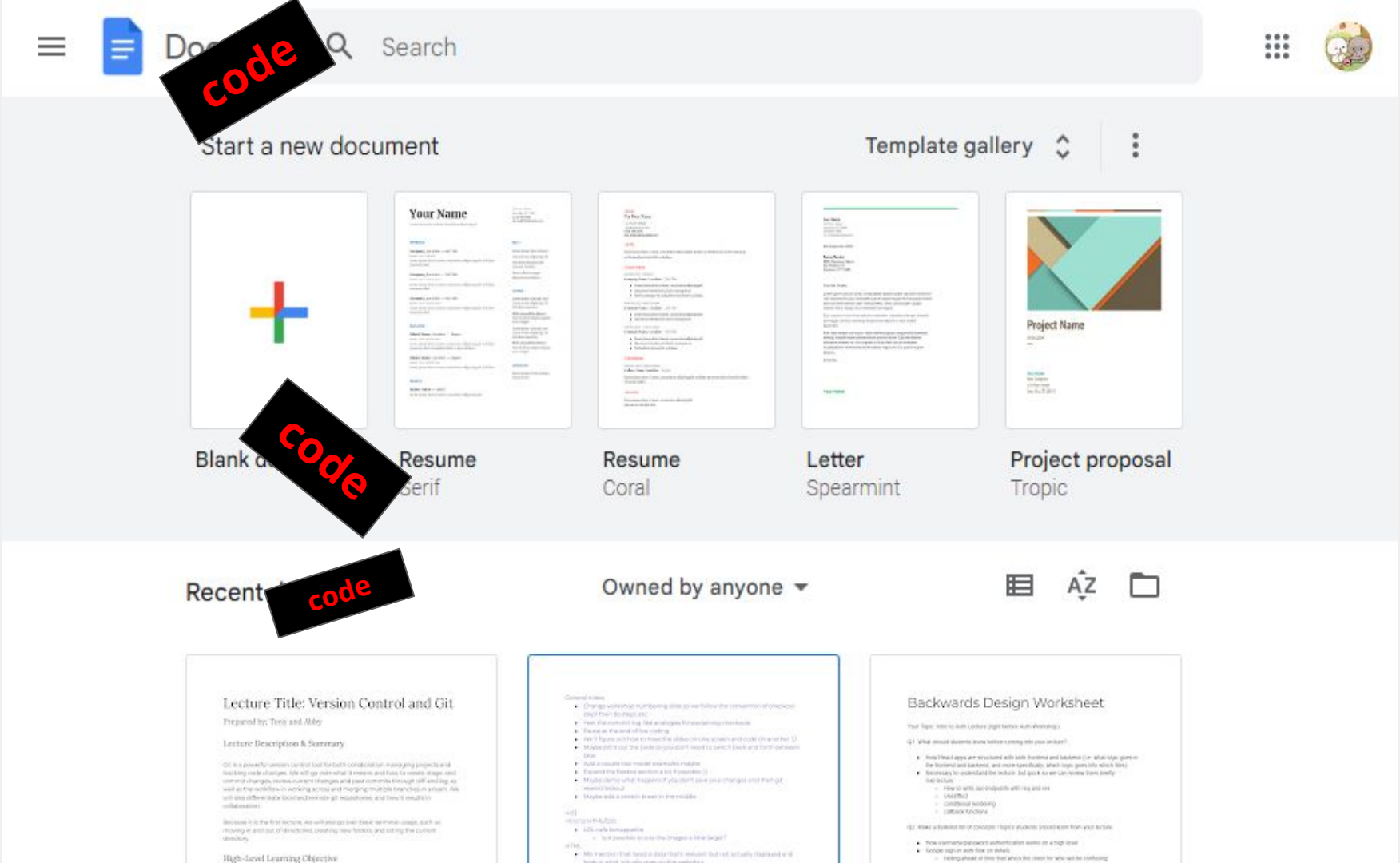
Git Basics

Tony Cui & Abby Chou

How do we collaborate on code?


code
Search
Start a new document
Template gallery
Your Name
Project Name
Blank
Resume
Serif
Resume
Coral
Letter
Spearmint
Project proposal
Tropic
code
code
Owned by anyone
Lecture Title: Version Control and Git
Backwards Design Worksheet


Consider
the
following

The issue with Code-Sharing

The issue with Code-Sharing

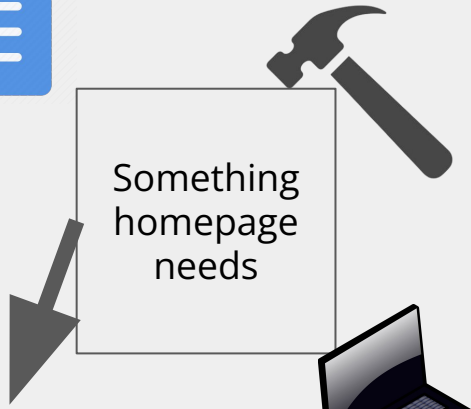
Home page

Something homepage needs

The issue with Code-Sharing

Home page

Something homepage needs



The issue with Code-Sharing

Bro what happened?

Home page

Something homepage needs

The issue with Code-Sharing

Bro what happened?

✨ upgrading ✨

Home page

Something homepage needs

The issue with Code-Sharing

Can u like not?

Home page

Something homepage needs

The issue with Code-Sharing

Can u like not?

Can u like ✨deal with it✨

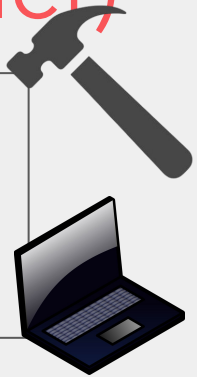
Home page

Something homepage needs

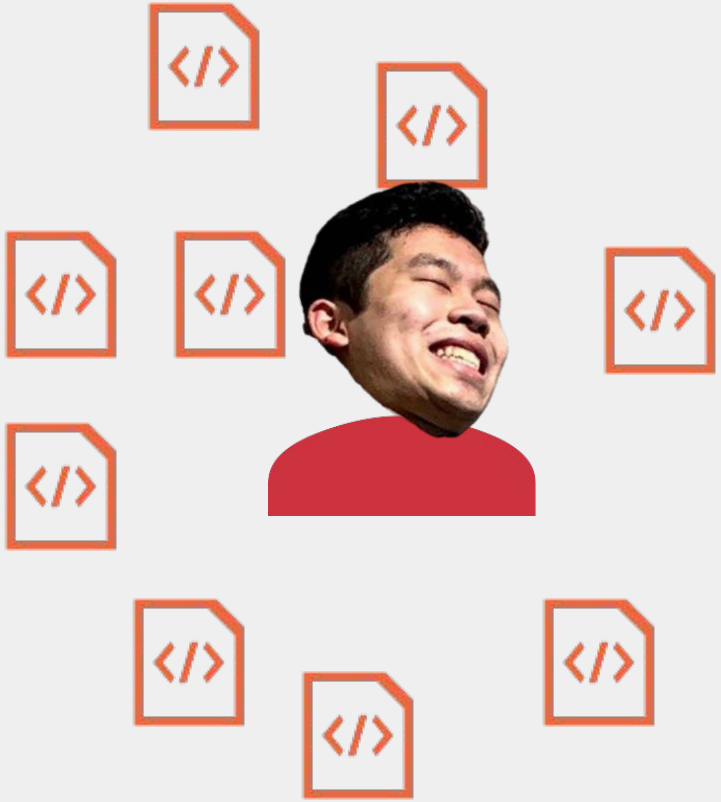
We need independent copies of the code! (so people can work without disrupting each other)

