

Git For Collaboration

Tony Cui & Abby Chou

Agenda

1. Recap: Git and Version Control
 - a. Staging, Commits, Commit log
 - b. Branches
2. Merging and Merge Conflicts
3. Remote Git

Recap: Making Commits

Sample Commit



SingleComment.js

```
47
-const SingleComment = () => {
+const SingleComment = (props) => {
  return (
    <div className="Card-commentBody">
-    { /* TODO (step7): use JSX and props to render comment creator and content */ }
-    </div>
+    <div className="Card-commentBody">
+    <span className="u-bold">{props.creator_name}</span>
+    <span>{" | " + props.content}</span>
+    </div>
+    </div>

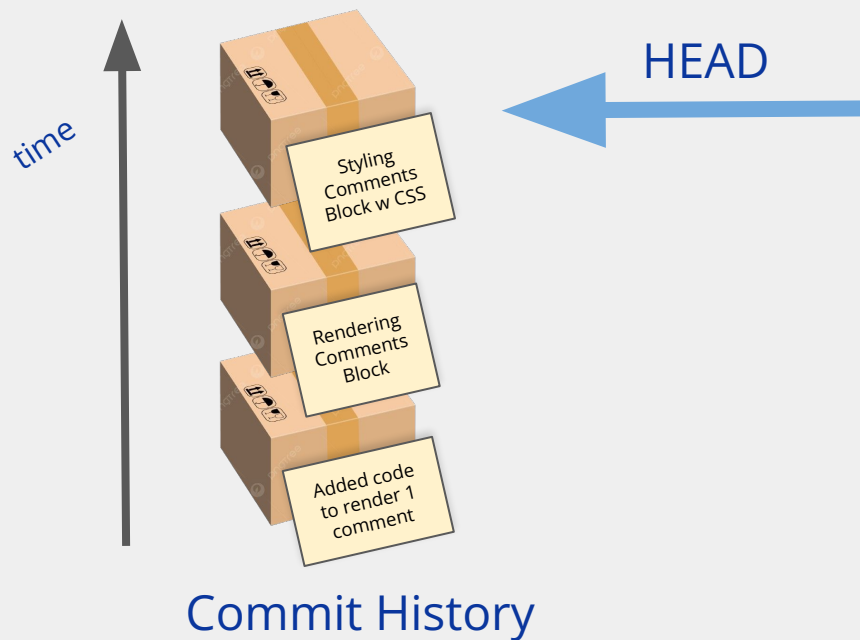
```

Card.js

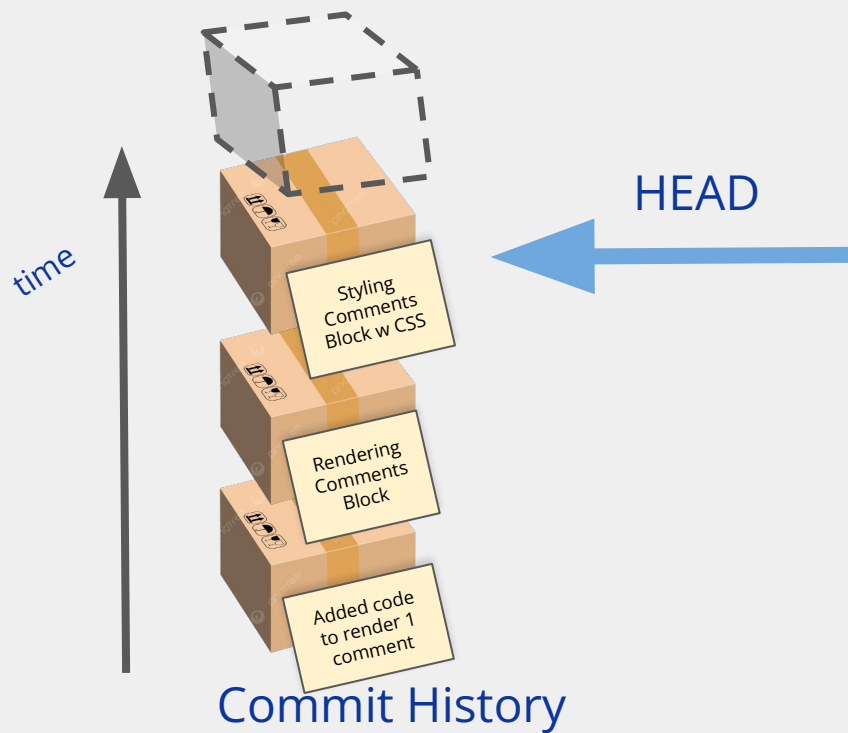
```
48 b/client/src/components/modules/Card.js
@@ -1,7 +1,7 @@
import React, { useState, useEffect } from "react";
import { get } from "../../utilities";
import SingleStory from "../SingleStory.js";
-// TODO (step7): import SingleComment
+import SingleComment from "../SingleComment.js";

```

Staging & Committing

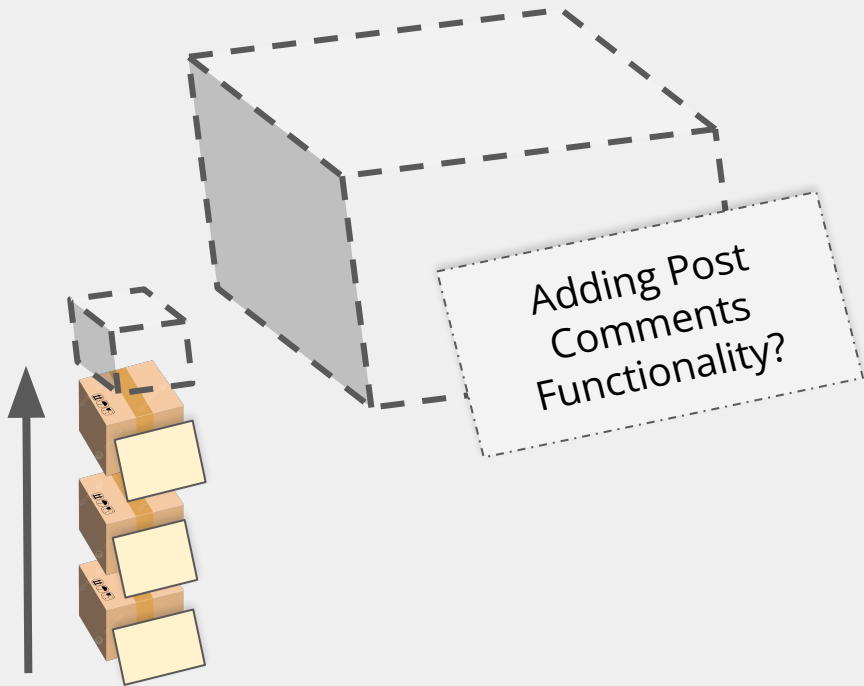


Staging & Committing

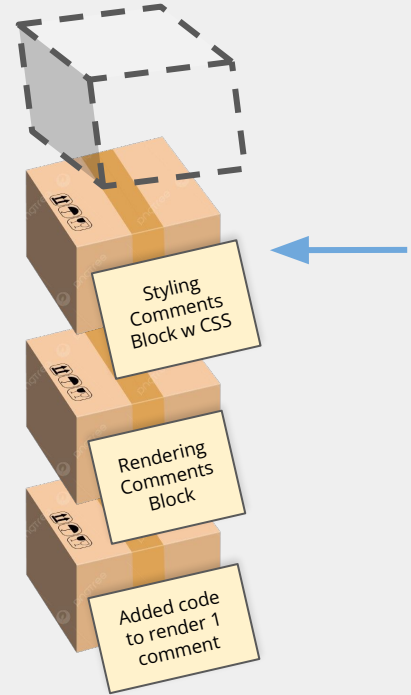


Add a new commit ? 🧑💻

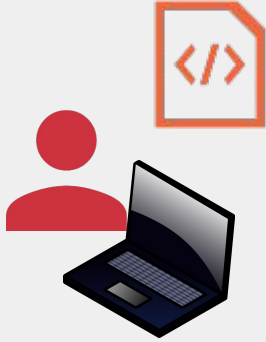
Staging & Committing



Staging & Committing



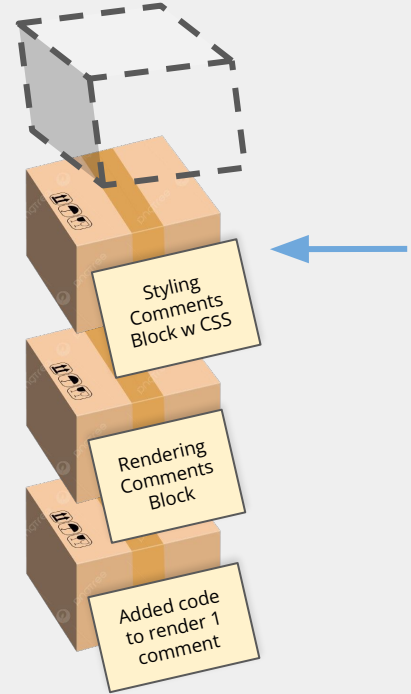
Staging & Committing Pt 1: Code Changes



```
const NewComment = (props) => {  
  const addComment = (value) => {  
    const body = { parent: props.storyId, content: value };  
    post("/api/comments", body).then((comment) => {  
      // display this comment on the screen  
      props.setNewComment(comment);  
    });  
  };  
};
```

