

UI vs. Terminal



```
bash-3.2$
bash-3.2$
bash-3.2$ ls
02_command_line_banner.png      terminal.png
TerminalWelcome_small.png       terminalLin.png
cli_vs_gui.png                  terminalMac.png
finder_magnifier.png            terminalWin.png
bash-3.2$
```

Graphical User Interface (GUI)

- **Visual**: buttons, menus, and icons make actions easy to discover
- **Beginner-friendly**: lowers the learning curve for common tasks

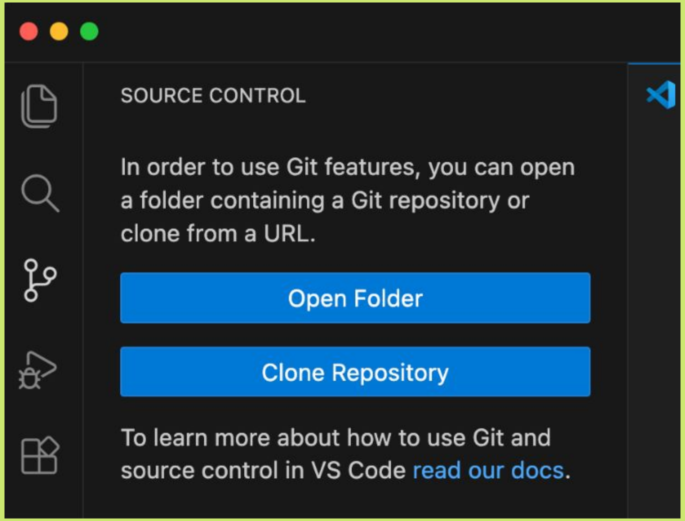
Command Line Interface (CLI)

- **Precise & fast**: direct commands give full control and speed
- **Powerful & scriptable:** enables automation and advanced workflows

Clone

Clone existing repo:

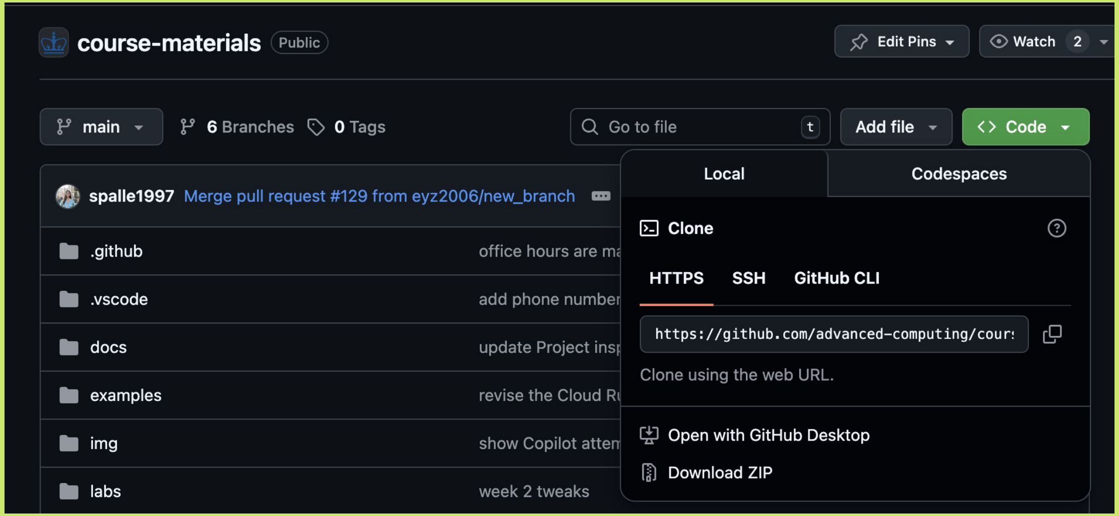
git clone <url>



SOURCE CONTROL

In order to use Git features, you can open a folder containing a Git repository or clone from a URL.

To learn more about how to use Git and source control in VS Code read our docs.