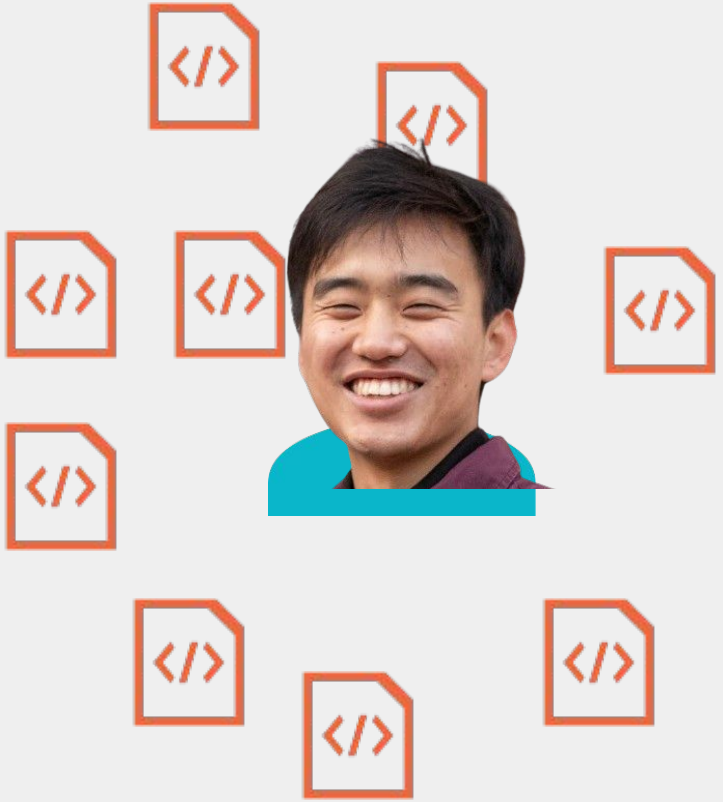
Who broke xyz i need it rn >:(

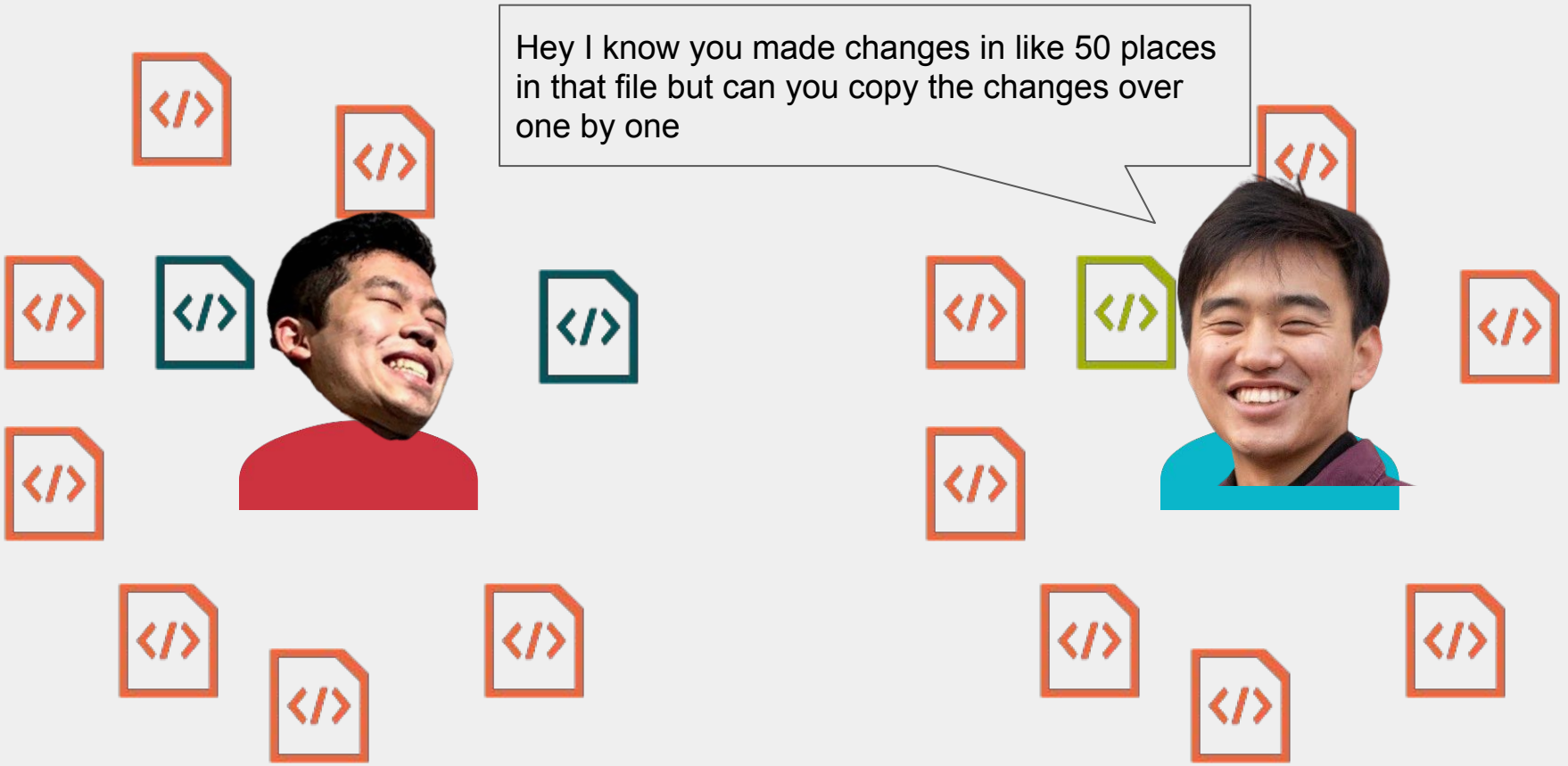
Something homepage needs

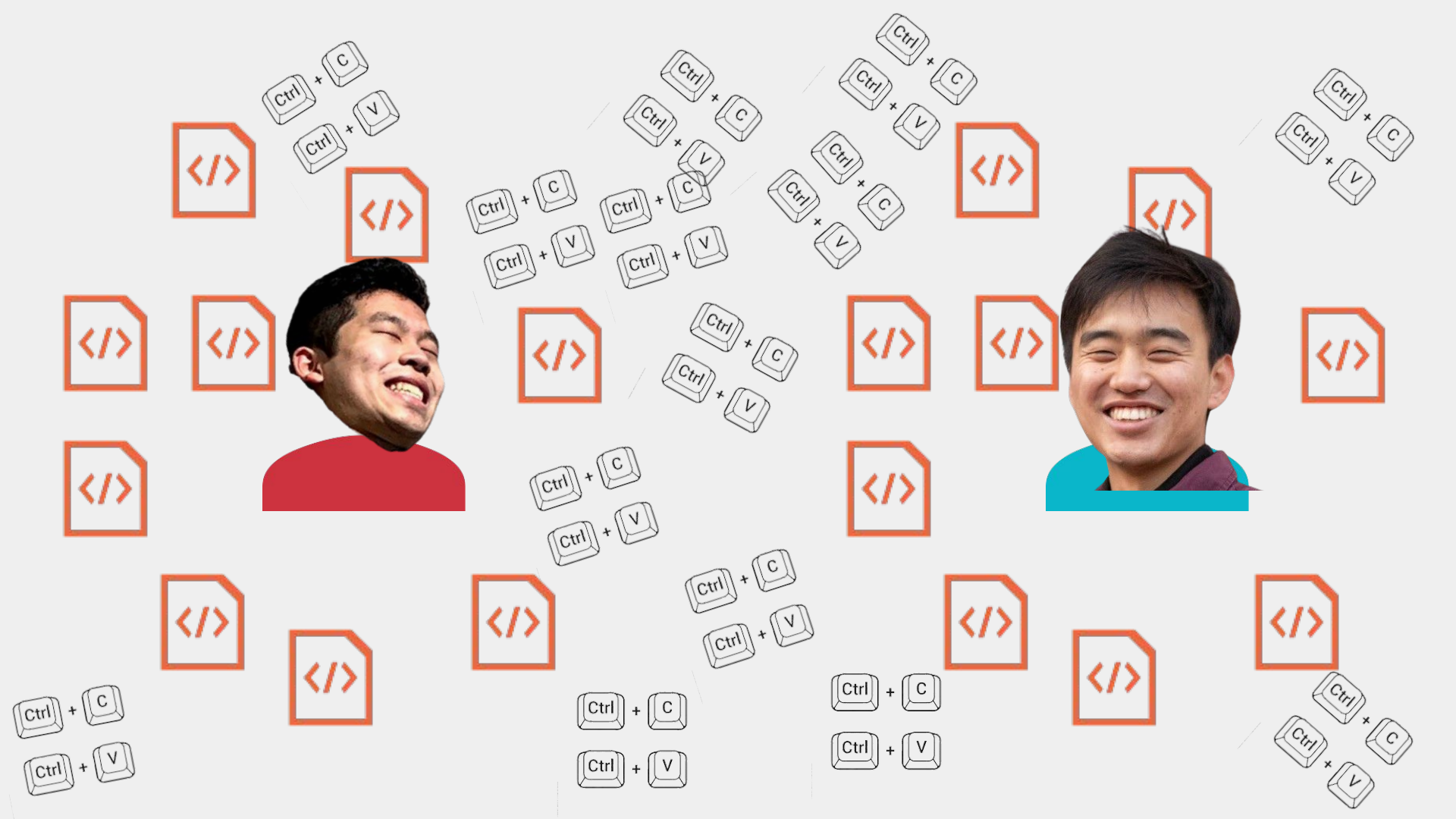


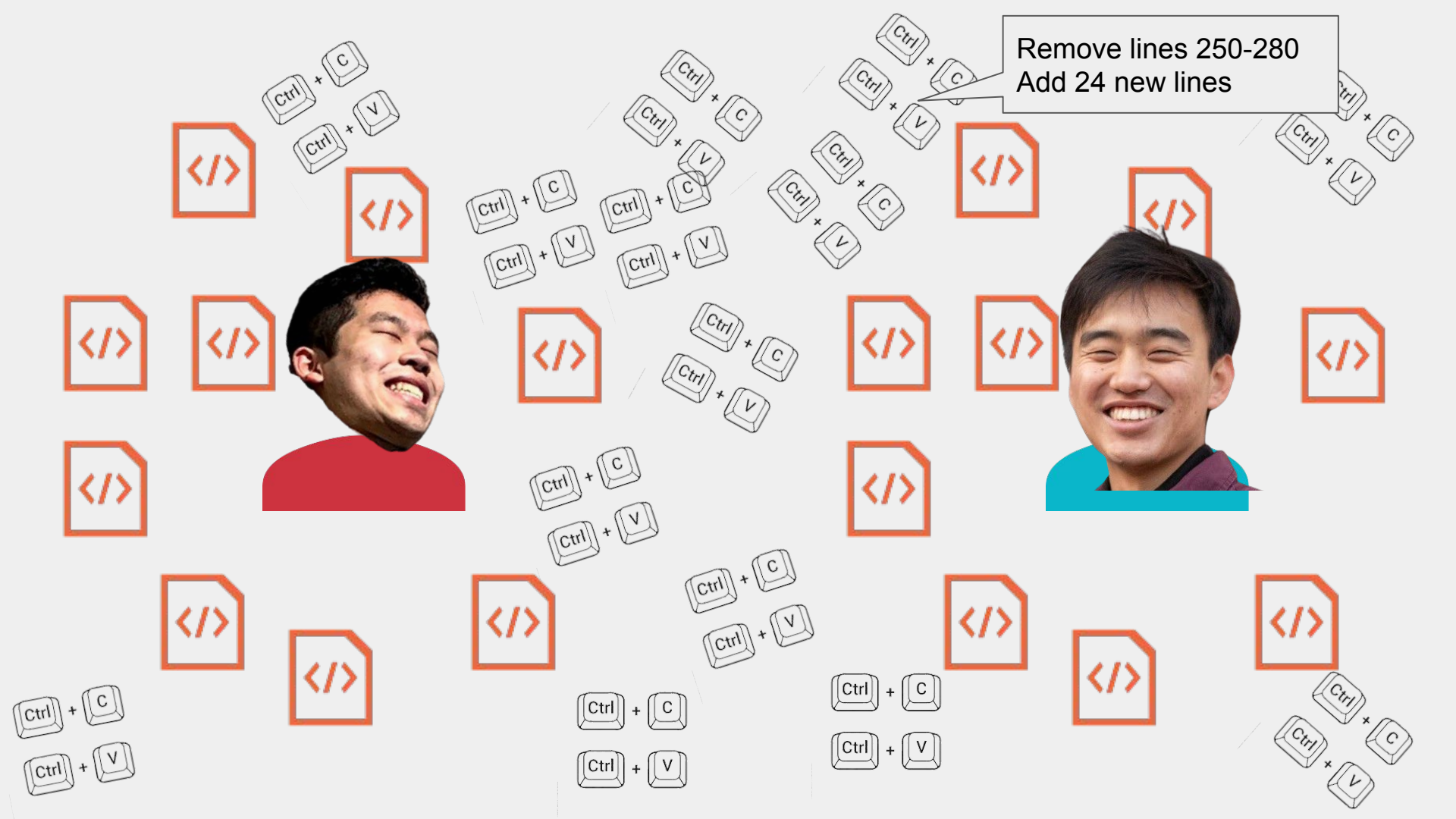
…but that creates so many problems 😖



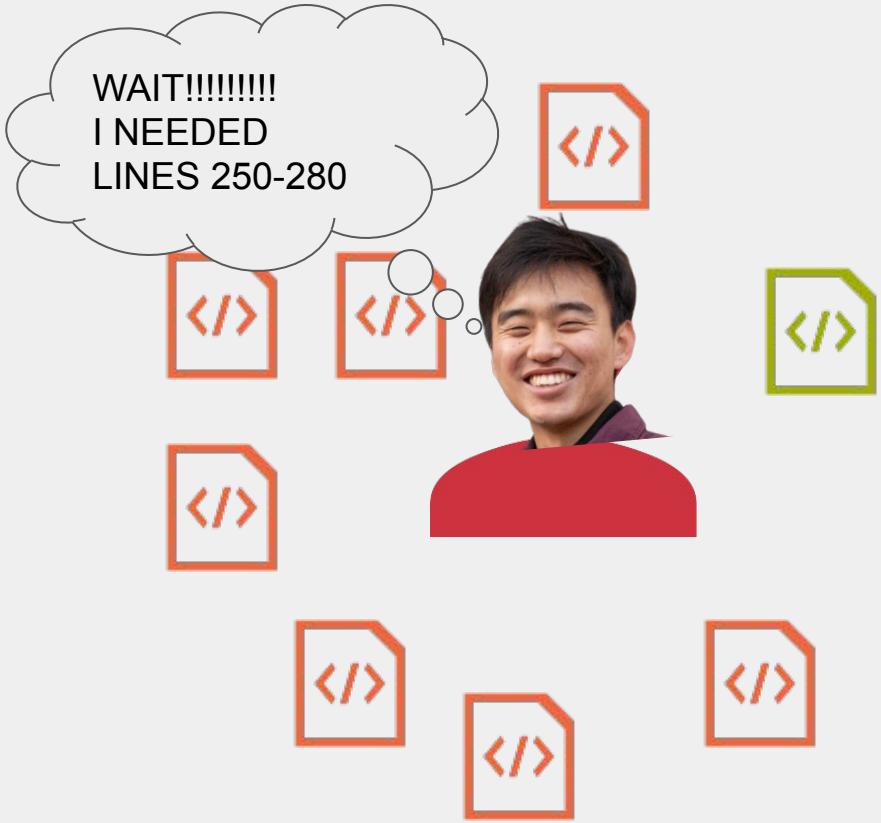