+
NewPostInput.js line 4

Working copy

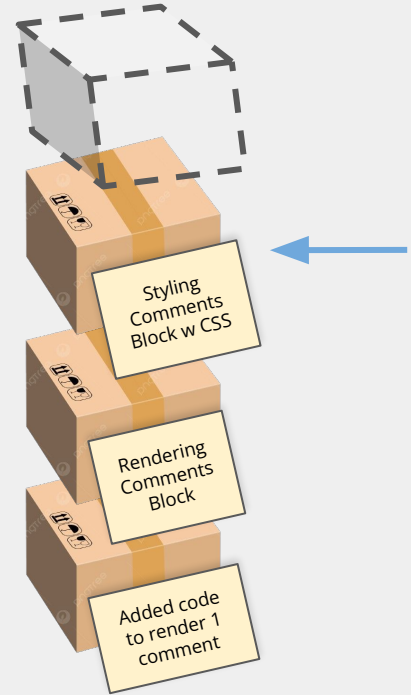


Commit History

Staging & Committing Pt 1: Code Changes

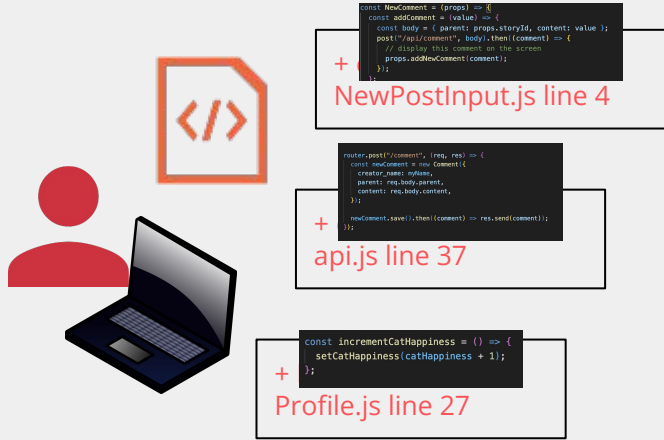


Working copy

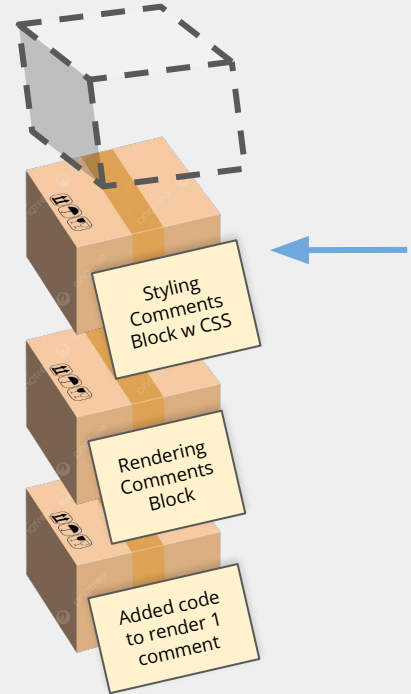


Commit History

Staging & Committing Pt 1: Code Changes



Working copy

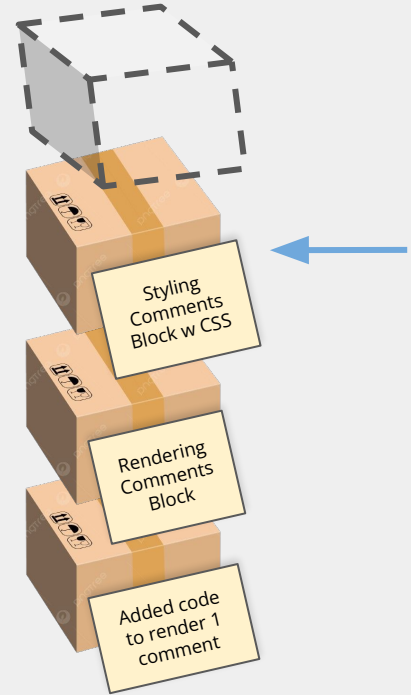


Commit History

Staging & Committing Pt 1: Code Changes

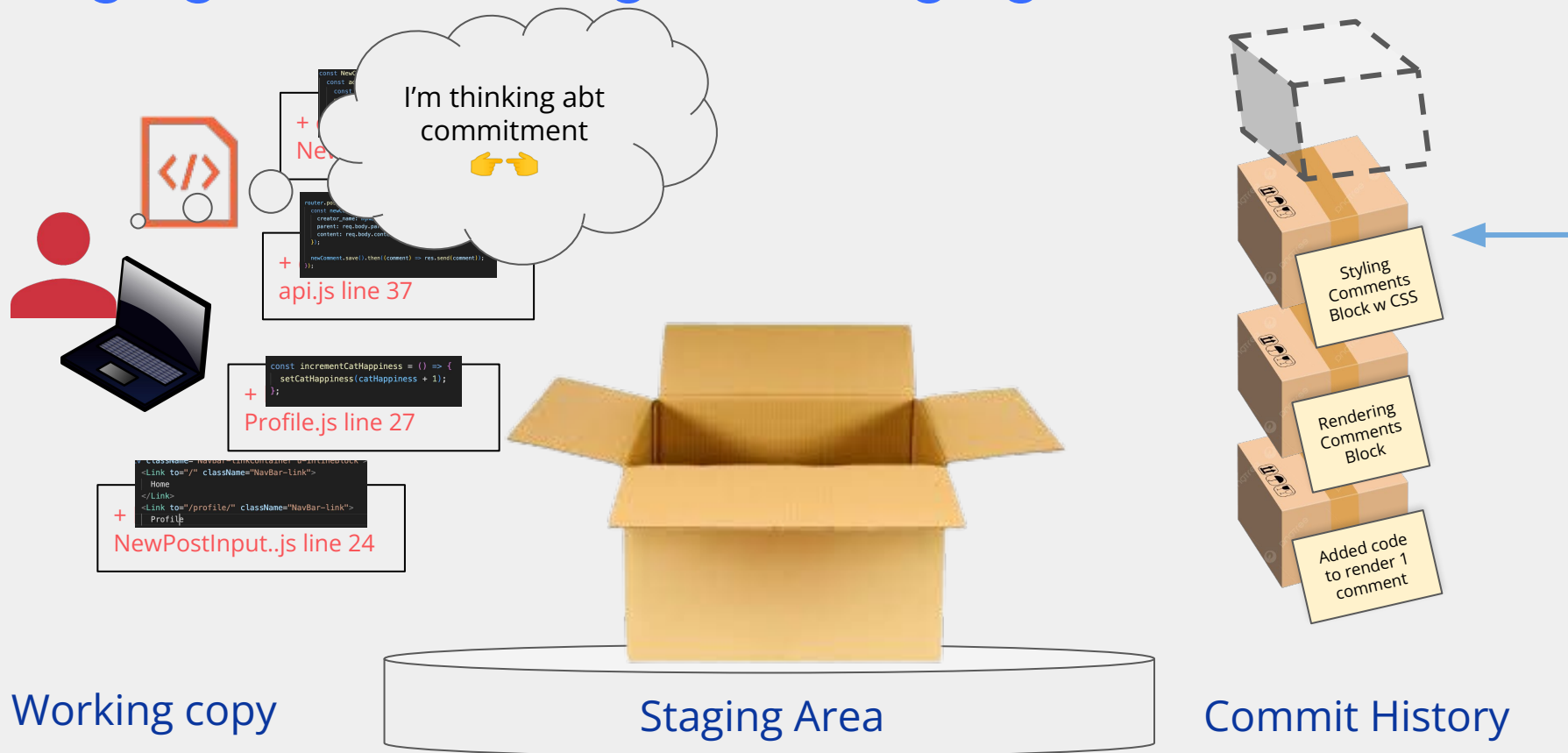


Working copy



Commit History

Staging & Committing Pt 2: Staging



Staging & Committing Pt 2: Staging



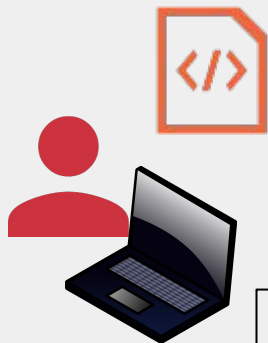
Staging & Committing Pt 2: Staging

```
git add NewPostInput.js
```



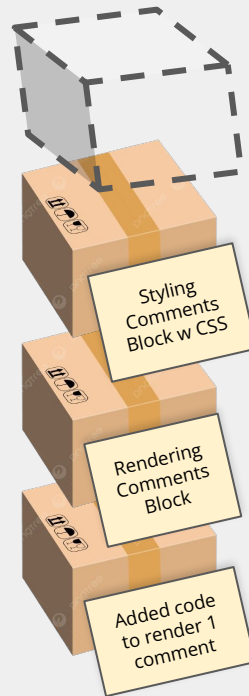
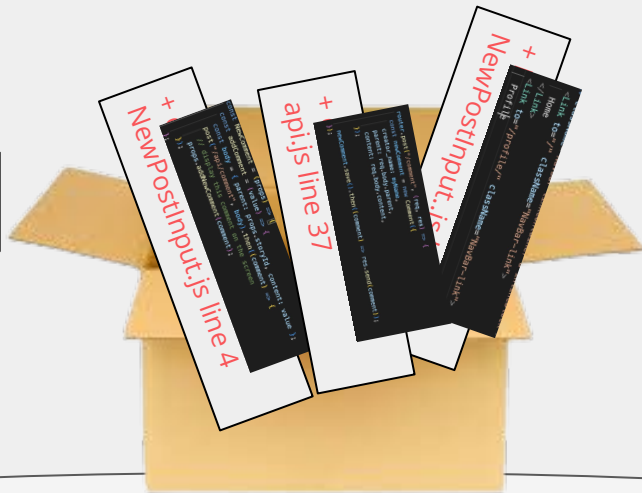
Staging & Committing Pt 2: Staging

```
git add NewPostInput.js  
git add api.js
```



```
const incrementCatHappiness = () => {  
  setCatHappiness(catHappiness + 1);  
};
```