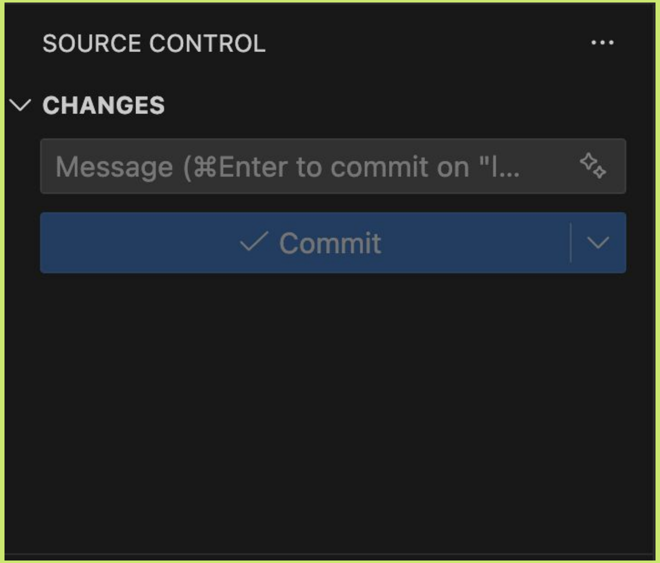
spalle1997 Merge pull request #129 from eyz2006/new_branch

.github	office hours are ma
.vscode	add phone number
docs	update Project insp
examples	revise the Cloud Ru
img	show Copilot attem
labs	week 2 tweaks

```
https://github.com/advanced-computing/cour:
```

Clone using the web URL.

Commit



Commit (staged) changes:

git commit -m <message>

Push/Pull

Push the current branch to (1) its remote “tracking branch” or (2) to *origin main* branch:

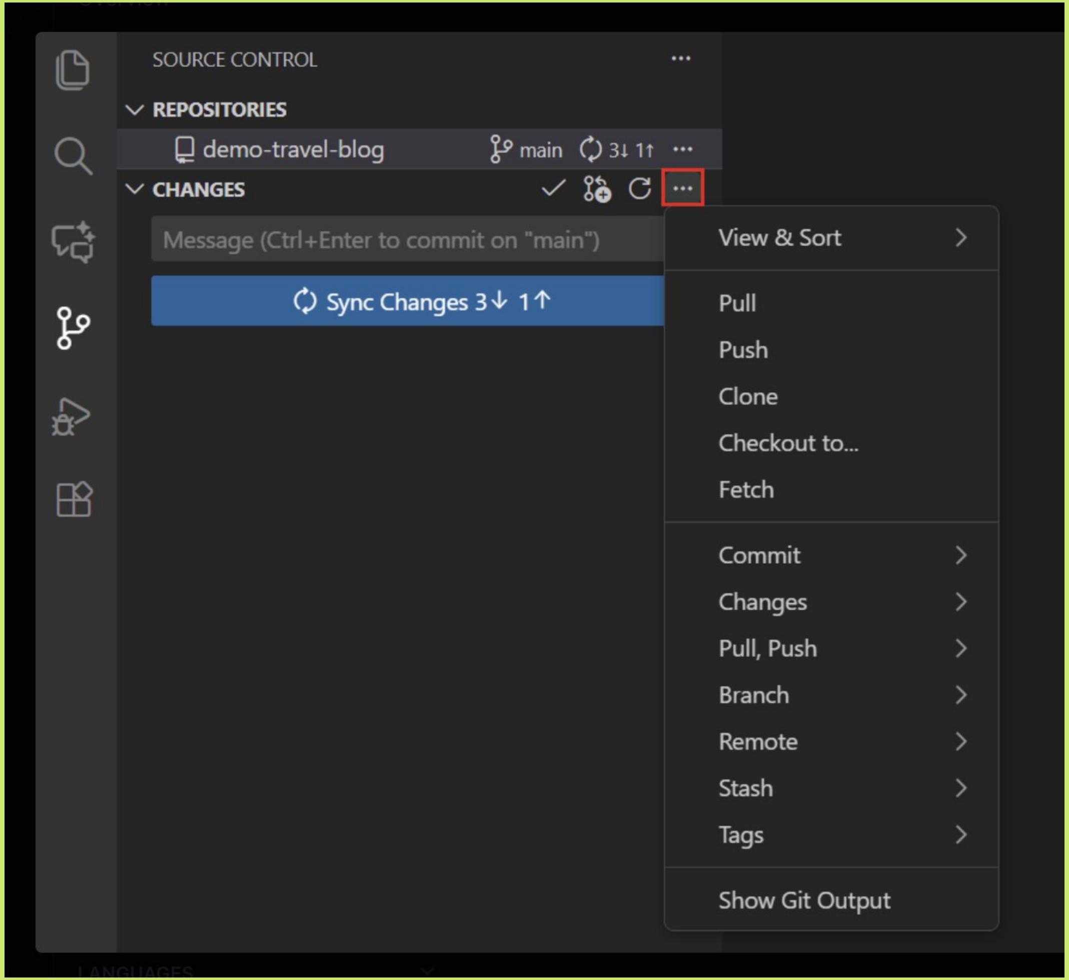
git push

git push origin main

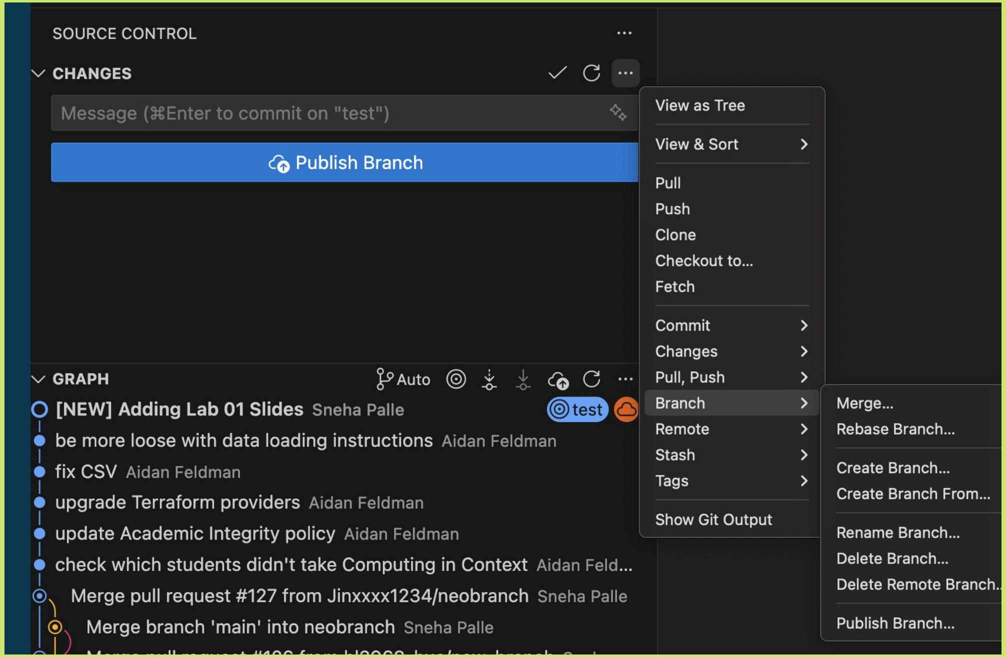
Fetch changes and then merge them into your current branch:

git pull

git pull origin main



Branches



SOURCE CONTROL

CHANGES

Message (⌘Enter to commit on "test")

Publish Branch

[NEW] Adding Lab 01 Slides Sneha Palle
be more loose with data loading instructions Aidan Feldman
fix CSV Aidan Feldman
upgrade Terraform providers Aidan Feldman
update Academic Integrity policy Aidan Feldman
check which students didn't take Computing in Context Aidan Feld...
Merge pull request #127 from Jinxxxx1234/neobranch Sneha Palle
Merge branch 'main' into neobranch Sneha Palle