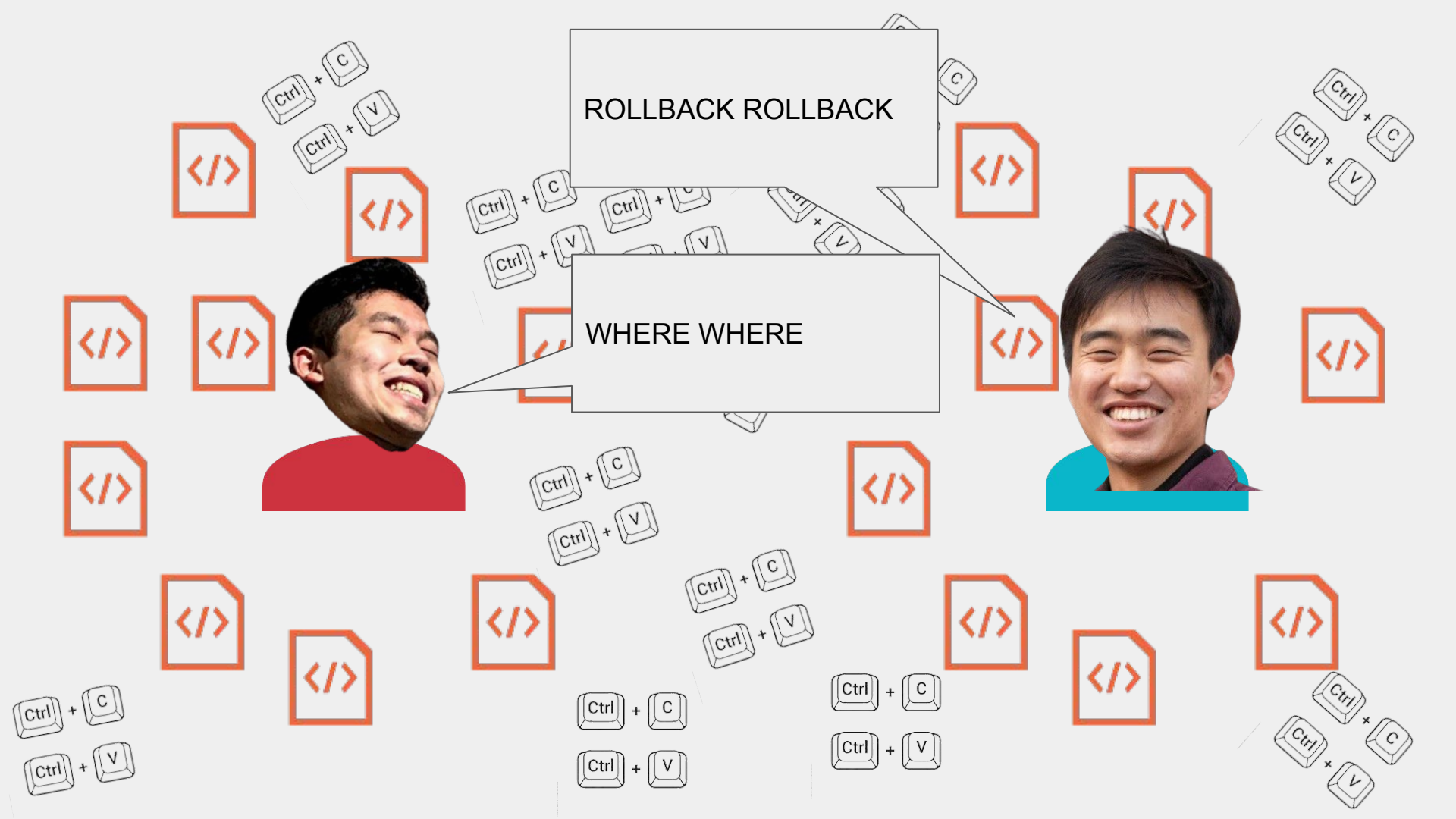

Hey I know you made changes in like 50 places in that file but can you copy the changes over one by one

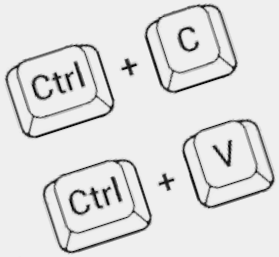

Ctrl + C
Ctrl + V

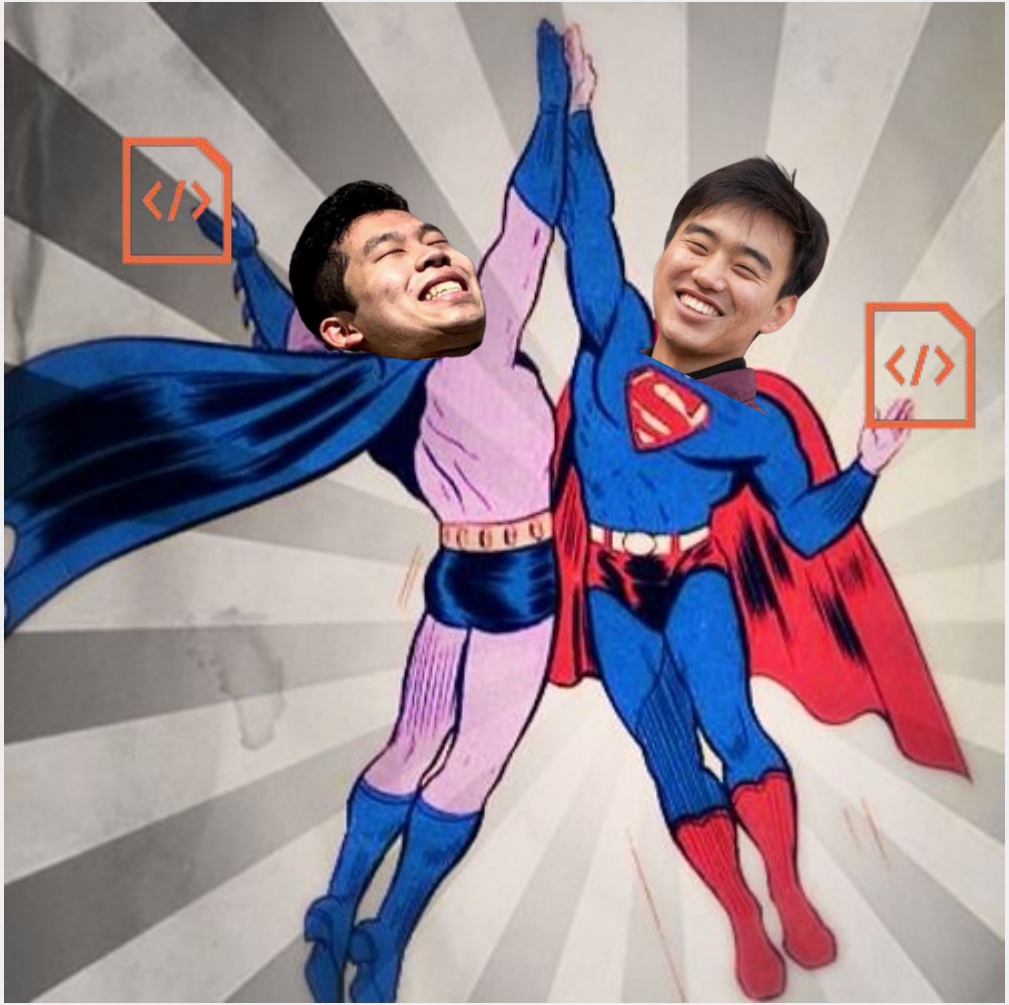

Remove lines 250-280
Add 24 new lines

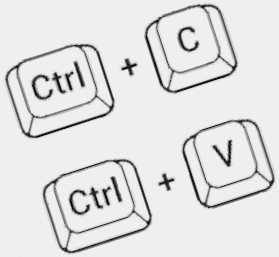
An hour later...