+ Profile.js line 27

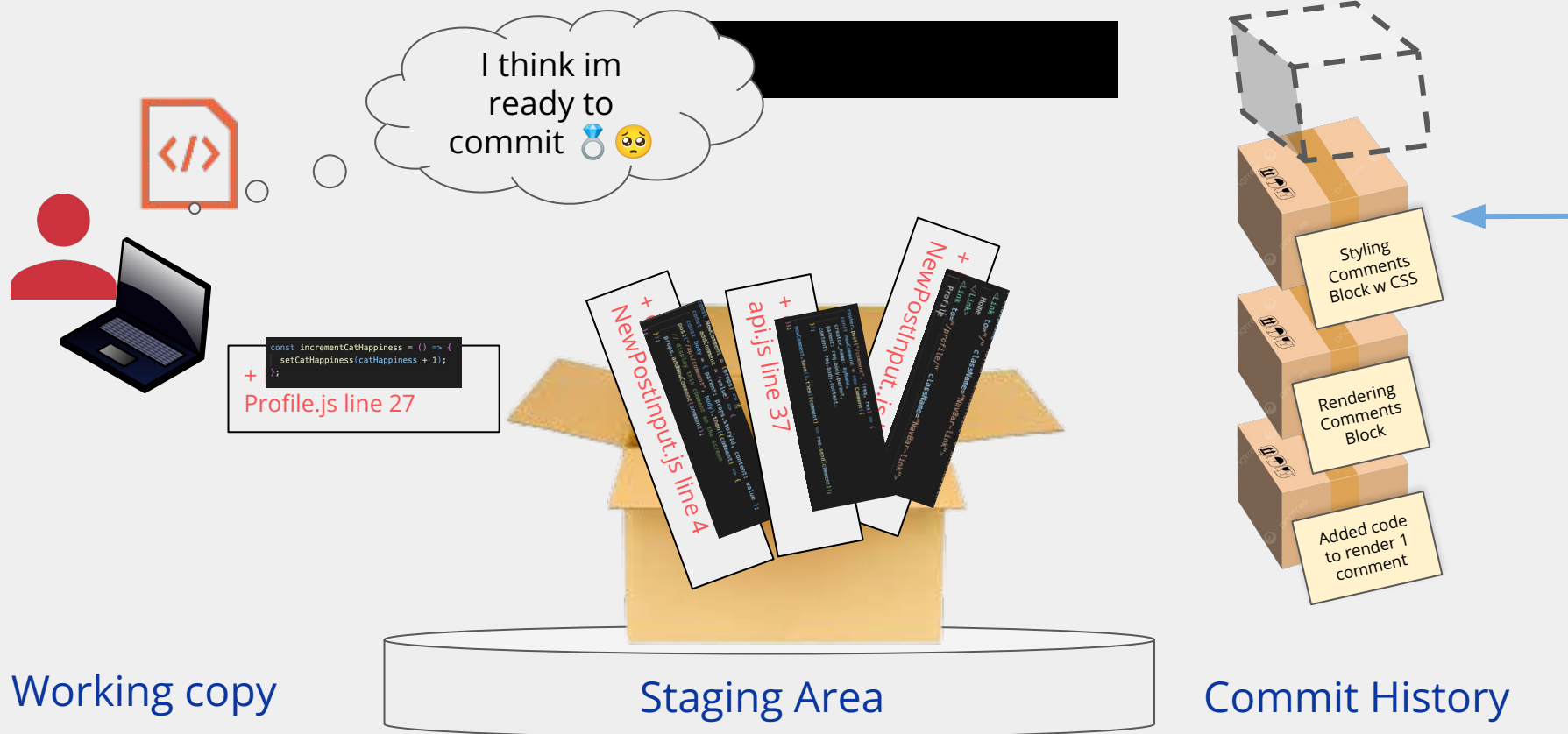


Working copy

Staging Area

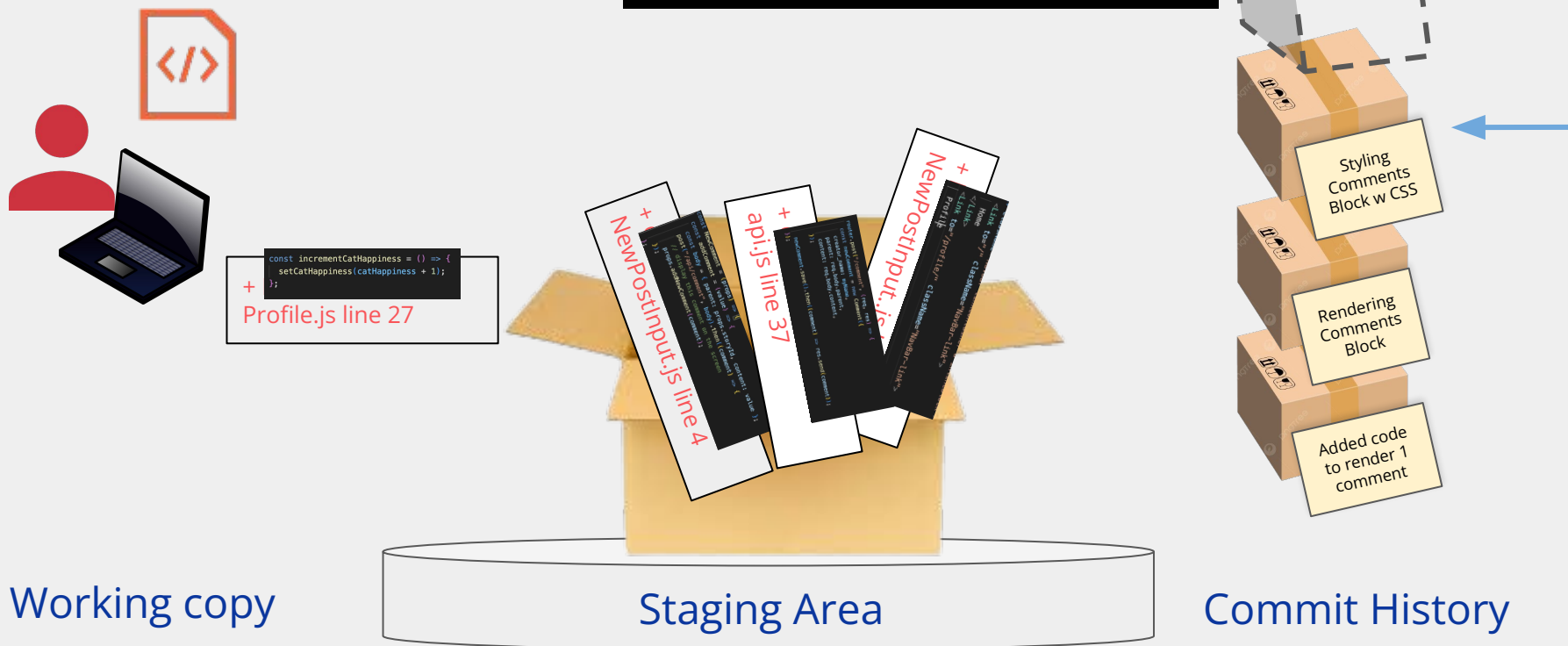
Commit History

Staging & Committing Pt 3: Commit



Staging & Committing Pt 3: Commit

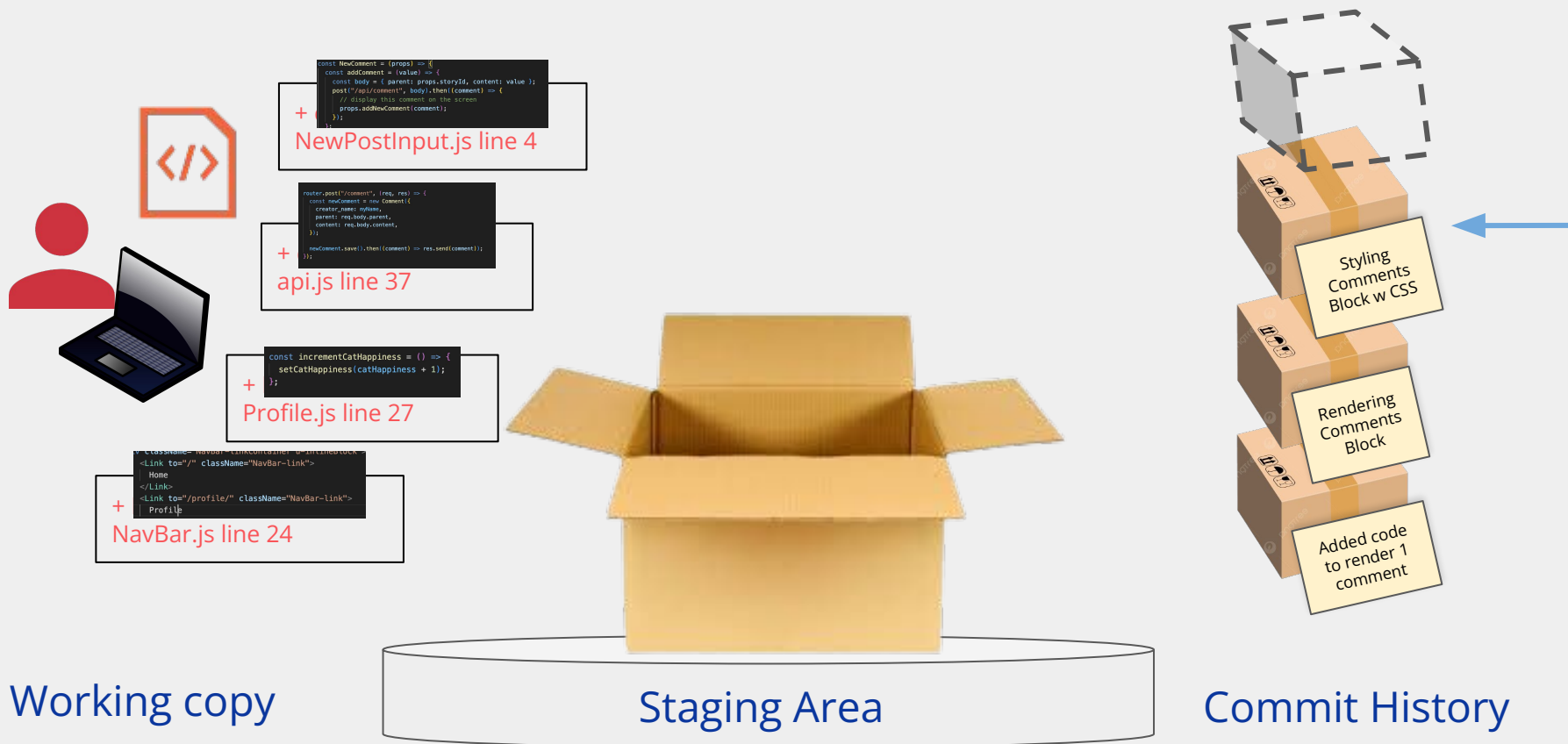
```
git commit -m <commit msg>
```



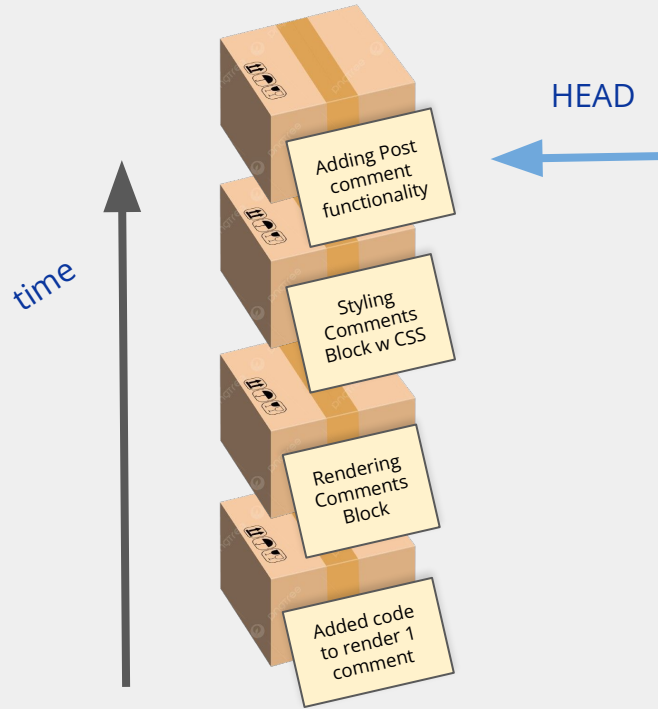
Staging & Committing



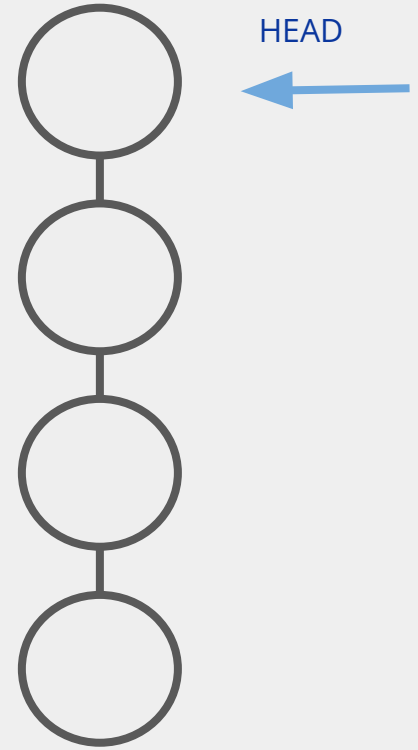
Staging & Committing Recap



Commit History -> Object Graph

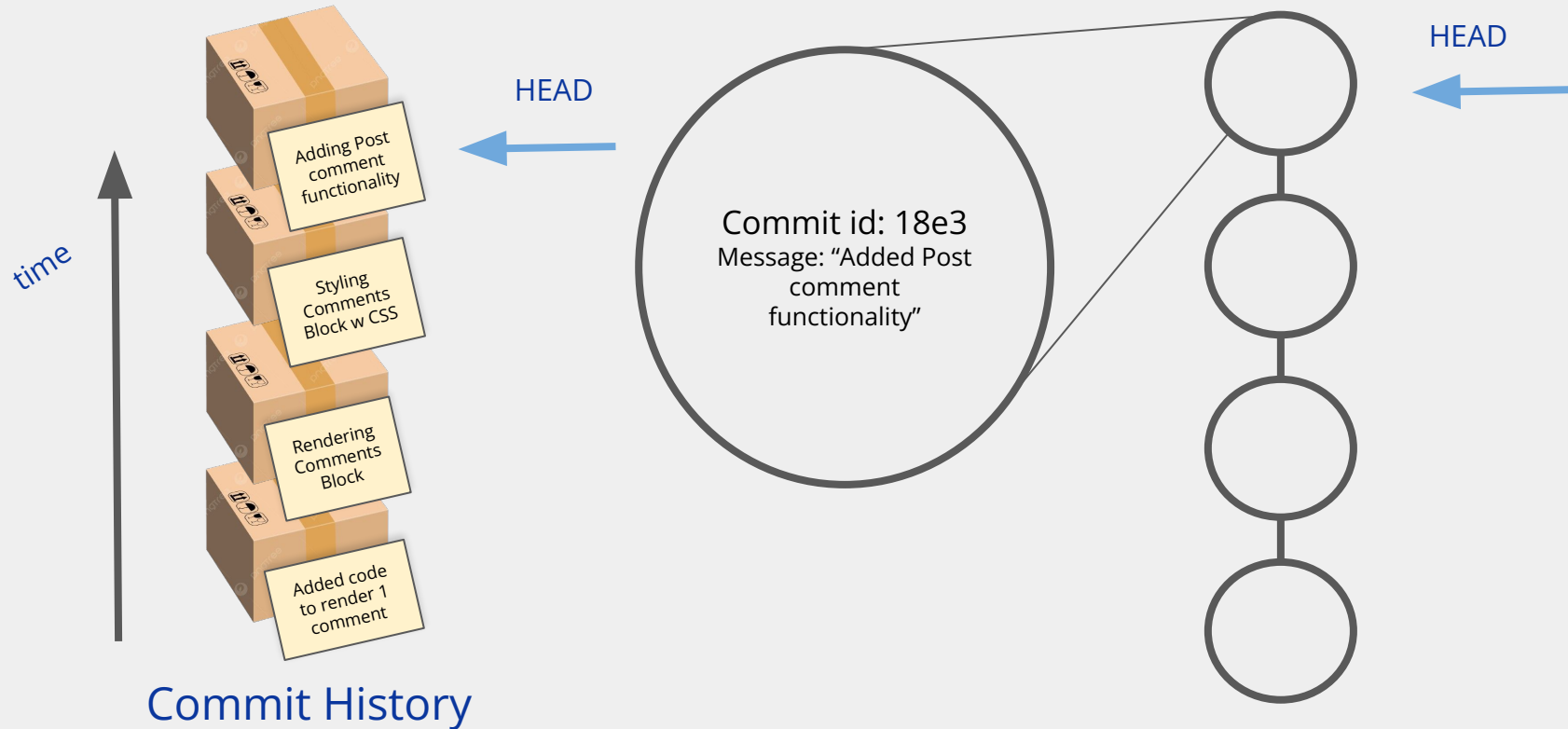


Commit History



Object Graph

Commit History -> Object Graph



Branching

The need for branching and merging

tony: "vibing working on my new and improved home page "

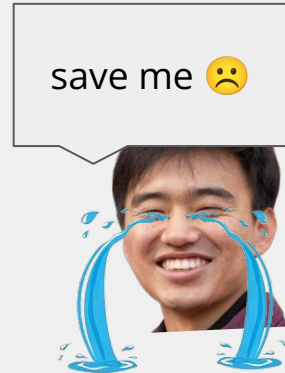
andrew: "can u help me fix this bug on the home page"

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?



The need for branching and merging

tony: **task A**: *Needs to work on his project*