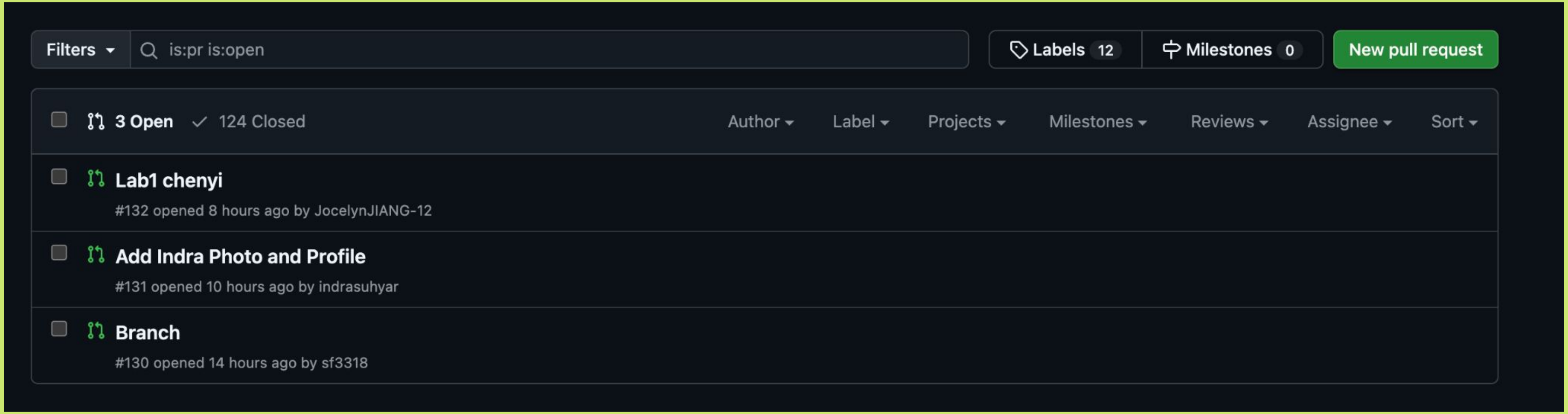
Make a new branch:
git checkout -b <branch>