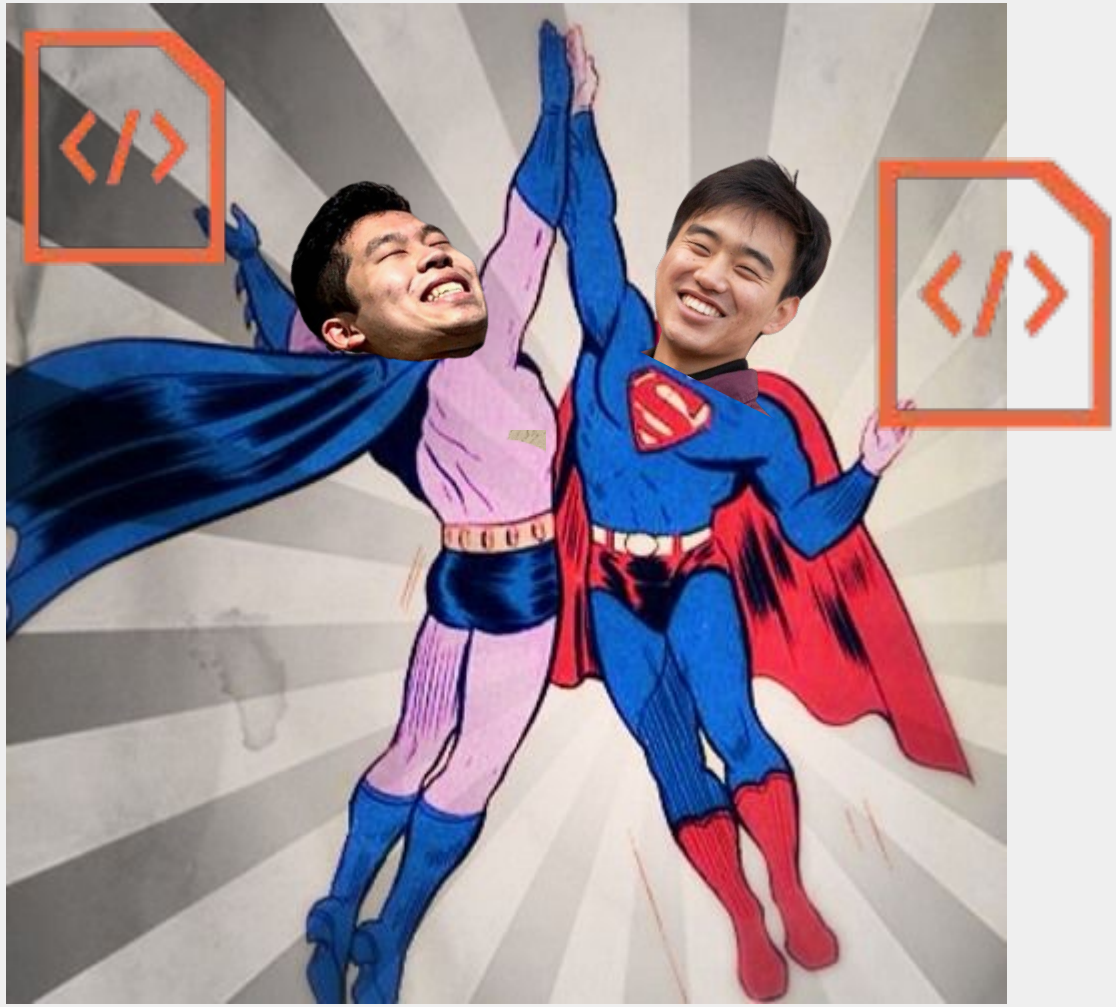

WAIT!!!!!!!!!
I NEEDED
LINES 250-280


ROLLBACK ROLLBACK
WHERE WHERE


Ctrl + C
Ctrl + V




Ctrl + C
Ctrl + V



????

The Problems

- Need independent local copies of the codebase
- Need to be able to merge different people’s changes together
- Need to keep track of versions
- Need to know which version is the most up-to-date

How do we solve these issues? Introducing...

What does Git do?

Git tracks *changes.*

(How? ✨math✨)

mydoc.txt


I’m writing a story.
It’s really good!
And about cats!
I’m writing a story.
It’s really good!
And about cats!
Like corgis :D
version1
version2

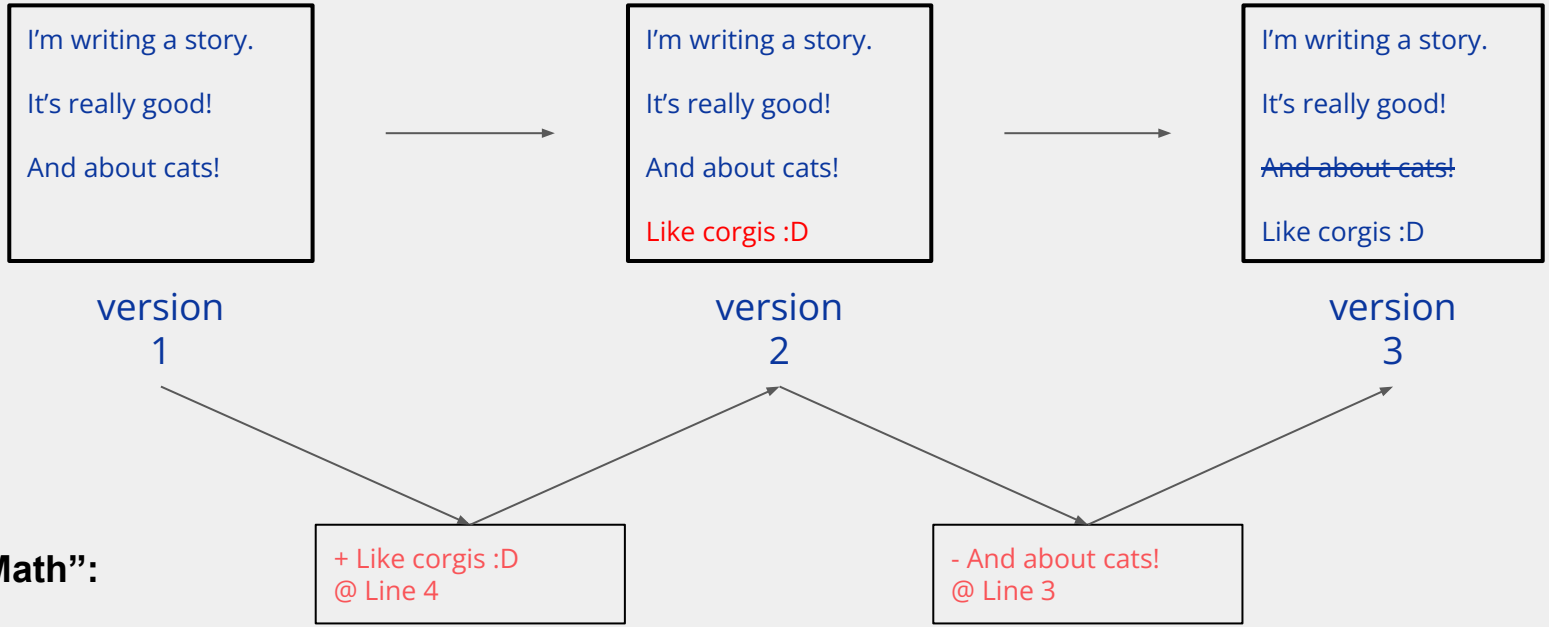
mydoc.txt

version 1

version 2

version 3

mydoc.txt



The “Math”:

example

Getting set up with git :D

Demonstration

Try the following:

1. Make a new directory
2. Cd into that directory
3. Turn your directory into a git repo!

Reference

cd [directory] change directory

ls list the contents of current directory

mkdir [folder_name] make directory

git init turns current directory into git repo!

Special Characters:

. current directory

.. parent directory

Your turn!

Try the following:

1. Make a new directory
2. Cd into that directory
3. Turn your directory into a git repo!

Reference

cd [directory] change directory

ls list the contents of current directory

mkdir [folder_name] make directory

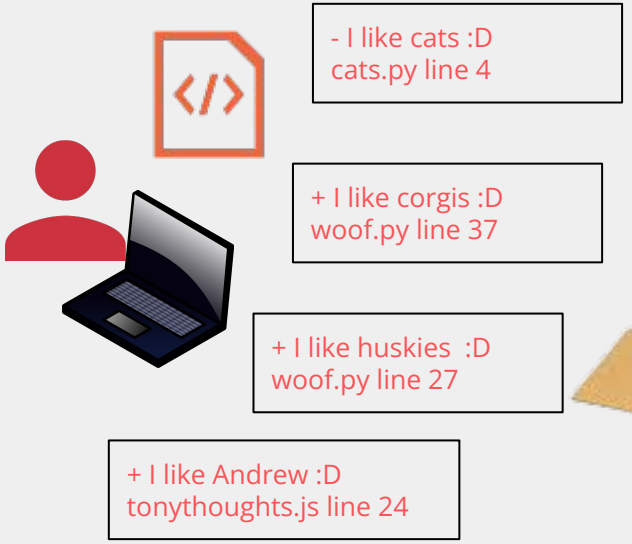
git init turns current directory into git repo!

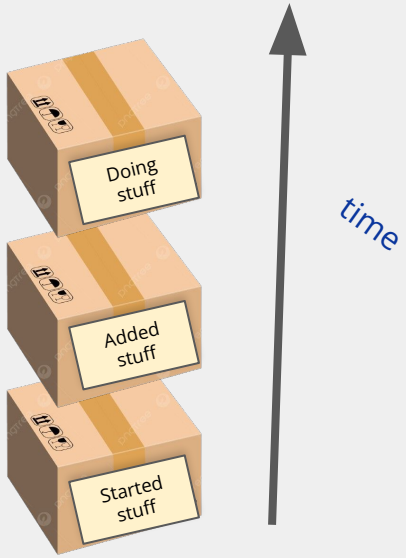
Special Characters:

. current directory

.. parent directory

Staging & Committing



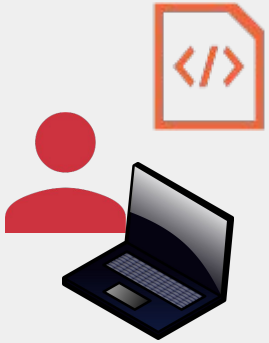


Working copy

Staging Area

Commit History

Staging & Committing



Working copy

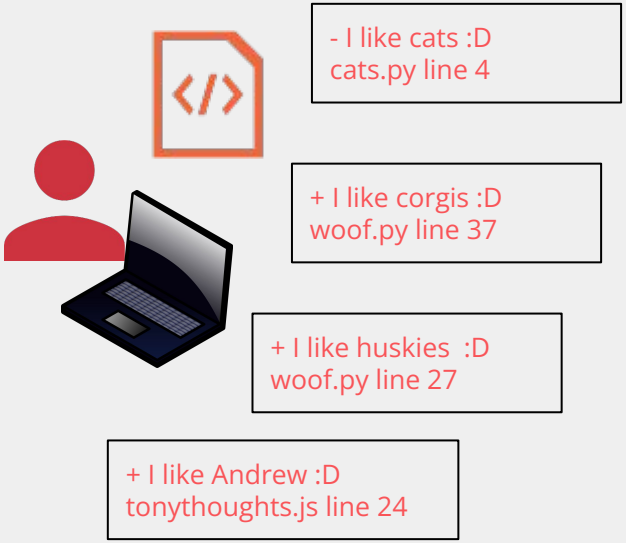
Staging & Committing

- I like cats :D
cats.py line 4

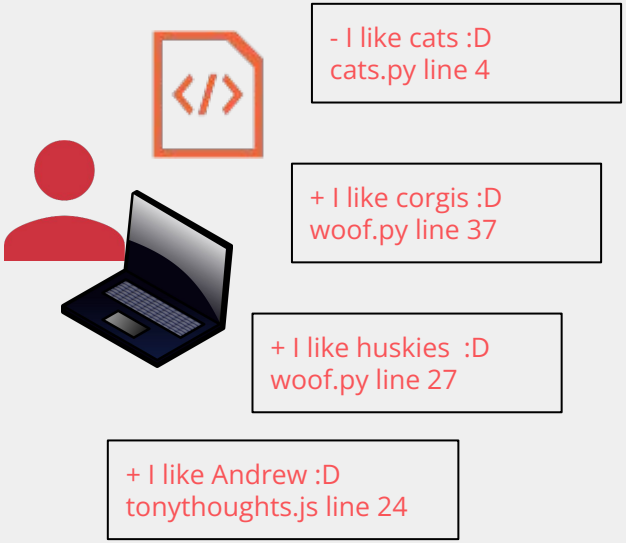
+ I like corgis :D
woof.py line 37

Staging & Committing

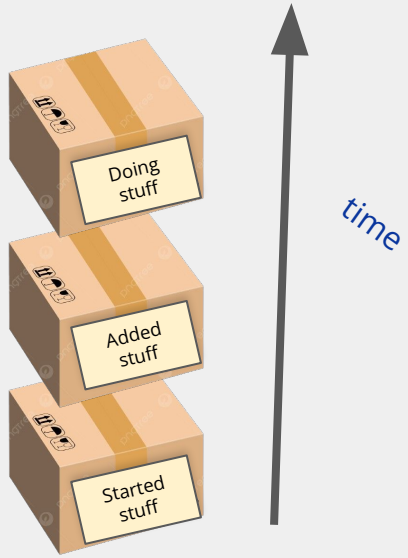
Staging & Committing



Staging & Committing

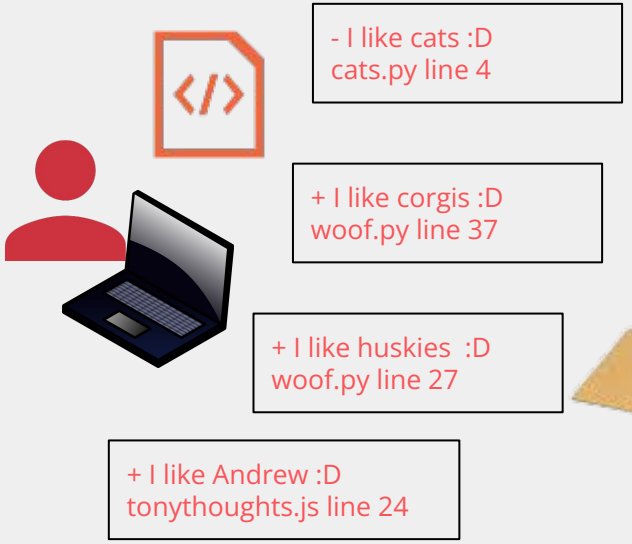


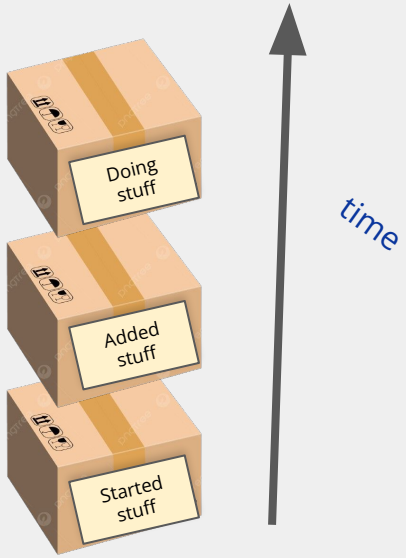
Working copy



Commit History

Staging & Committing



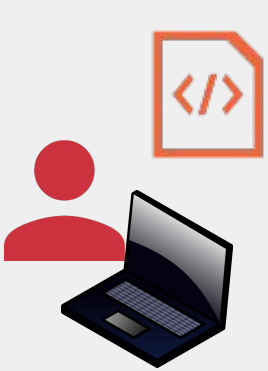


Working copy

Staging Area

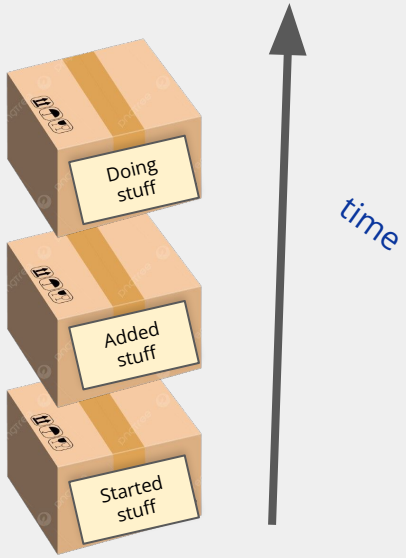
Commit History

Staging & Committing



- I like cats :D
cats.py line 4

+ I like Andrew :D
tonythoughts.js line 24

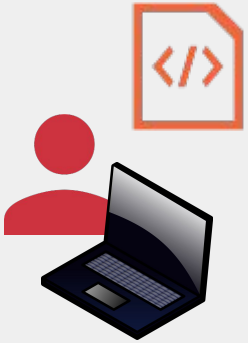


Working copy

Staging Area

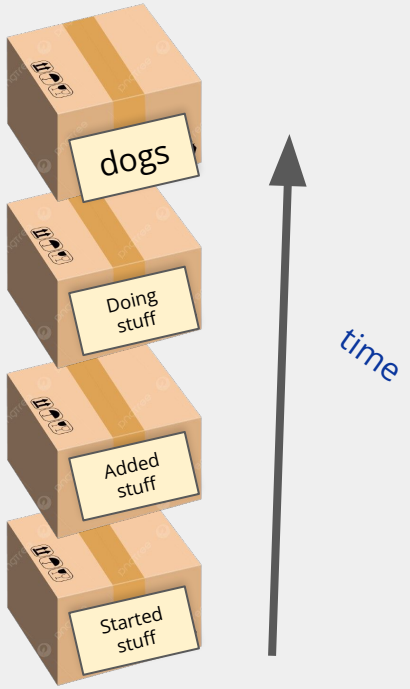
Commit History

Staging & Committing



- I like cats :D
cats.py line 4

+ I like Andrew :D
tonythoughts.js line 24

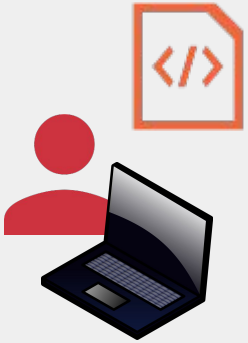


Working copy

Staging Area

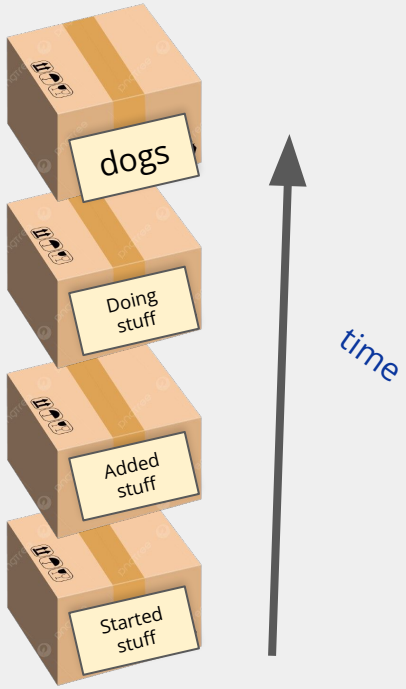
Commit History

Staging & Committing



- I like cats :D
cats.py line 4

+ I like Andrew :D
tonythoughts.js line 24



Working copy

Staging Area

Commit History

Demo Staging & Commits

Demonstration

1. Make some changes
    a. Git status
2. Stage some changes
    a. Git status
3. Make a commit
    a. Git status
4. Git log
    a. Git status

git diff

git status

git add

git commit -m

Your turn!

In your new repo...

1. Make some code changes
2. Stage changes with git add
3. Commit the changes
4. Check your commit log
5. (Optional) Repeat! (Try deleting stuff too)
    a. After you make changes (before you stage them), git diff to see your changes!

Commands Ref

git add stage changes for commit

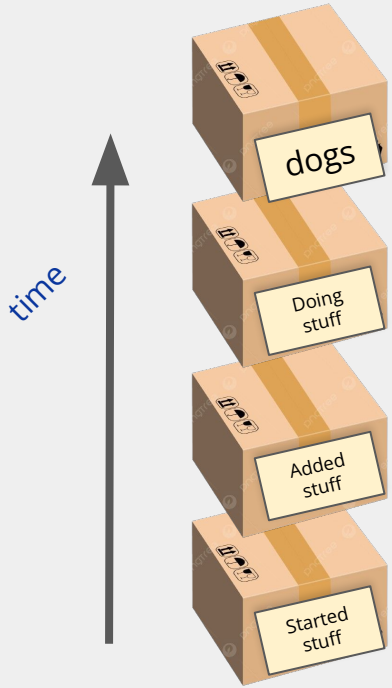
git commit -m “[msg]”

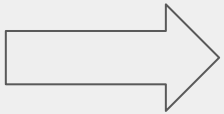
git status <- use this often!

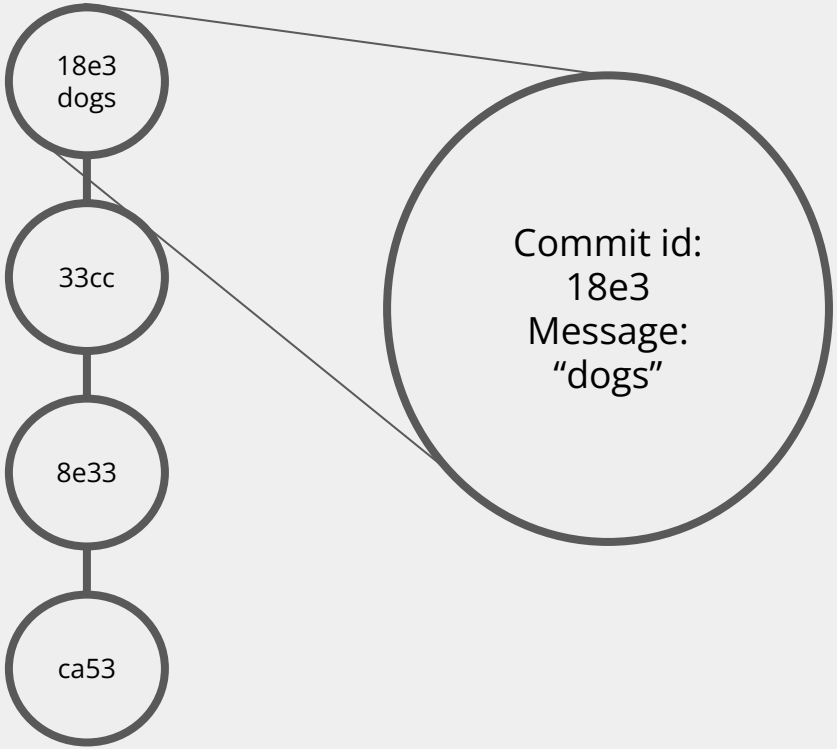
git log show commit history

git diff show the diff (changes) between working copy and staged/committed copy

Object Graph



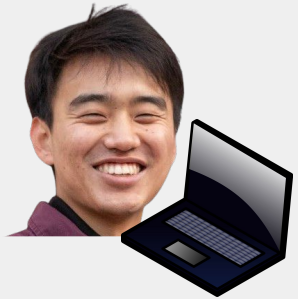





Consider
the
following

Uh oh, my boss is madge

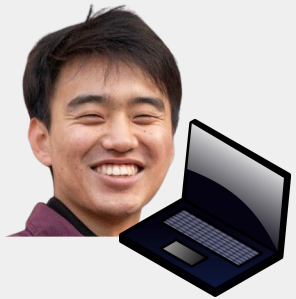
tony: “vibing working on my new and improved home page ”



Uh oh, my boss is madge

tony: “vibing working on my new and improved home page ”

andrew: “can u help me fix this bug on the home page uwu”