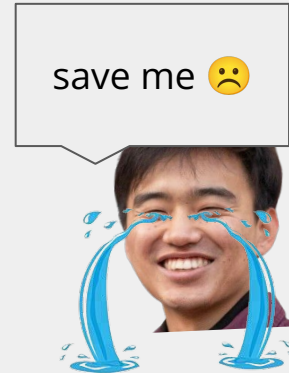
andrew: "can u help me"

tony: **task B**: *Needs to help Andrew*

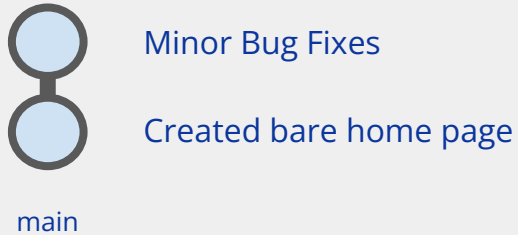
Abby: "tony do this for me"

tony: **task C**: *cries*

How do we work on multiple features at once?



Commits and Branches



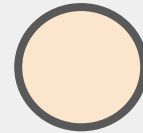
Legend:



Task A



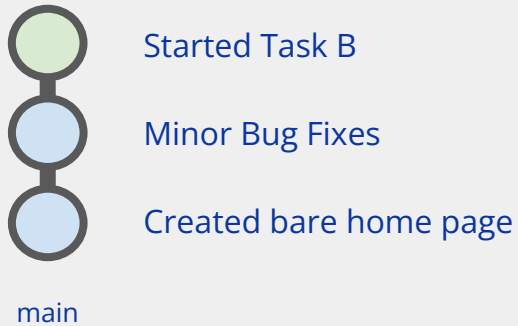
Task B



Task C



Commits and Branches



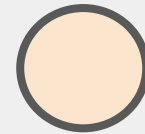
Legend:



Task A



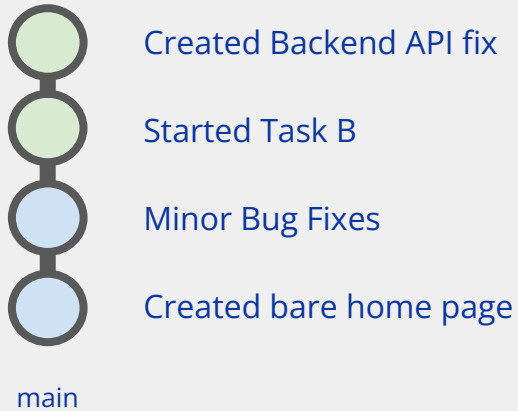
Task B



Task C



Commits and Branches



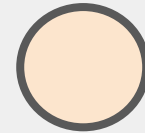
Legend:



Task A



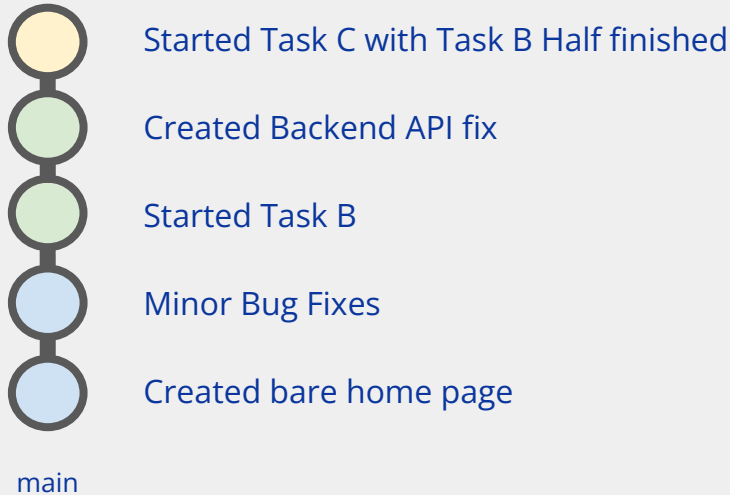
Task B



Task C



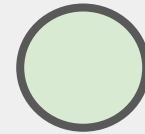
Commits and Branches



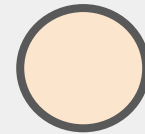
Legend:



Task A



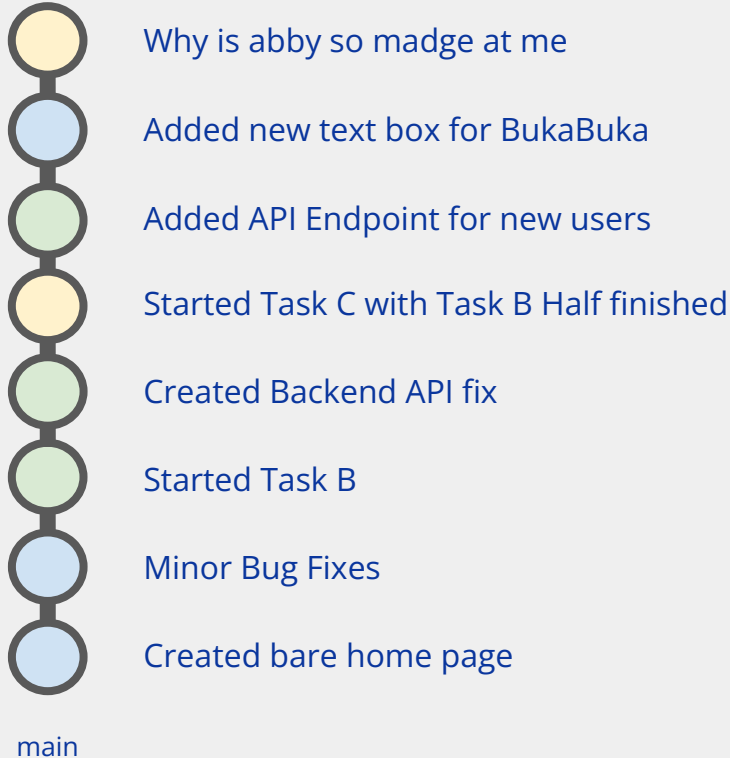
Task B



Task C



Commits and Branches



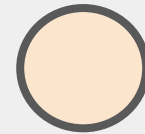
Legend:



Task A



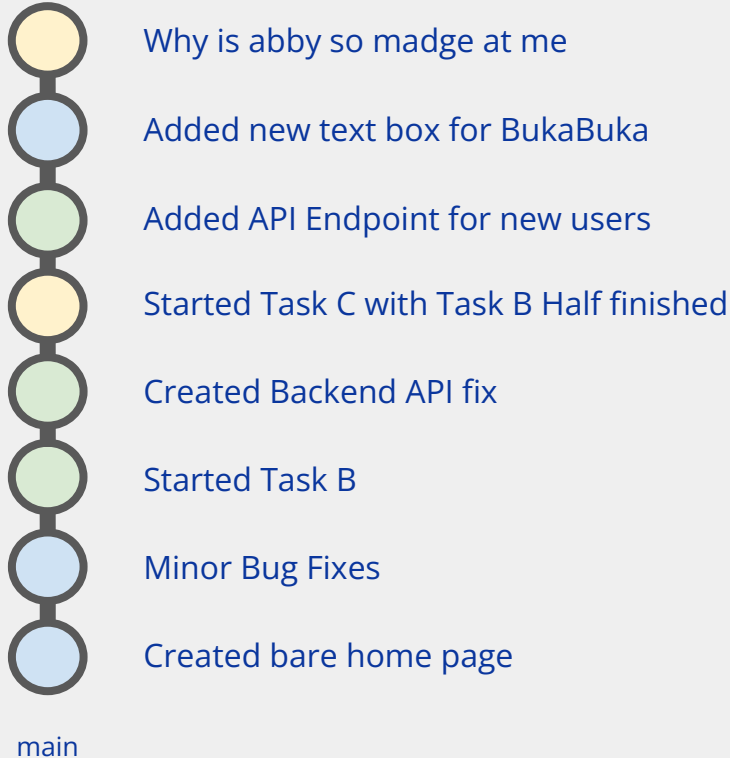
Task B



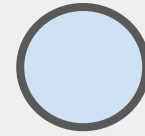
Task C



Commits and Branches



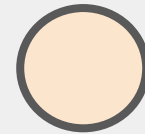
Legend:



Task A



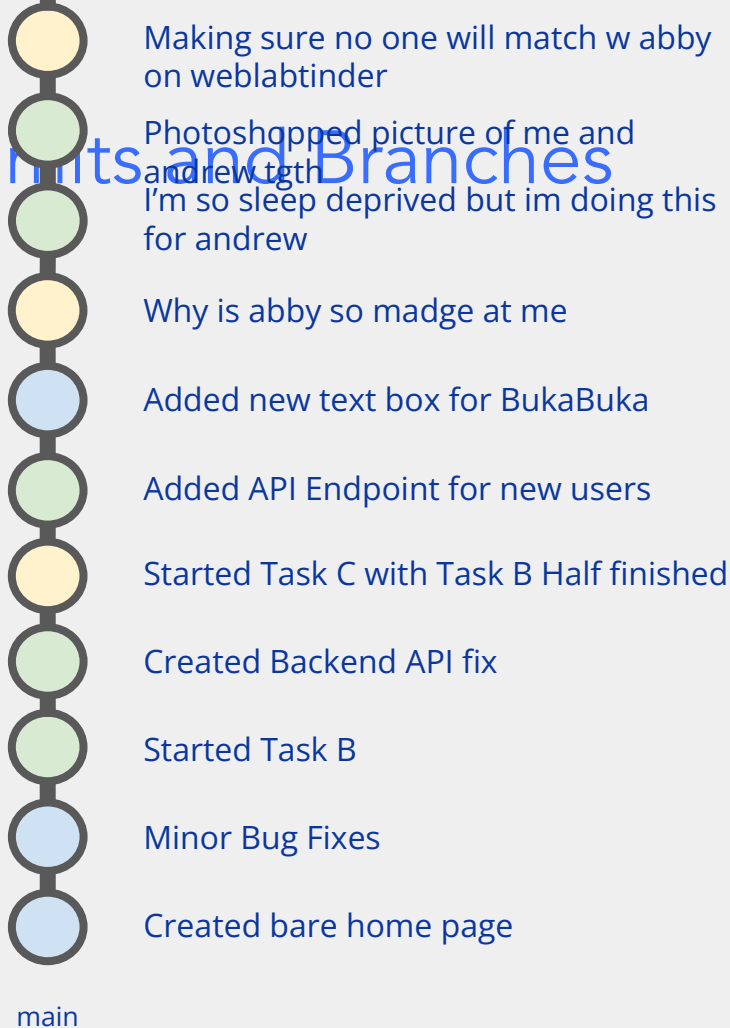
Task B



Task C



Commits and Branches



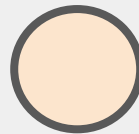
Legend:



Task A



Task B



Task C



Commits and Branches

- Making sure no one will match w abby on weblabtinder
- Photoshopped picture of me and andrew tgh
- I'm so sleep deprived but im doing this for andrew
- Why is abby so madge at me
- Added new text box for BukaBuka
- Added API Endpoint for new users
- Started Task C with Task B Half finished
- Created Backend API fix
- When did bug #6 start on Task C?
- Minor fixes
- Created bare home page

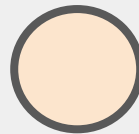
Legend:



Task A



Task B



Task C



Commits and Branches

Making sure no one will match w abby
on weblabtinder

Photoshopped picture of me and
andrew tgth
I'm so sleep deprived but im doing this
for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished

Created Backend API fix

When did bug #6
start on Task C?

Mine & Fixes

Created bare home page

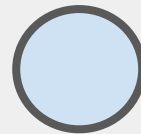
Legend:



Task A



Task B



Task C



Commits and Branches

Making sure no one will match w abby on weblabtinder

Photoshopped picture of me and andrew tgth
I'm so sleep deprived but im doing this for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished

Created Backend API fix

When did bug #6 start on Task C?

Mine & Fixes

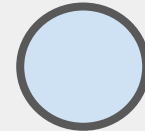
Created bare home page



Legend:



Task A



Task B



Task C



Commits and Branches

Making sure no one will match w abby on weblabtinder

Photoshopped picture of me and andrew tggth
I'm so sleep deprived but im doing this for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished ✓

Created Backend API fix

When did bug #6 start on Task C?

Mine & Fixes

Created bare home page

Legend:



Task A



Task B



Task C



Commits and Branches

Making sure no one will match w abby
on weblabtinder

Photoshopped picture of me and
andrew tggth
I'm so sleep deprived but im doing this
for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished

Created Backend API fix

When did bug #6
start on Task C?

Minor fixes

Created bare home page

Legend:



Task A



Task B



Task C



Commits and Branches

Making sure no one will match w abby
on weblabtinder

Photoshopped picture of me and
andrew tgth
I'm so sleep deprived but im doing this
for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished

Created Backend API fix

When did bug #6
start on Task C?

Mine & Fixes

Created bare home page

Legend:



Task A



Task B



Task C



Commits and Branches



Making sure no one will match w abby on weblabtinder

Photoshopped picture of me and andrew tgth
I'm so sleep deprived but im doing this for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished

Created Backend API fix

When did bug #6 start on Task C?

Mine & Fixes

Created bare home page

Legend:



Task A



Task B



Task C



Commits and Branches

Making sure no one will match w abby
on weblabtinder

Photoshopped picture of me and
andrew tgth
I'm so sleep deprived but im doing this
for andrew

Why is abby so madge at me

Added new text box for BukaBuka

Added API Endpoint for new users

Started Task C with Task B Half finished

Created Backend API fix

When did bug #6
start on Task C?

Mine & Fixes

Created bare home page

Legend:



Task A



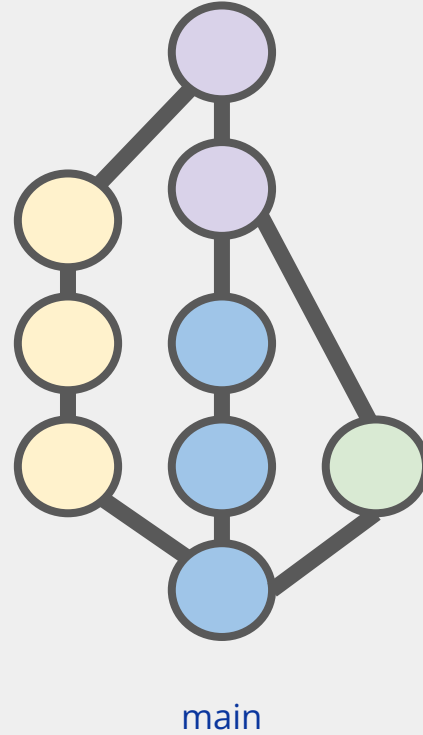
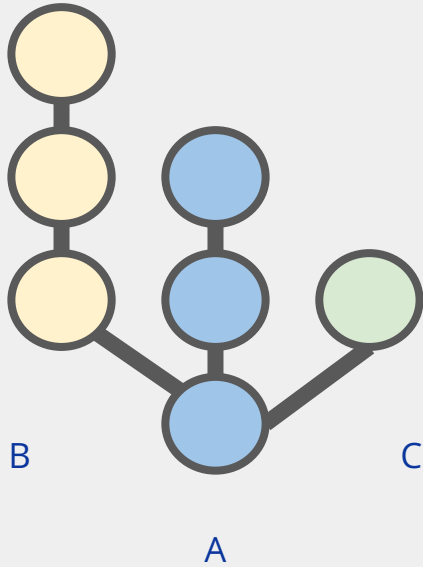
Task B



Task C



The need for branching and merging

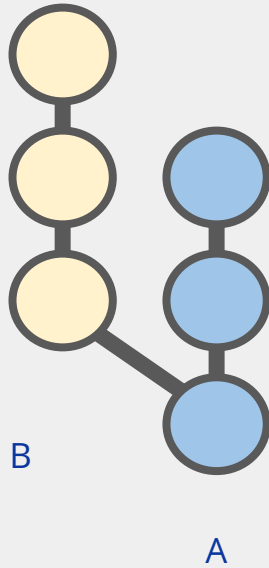


Merging *but good

Very smooth merge

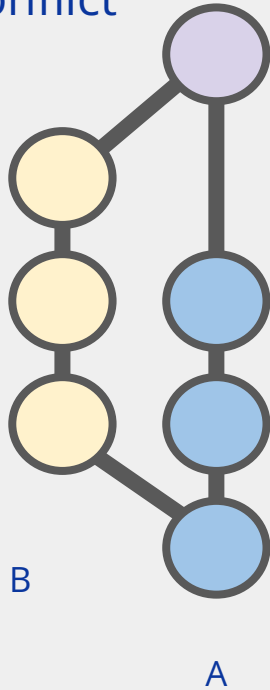
Merging *but bad

Merge conflict



Merging *but bad

Merge conflict



Merging *but bad

Merge conflict



A



remote

```
git clone <link>
```



main



main



Remote Server (github)

main



```
git clone <link>
```



main



main

Remote Server (github)