Switch branches:
git checkout <branch>

Delete a branch (safety check):
git delete -d <branch>

Force delete a branch:
git delete -D <branch>

GitHub UI: Pull Requests

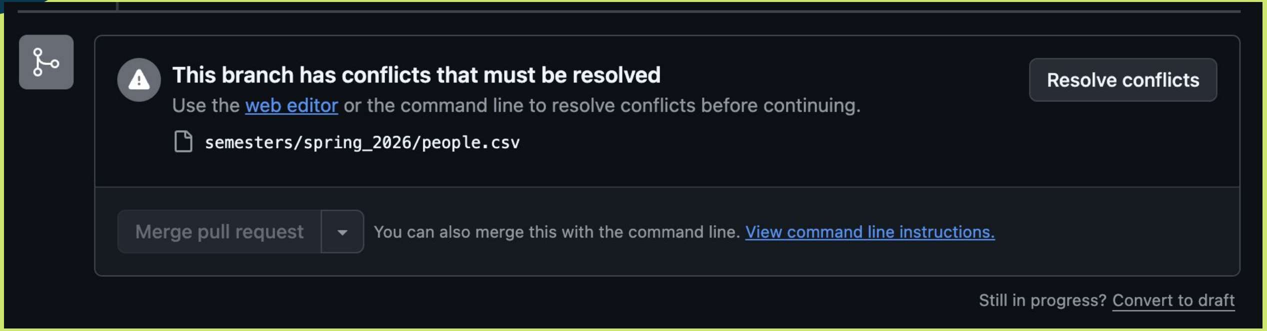


Lab1 chenyi
#132 opened 8 hours ago by JocelynJIANG-12

Add Indra Photo and Profile
#131 opened 10 hours ago by indrasuhyar

Branch
#130 opened 14 hours ago by sf3318

GitHub UI: Merge Conflicts

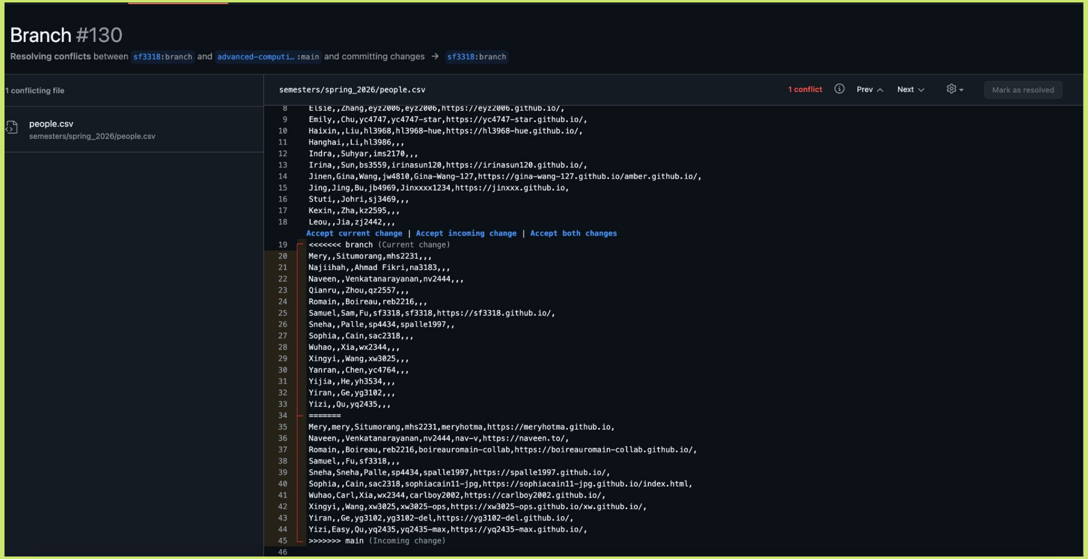

This branch has conflicts that must be resolved

Use the web editor or the command line to resolve conflicts before continuing.

semesters/spring_2026/people.csv

You can also merge this with the command line. View command line instructions.

Still in progress? Convert to draft


Branch #130

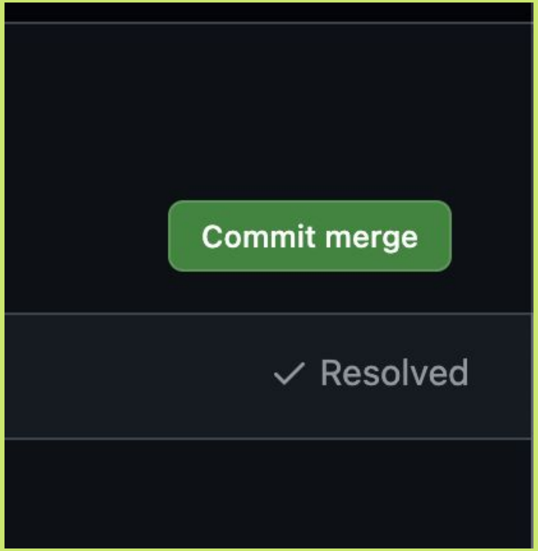
Resolving conflicts between sf3318:branch and advanced-computi… :main and committing changes → sf3318:branch

1 conflicting file

people.csv
semesters/spring_2026/people.csv

semesters/spring_2026/people.csv

```
Elsie,,Zhang,eyz2006,eyz2006,https://eyz2006.github.io/,
Emily,,Chu,yc4747,yc4747-star,https://yc4747-star.github.io/,
Haixin,,Liu,hl3968,hl3968-hue,https://hl3968-hue.github.io/,
Hanghai,,Li,hl3986,,,
Indra,,Suhyar,ims2170,,,
Irina,,Sun,bs3559,irinasun120,https://irinasun120.github.io/,
Jinen,Gina,Wang,jw4810,Gina-Wang-127,https://gina-wang-127.github.io/amber.github.io/,
Jing,Jing,Bu,jb4969,Jinxxxx1234,https://jinxxx.github.io,
Stuti,,Johri,sj3469,,,
Kexin,,Zha,kz2595,,,
Leou,,Jia,zj2442,,,
Accept current change | Accept incoming change | Accept both changes
<<<<<<< branch (Current change)
Mery,,Situmorang,mhs2231,,,
Najiihah,,Ahmad Fikri,na3183,,,
Naveen,,Venkatanarayanan,nv2444,,,
Qianru,,Zhou,qz2557,,,
Romain,,Boireau,reb2216,,,
Samuel,Sam,Fu,sf3318,sf3318,https://sf3318.github.io/,
Sneha,,Palle,sp4434,spalle1997,,
Sophia,,Cain,sac2318,,,
Wuhao,,Xia,wx2344,,,
Xingyi,,Wang,xw3025,,,
Yanran,,Chen,yc4764,,,
Yijia,,He,yh3534,,,
Yiran,,Ge,yg3102,,,
Yizi,,Qu,yq2435,,,
=======
Mery,mery,Situmorang,mhs2231,meryhotma,https://meryhotma.github.io,
Naveen,,Venkatanarayanan,nv2444,nav-v,https://naveen.to/,
Romain,,Boireau,reb2216,boireauromain-collab,https://boireauromain-collab.github.io/,
Samuel,,Fu,sf3318,,,
Sneha,Sneha,Palle,sp4434,spalle1997,https://spalle1997.github.io/,
Sophia,,Cain,sac2318,sophiacain11-jpg,https://sophiacain11-jpg.github.io/index.html,
Wuhao,Carl,Xia,wx2344,carlboy2002,https://carlboy2002.github.io/,
Xingyi,,Wang,xw3025,xw3025-ops,https://xw3025-ops.github.io/xw.github.io/,
Yiran,,Ge,yg3102,yg3102-del,https://yg3102-del.github.io/,
Yizi,Easy,Qu,yq2435,yq2435-max,https://yq2435-max.github.io/,
>>>>>>> main (Incoming change)
```