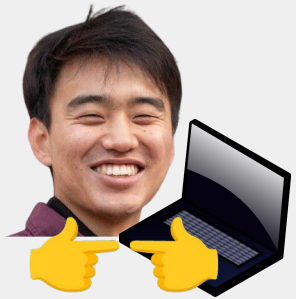


Uh oh, my boss is madge

tony: “vibing working on my new and improved home page ”

andrew: “can u help me fix this bug on the home page uwu”

tony: “anything for u bb 😍”



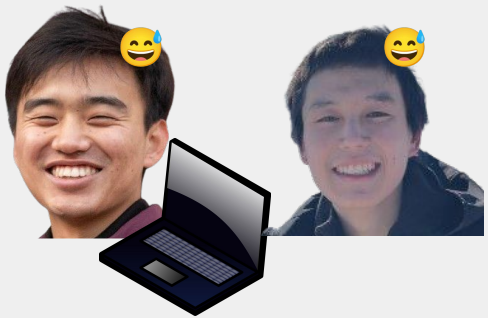
Uh oh, my boss is madge

tony: “vibing working on my new and improved home page ”

andrew: “can u help me fix this bug on the home page”

tony: “anything for u bb 😍”

boss abby (who got dumped recently 😒) : “tony we need a weblab-tinder feature TOMORROW and u hav to do it”



Uh oh, my boss is madge

tony: “vibing working on my new and improved home page ”

andrew: “can u help me fix this bug on the home page”

tony: “anything for u bb 😍”

boss abby (who got dumped recently 😒) : “tony we need a weblab-tinder feature TOMORROW and u hav to do it”

Uh oh, my boss is madge

tony: “vibing working on my new and improved home page ”

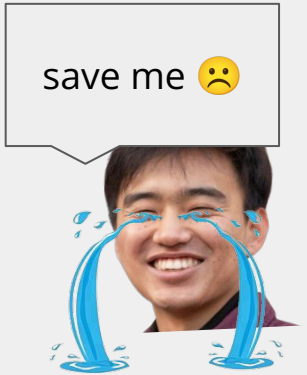
andrew: “can u help me fix this bug on the home page”

tony: “anything for u bb 😍”

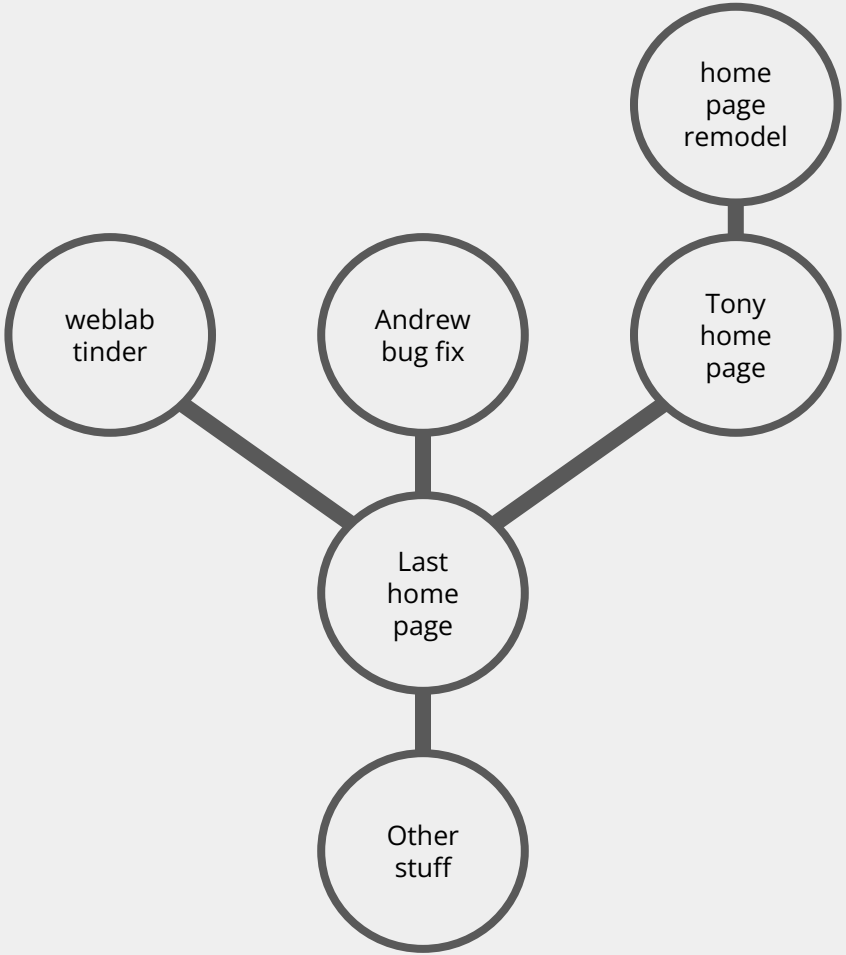
boss abby (who got dumped recently 😒) : “tony we need a weblab-tinder feature TOMORROW and u hav to do it”

tony: “bruh”

How do we work on multiple features at once?



Branching


home page remodel
weblab tinder
Andrew bug fix
Tony home page
Last home page
Other stuff

Demo Branches

Demonstration

1. Creating a new branch
2. Switching between branches

Your turn!

In your new directory...

1. Create a new branch
2. Add a couple of commits to your branch, check the log
3. Checkout back to main, check the log
4. Add a couple of commits to main, check the log
5. Try and draw and/or explain to your partner what you think the object graph looks like!

Handy commands

git branch see branches

git checkout [branch-name]

switch to existing branch

git checkout -b [branch-name]

create and checkout new branch

Previous Commands

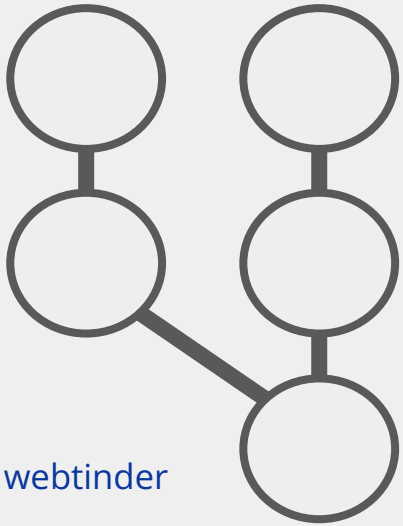
git status

git add

git commit -m

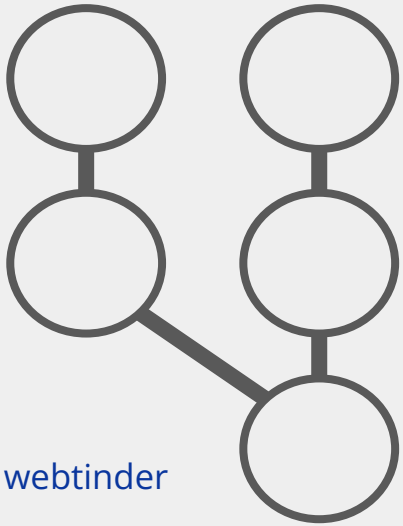
git log

What is happening in the object graph :0?



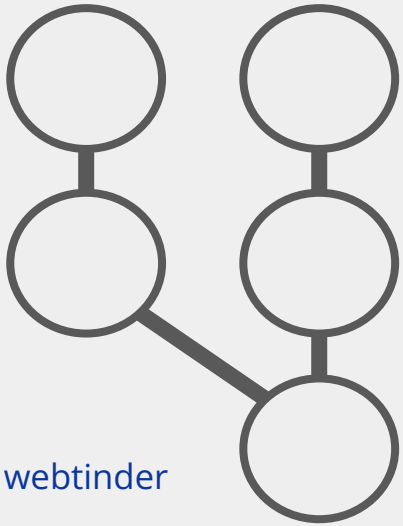
main

What is happening in the object graph :0?



main

How do we put them back together?



main

example

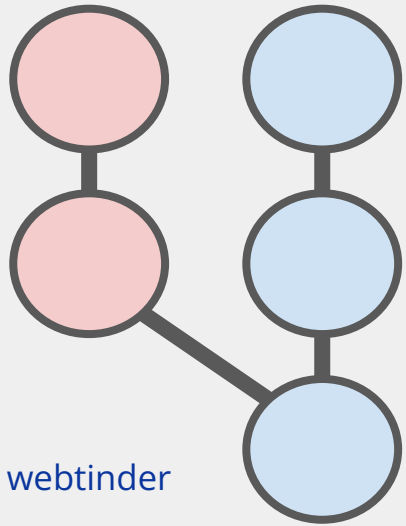
merging

Follow Along

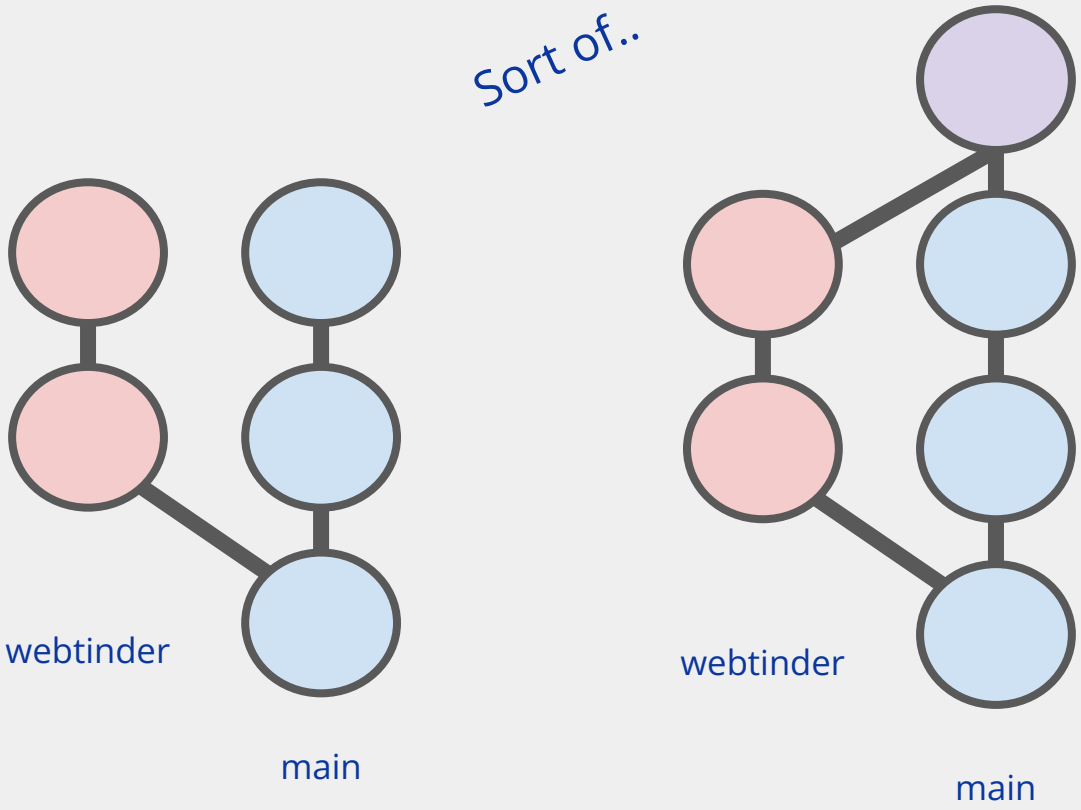
Follow along!