HEAD

main



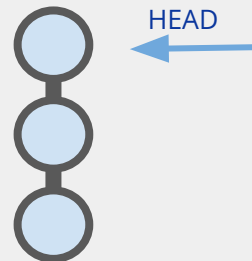
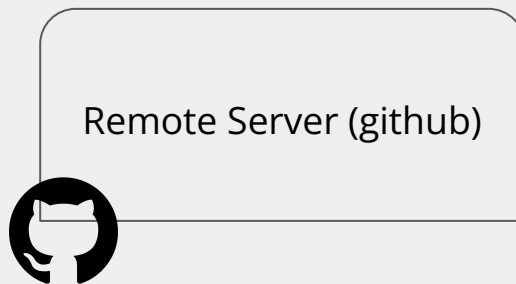

```
git add <file>  
git commit -m "message"
```



main



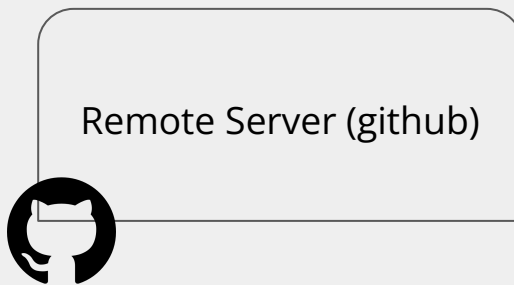
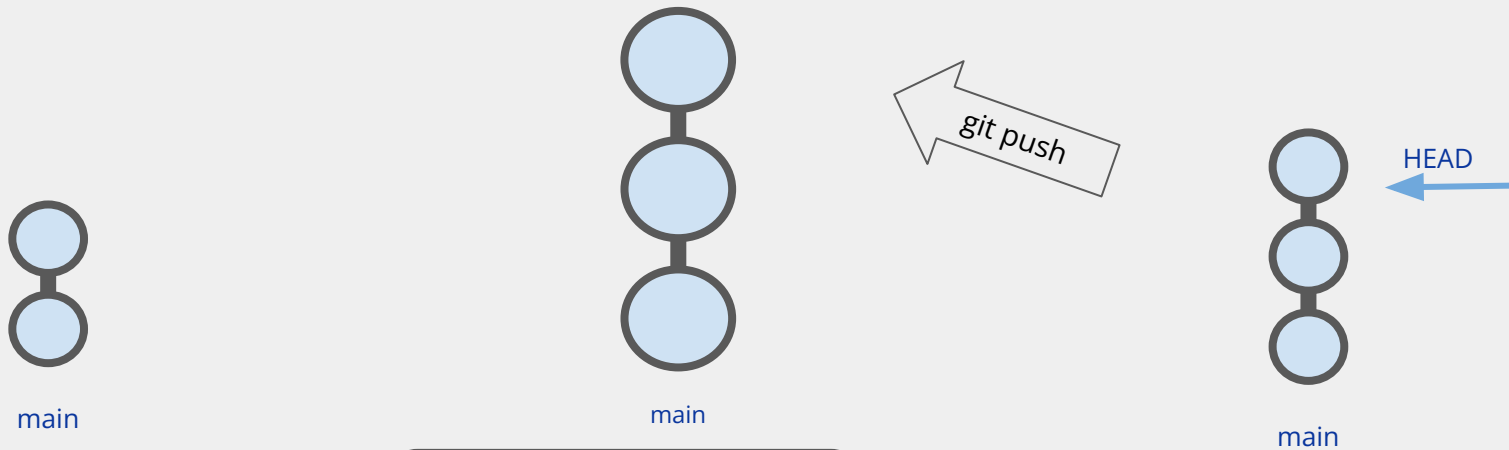
main



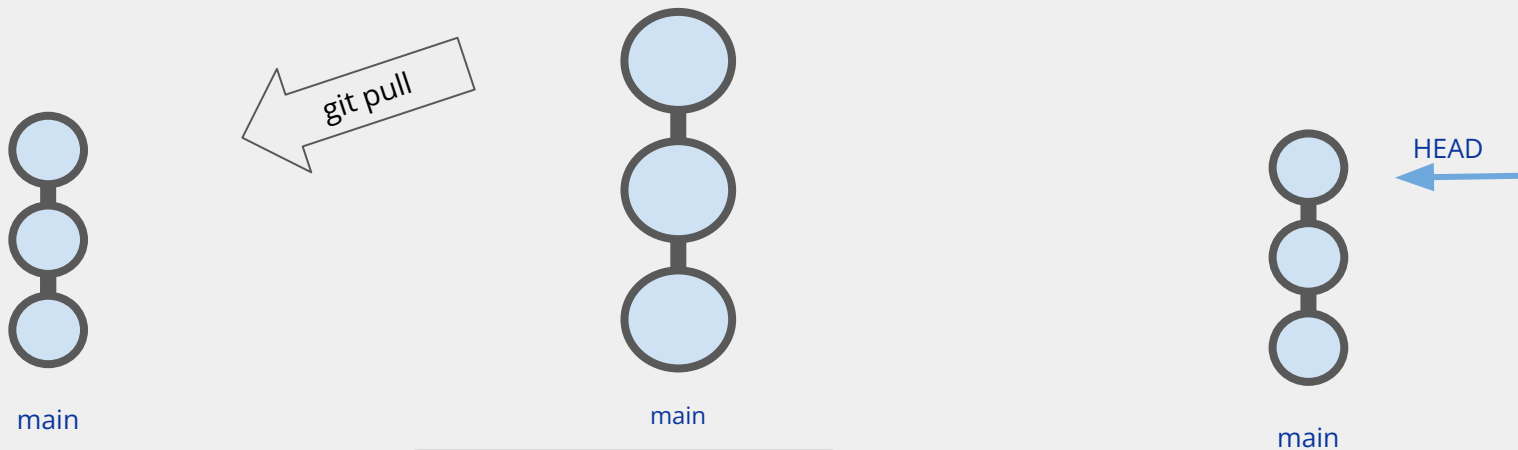
main



git push



git pull

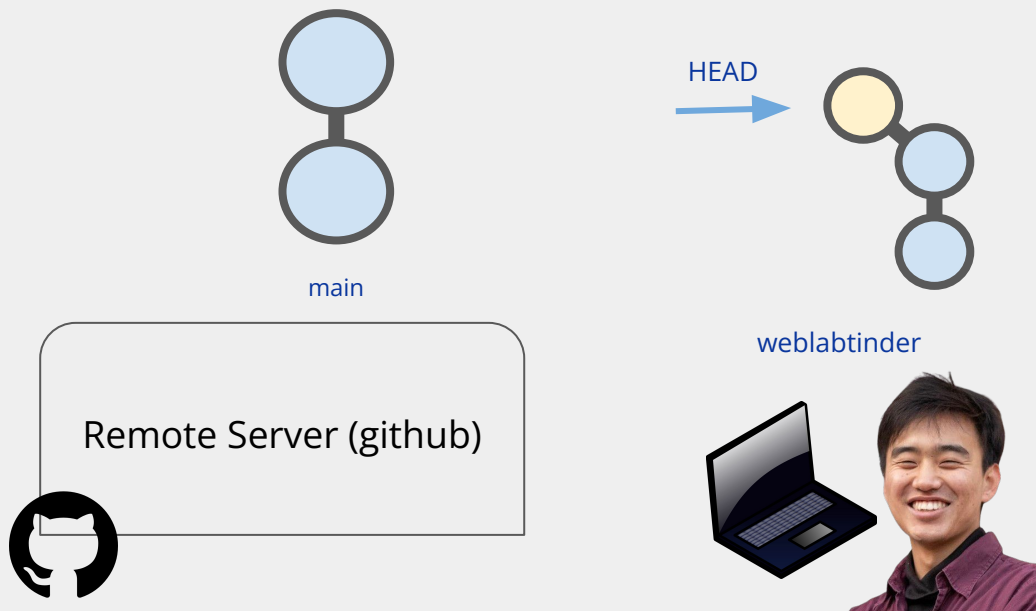


Basic Workflow with Remote

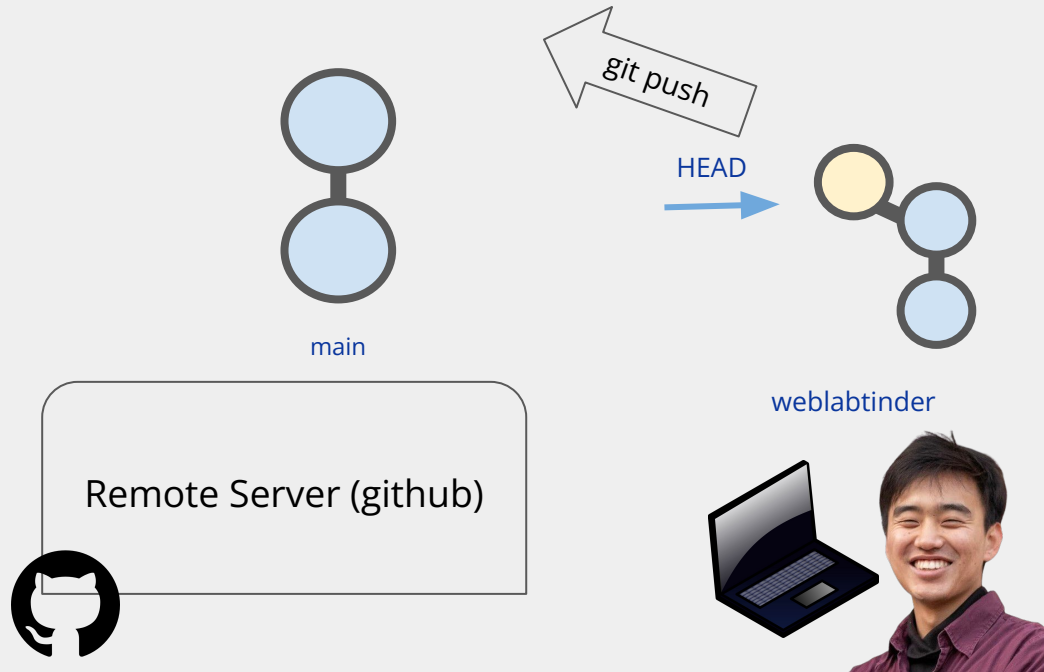
1. Pull
2. Work on stuff/add/commit
3. Pull again
4. Push