GIT Cheat Sheet:
https://git-scm.com/cheat-sheet

Visual Studio Code QuickStart:
https://code.visualstudio.com/docs/sourcecontrol/quickstart

Reminders / Best Practices

1. Don't learn every single command.

- There are thousands of commands. You'll never know all of them.
- And you are **not** expected to.

2. Know the functionality.

- You **should** know *what kinds of actions* there are, and how the system *works*.
- **Know the general workflow.**

3. Get comfortable looking things up.

- You will **have** to look stuff up at some point
    - Google, StackOverflow, and ChatGPT are your best friends.
- But know what is **SAFE** and **UNSAFE**.

4. Use what works best for you!

- There's no right or wrong answer!
- For specific things, one or other *might* be easier or faster.
- But in most cases, **it's up to you**!

What is a “bad state”?

A situation where your working directory, staging area, or repository history is in an **unexpected, broken, or undesirable configuration** that makes it **difficult to continue working**

These states can range from **minor user-induced confusion** to **severe repository corruption**

1. **Figure out what’s going on:** Pause and review the current state of your important files.

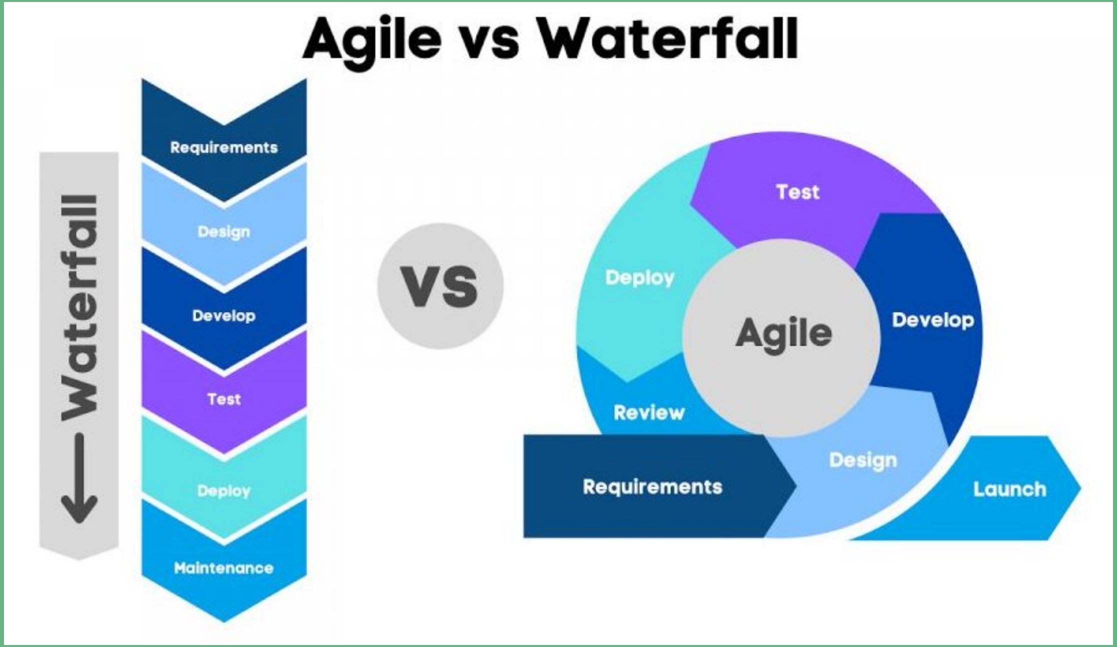
2. **Protect your work:** Make sure anything important is backed up before making changes.