1. Checkout into main branch
2. Git merge [new branch name]

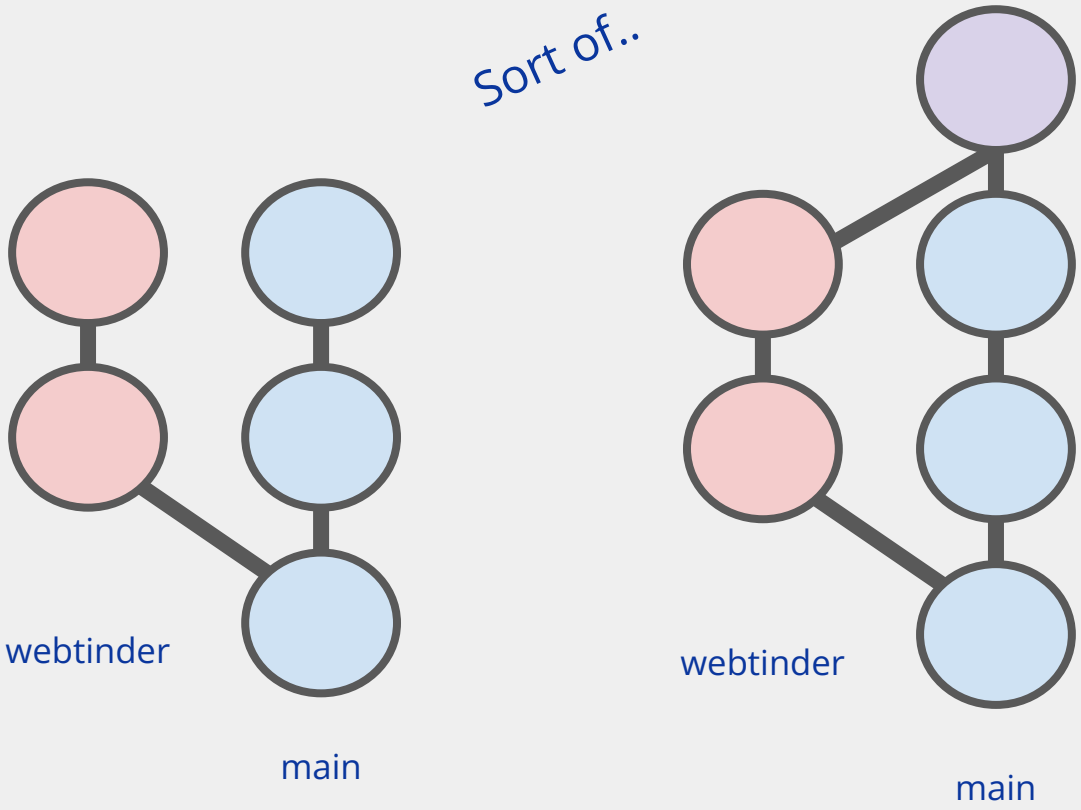
How do we put them back together?



How do we put them back together?

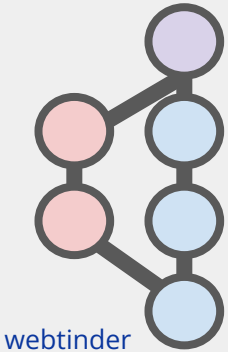


How do we put them back together?



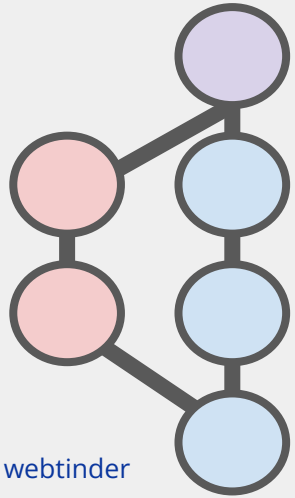
Preview

🐳, how does collaboration work then?



webtinder

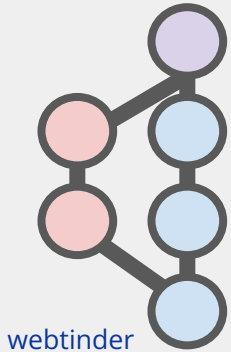
main



webtinder

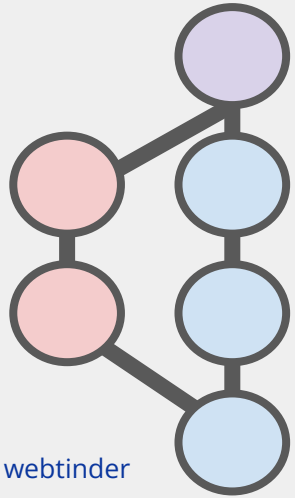
main

Remote Server (github)



webtinder

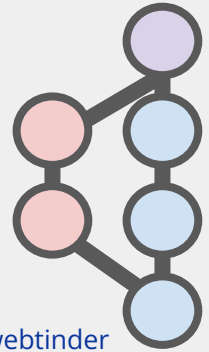
main



webtinder

main

git clone

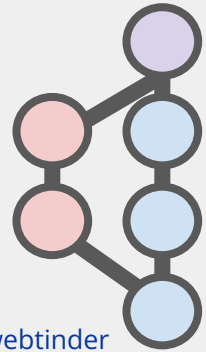


webtinder

main

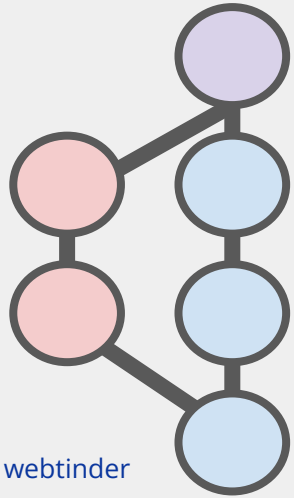
Remote Server (github)

git commit



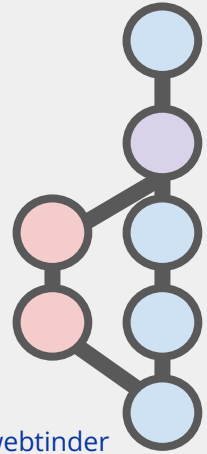
webtinder

main



webtinder

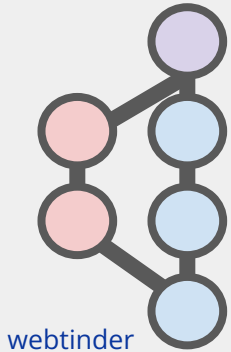
main

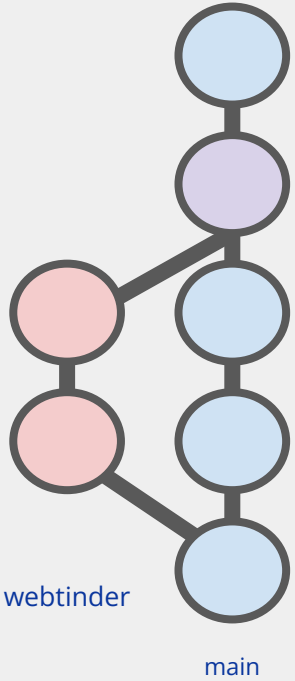


webtinder

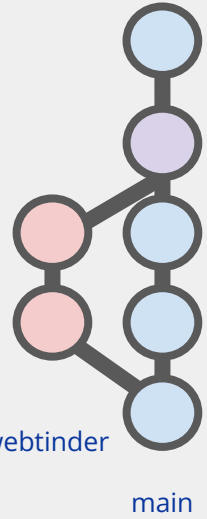
main

Remote Server (github)



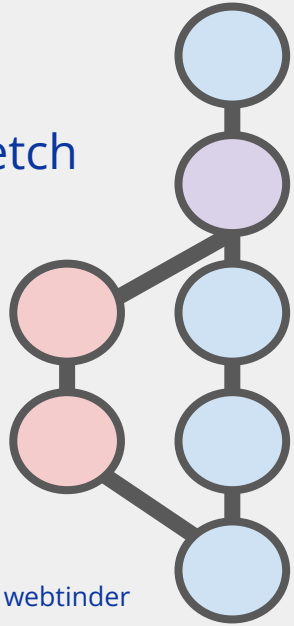


git push



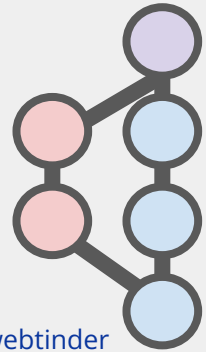
Remote Server (github)

Git fetch



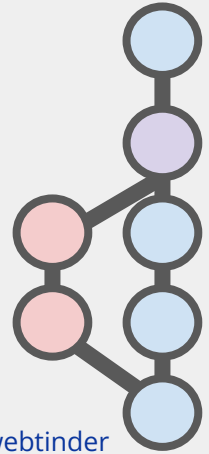
webtinder

main



webtinder

main

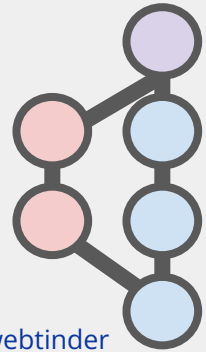


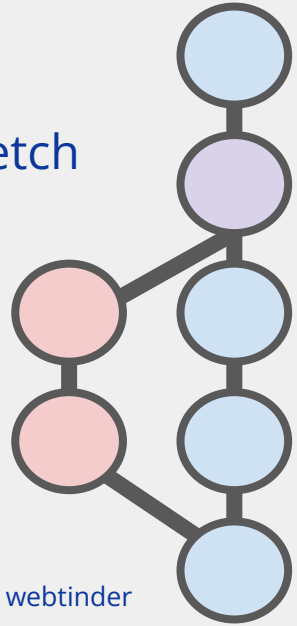
webtinder

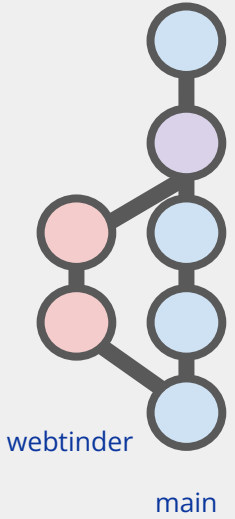
main

Remote Server (github)

Git fetch

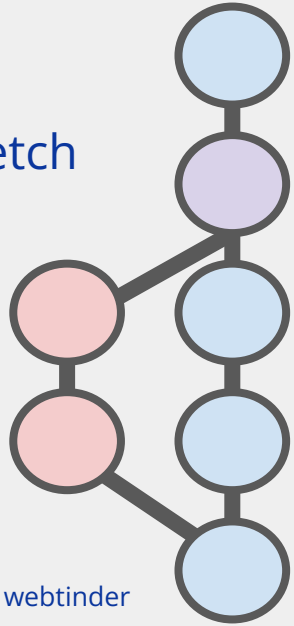






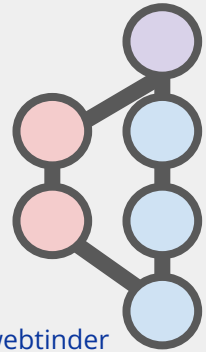
Remote Server (github)