Remote + local branches

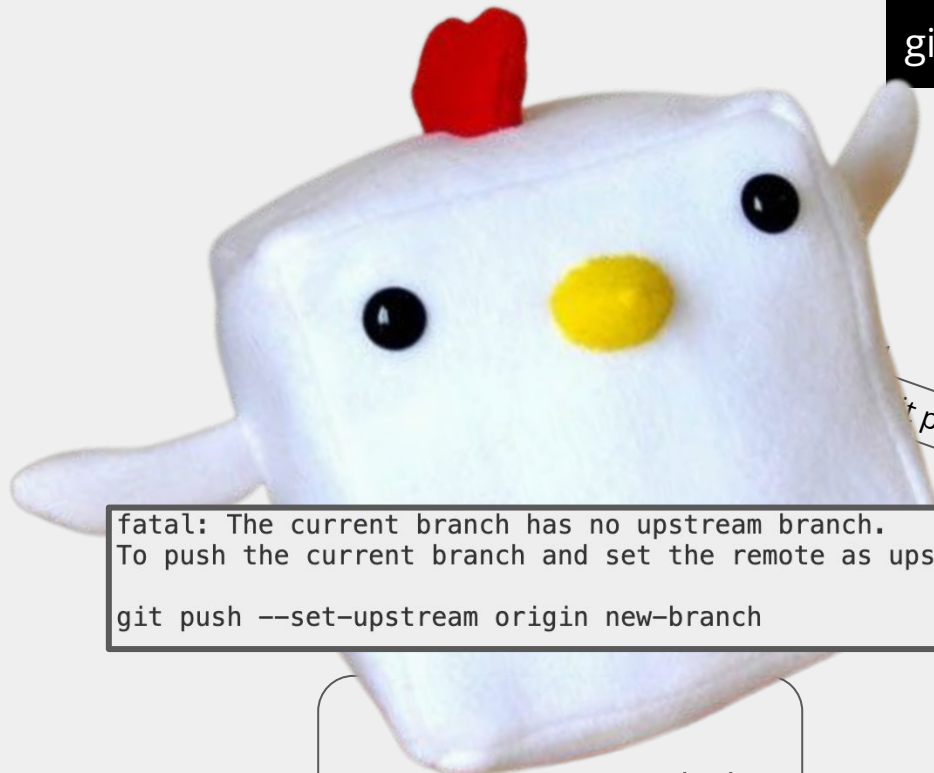
```
git checkout -b weblabtinder  
git add weblabtinder.js  
git commit -m "added tinder"
```



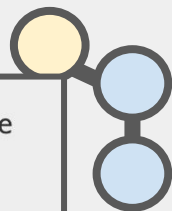
git push



git push



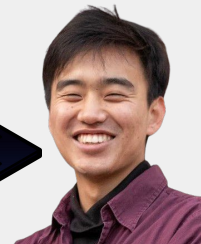
```
fatal: The current branch has no upstream branch.  
To push the current branch and set the remote as upstream, use  
  
git push --set-upstream origin new-branch
```

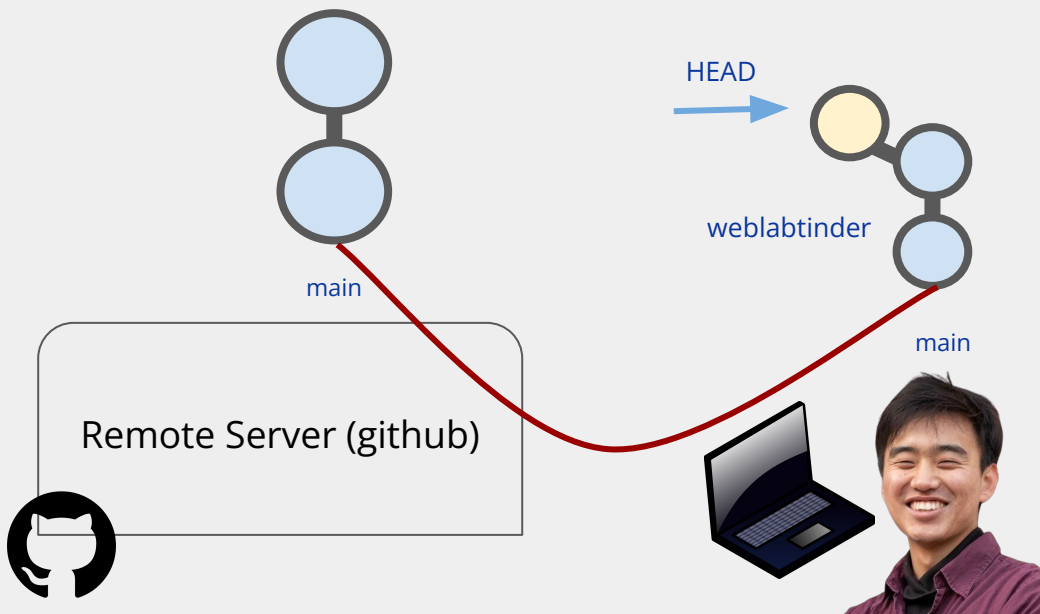


Remote Server (github)

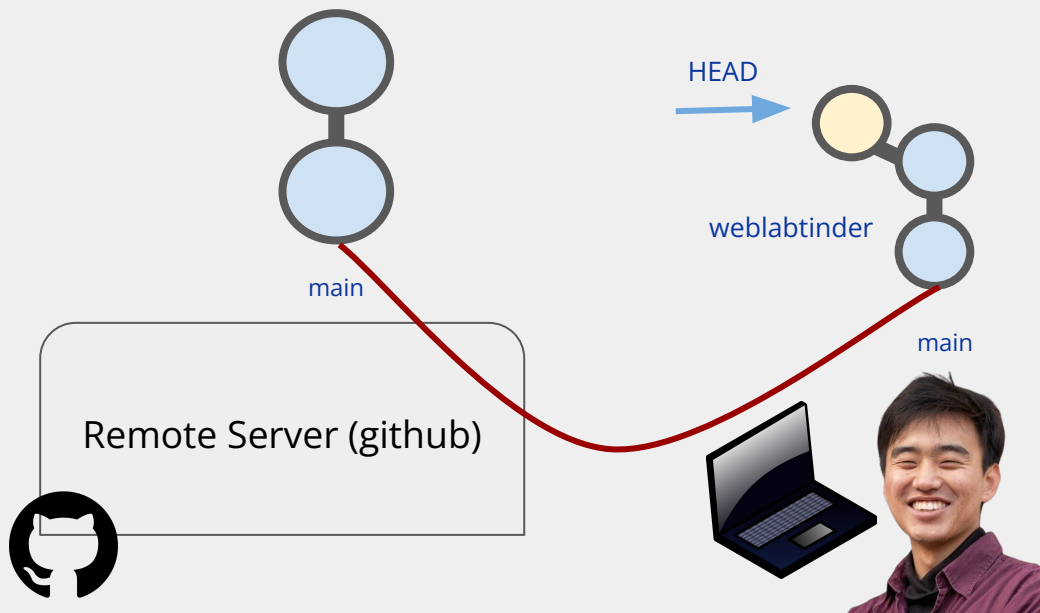


weblabtinder

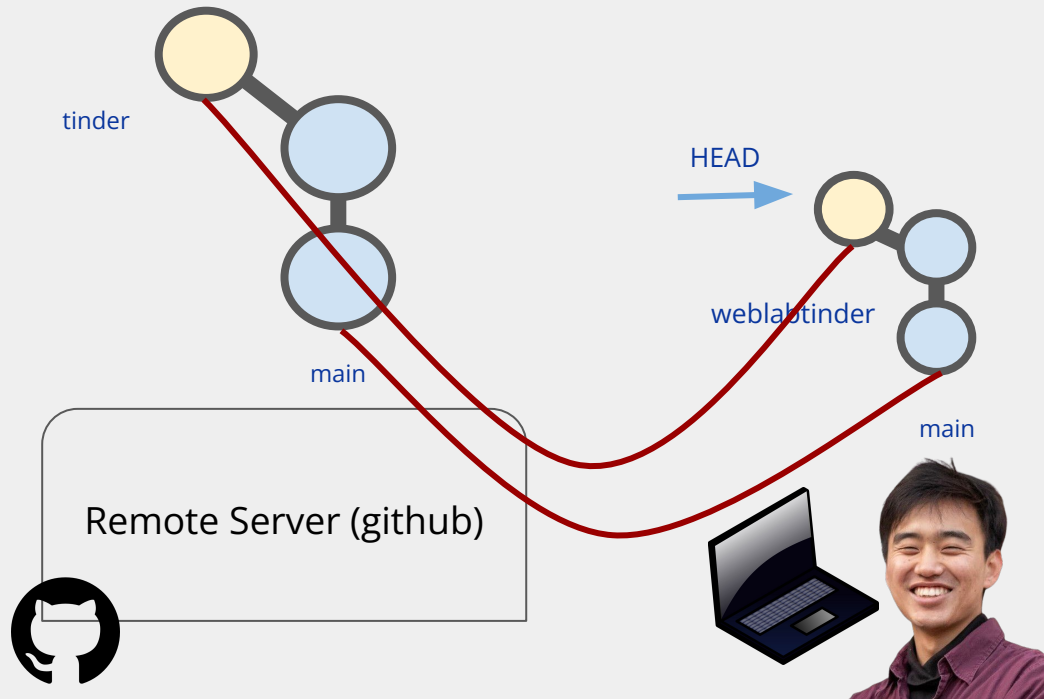


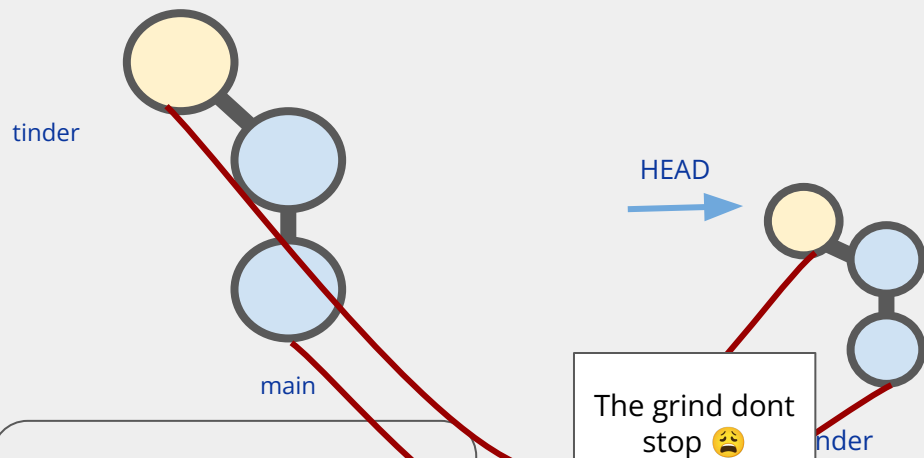


```
git push --set-upstream origin <new-branch>
```



`git push --set-upstream origin tinder`





Remote Server (github)