3. **Fix what you can:** Carefully clean things up and keep only what you want.

4. **Last resort:** If it’s too messy, start over and rebuild from a clean state.

DESCENT INTO MADNESS
DOCUMENTED IN GIT COMMITS

Review: Project Management



Waterfall

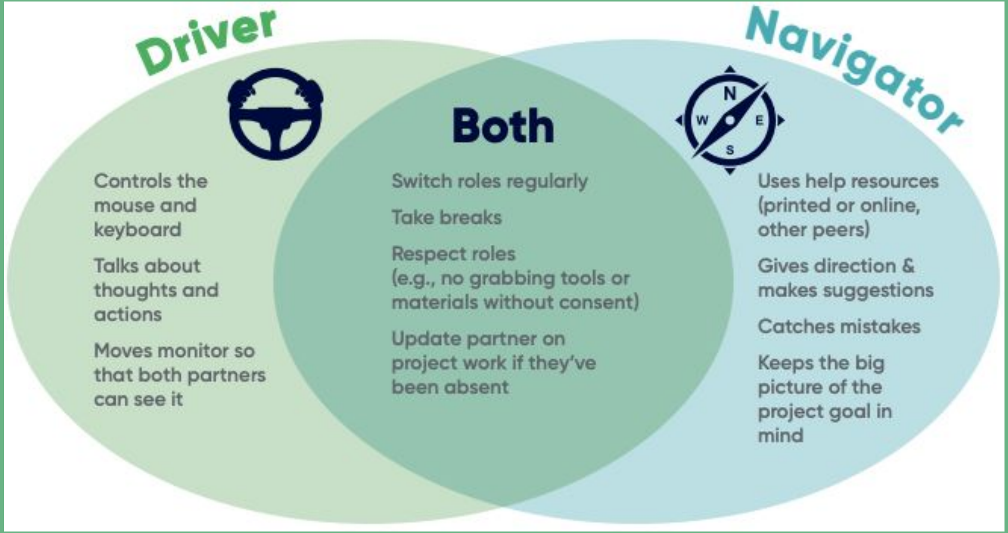
- **Linear**, step-by-step process with fixed phases
- Changes are **hard** and usually happen **late**

Agile

- **Iterative** development in **short** cycles (sprints)
- **Flexible** and **adapts quickly** to change

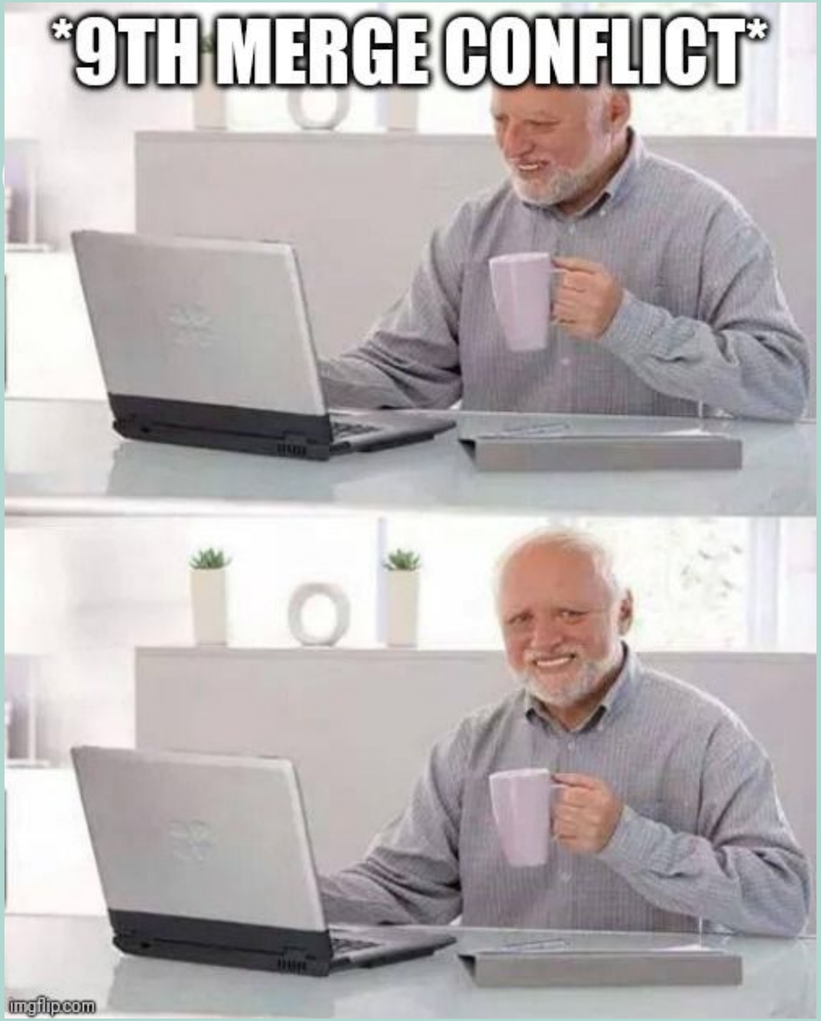
What is pair programming?