Git fetch



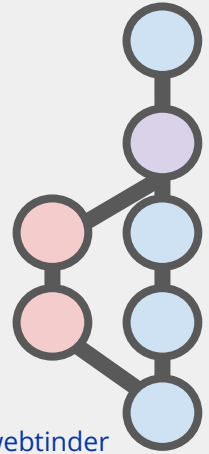
webtinder

main



webtinder

main

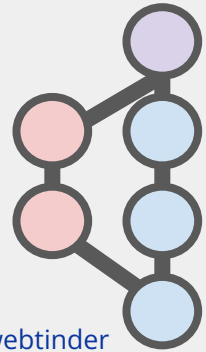


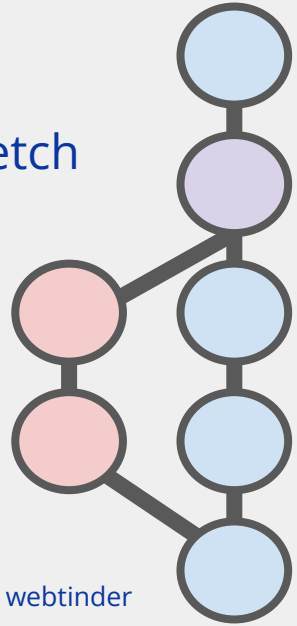
webtinder

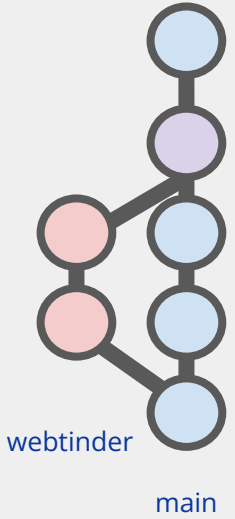
main

Remote Server (github)

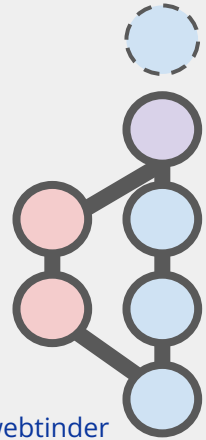
Git fetch





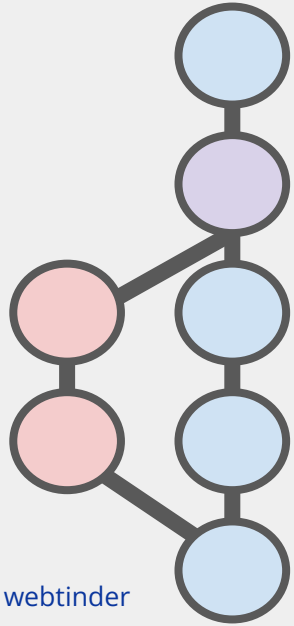


Remote Server (github)



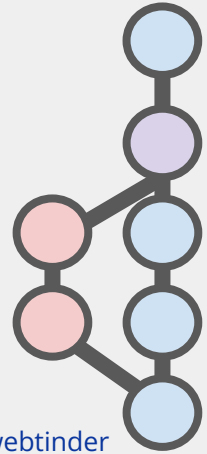
webtinder

main



webtinder

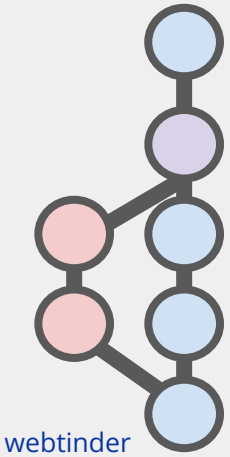
main



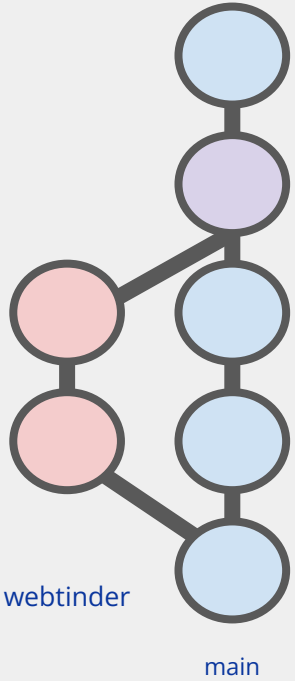
webtinder

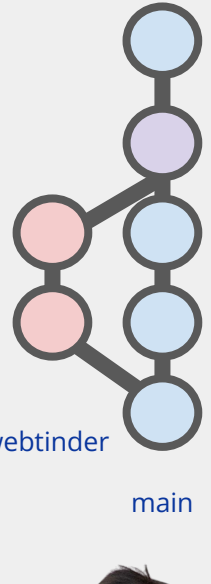
main

Remote Server (github)



git pull



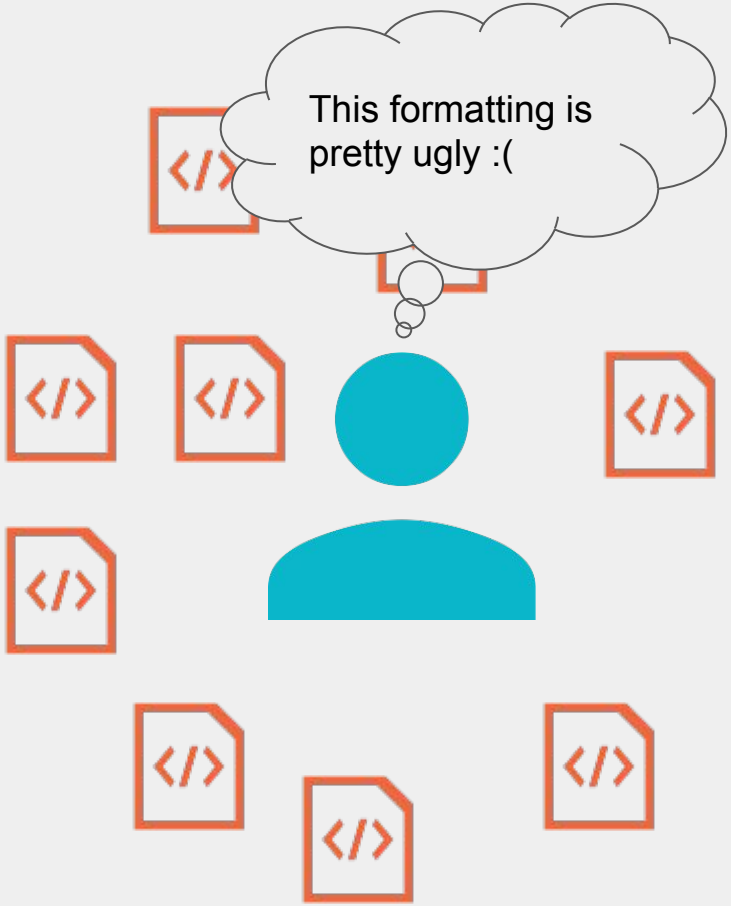


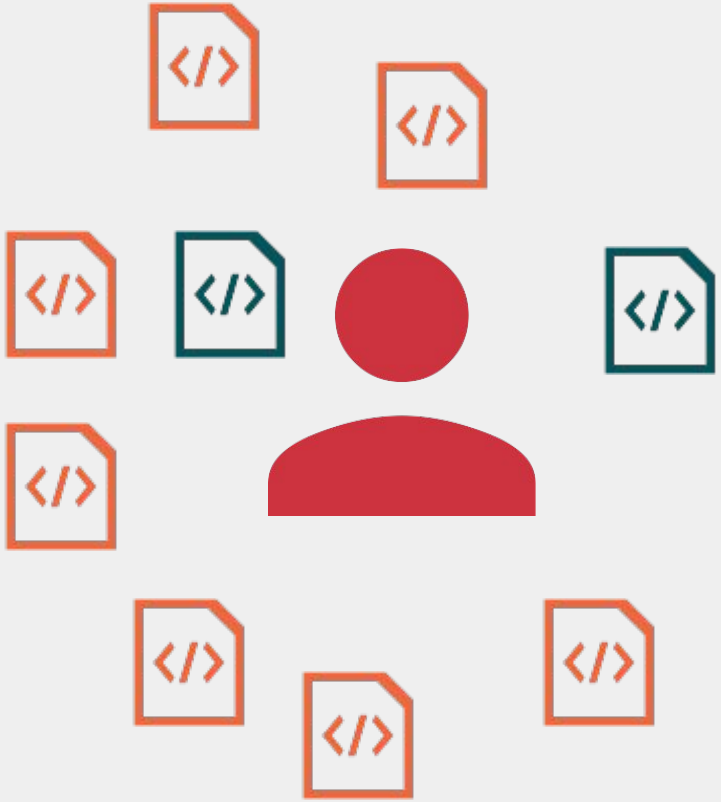
Remote Server (github)

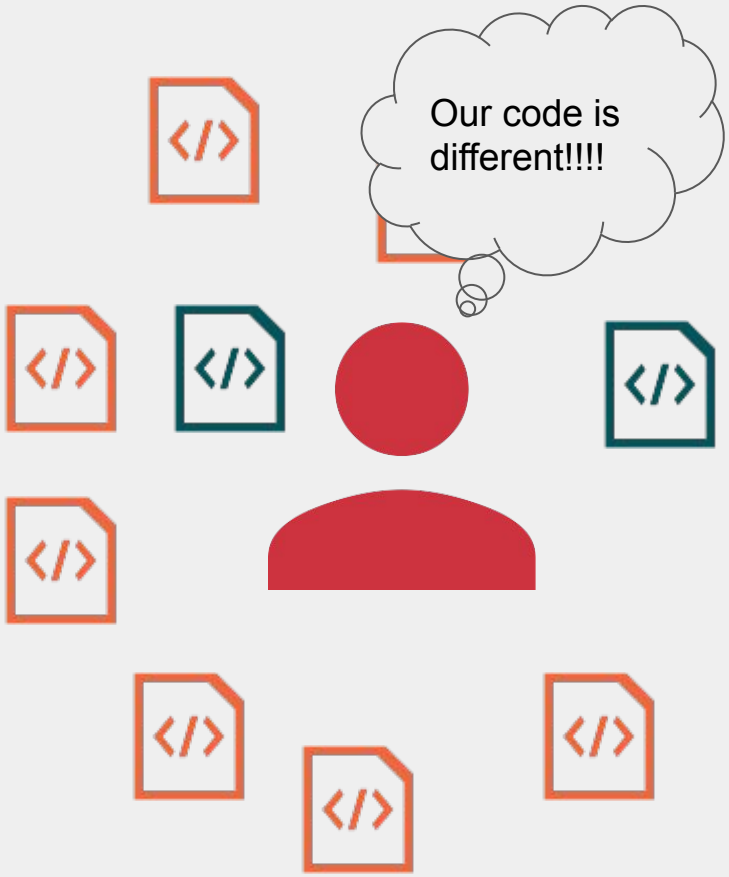
Stay tuned!

git reset –hard


There’s a bug!


This formatting is
pretty ugly :(




Our code is different!!!!