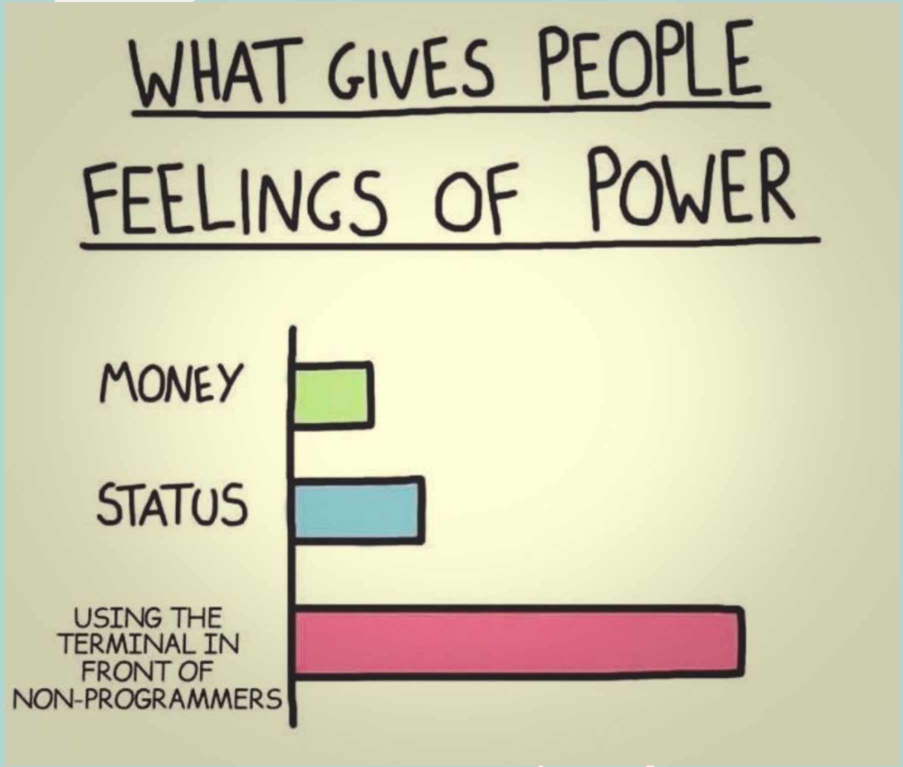
- Two people work on the same code together, at the same time
- One writes code (driver) while the other reviews and thinks ahead (navigator)
- Improves code quality, learning, and shared understanding



Driver
Both
Navigator
N
W
E
S
Controls the mouse and keyboard
Talks about thoughts and actions
Moves monitor so that both partners can see it
Switch roles regularly
Take breaks
Respect roles (e.g., no grabbing tools or materials without consent)
Update partner on project work if they've been absent
Uses help resources (printed or online, other peers)
Gives direction & makes suggestions
Catches mistakes
Keeps the big picture of the project goal in mind

WHAT IS YOUR AREA OF EXPERTISE?
WELL, I ACTUALLY UNDERSTAND HOW GIT WORKS
NO YOU DON'T MR SIMPSON NOBODY DOES


9TH MERGE CONFLICT
imgflip.com


WHAT GIVES PEOPLE FEELINGS OF POWER
MONEY
STATUS
USING THE TERMINAL IN FRONT OF NON-PROGRAMMERS

EXERCIS
E

Lfg