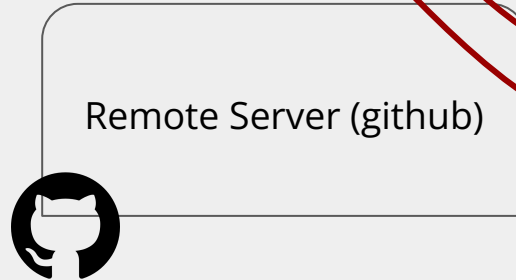
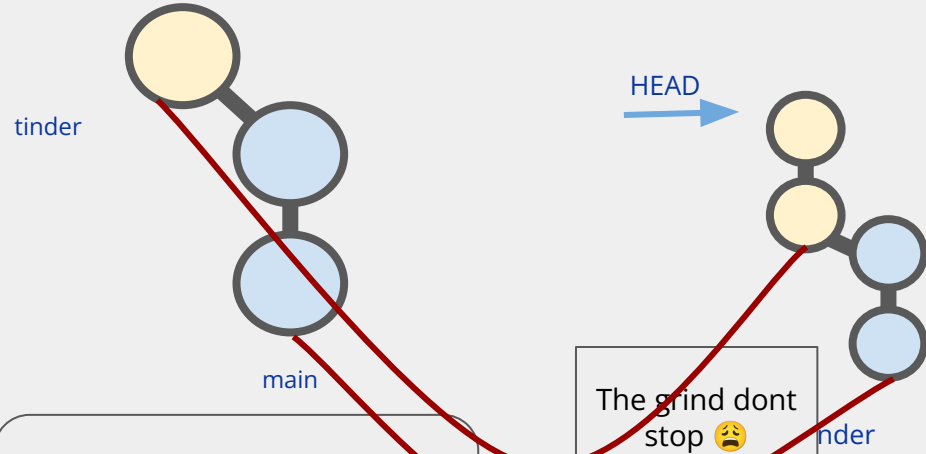
HEAD

A blue arrow pointing to the right, labeled "HEAD".

The grind dont stop 🥴



```
git add tinder_profile.js  
git commit -m "profile"
```



The grind dont stop 🥲



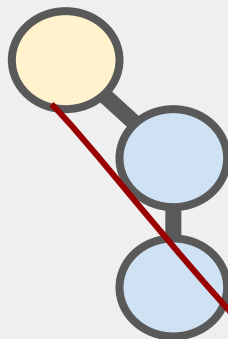
```
git add tinder_profile.js  
git commit -m "profile"  
git add tinder_algo.js  
git commit -m "imported LP"
```



main



tinder

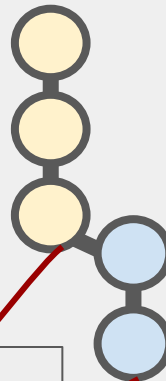
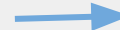


main

Remote Server (github)



HEAD



tinder

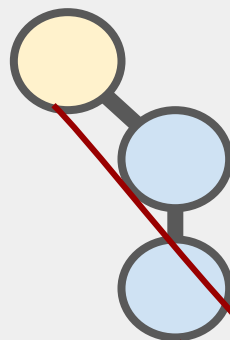
The grind dont
stop 🙄



```
git add tinder_profile.js  
git commit -m "profile"  
git add tinder_algo.js  
git commit -m "imported LP"
```



main

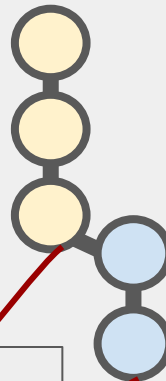
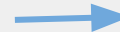


main

Remote Server (github)



HEAD



tinder

The grind dont
stop 😞



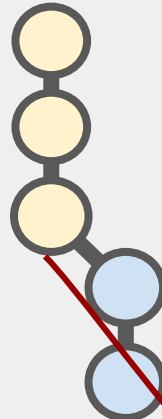
git push



main

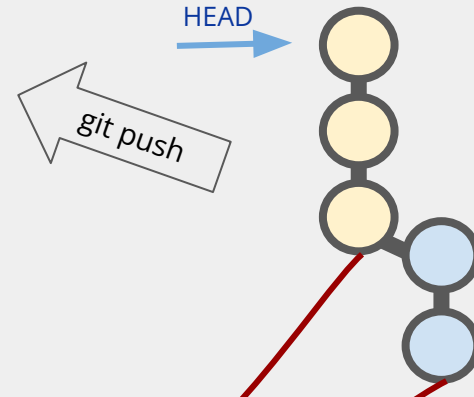


tinder



main

Remote Server (github)



weblabtinder

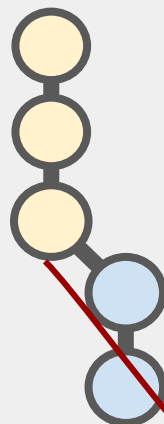




main

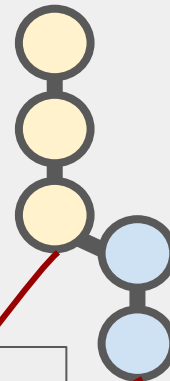
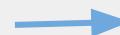


tinder



main

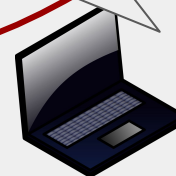
HEAD



tinder

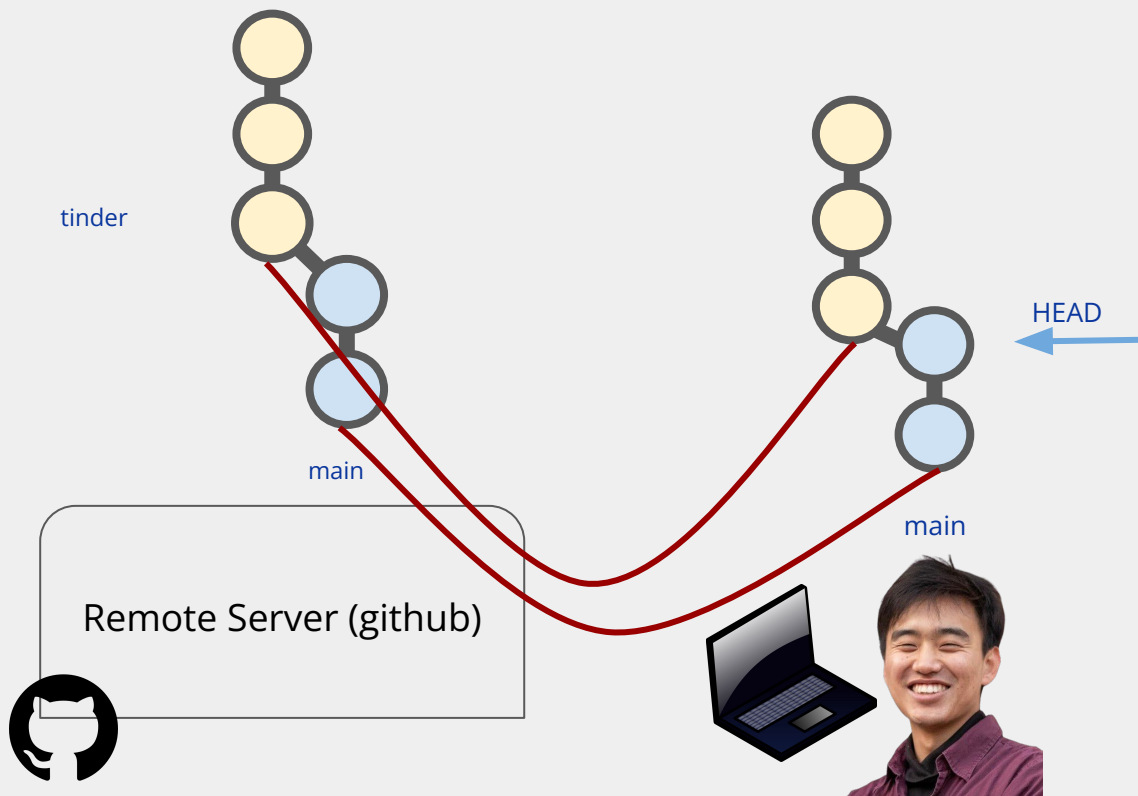
weblab tinder
is ready 🙄

Remote Server (github)

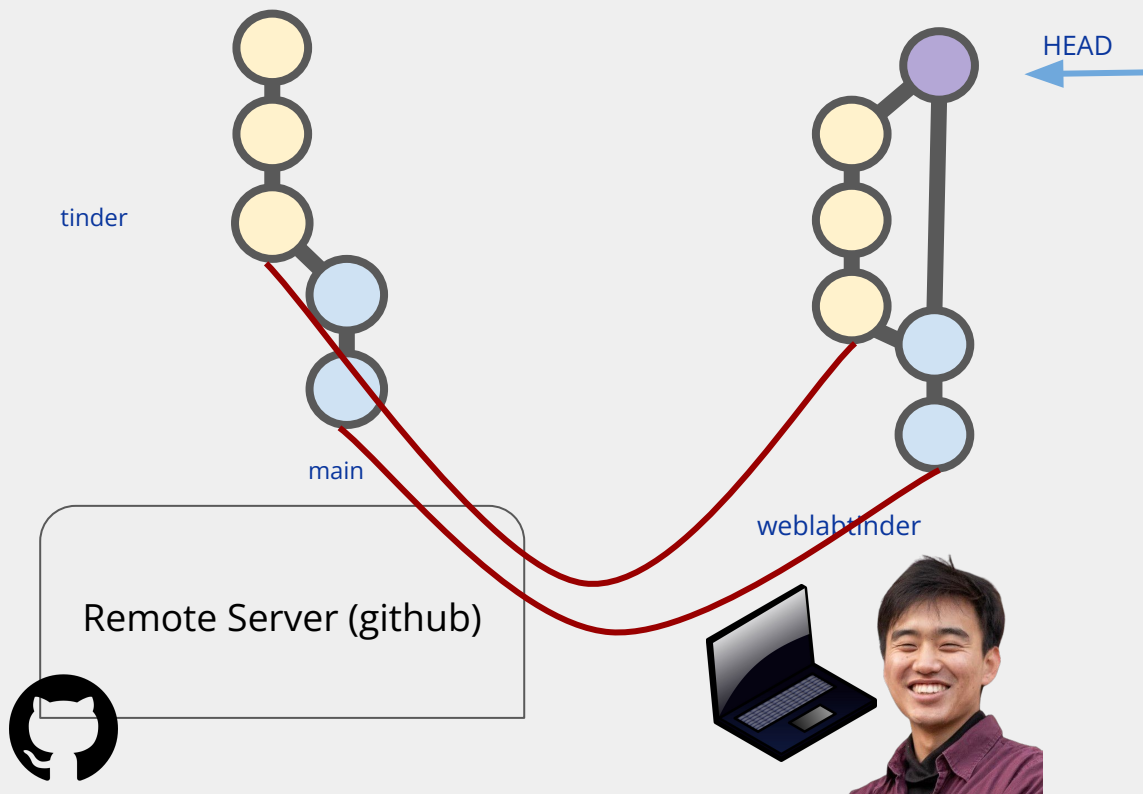


Merging a Local + Remote Branch

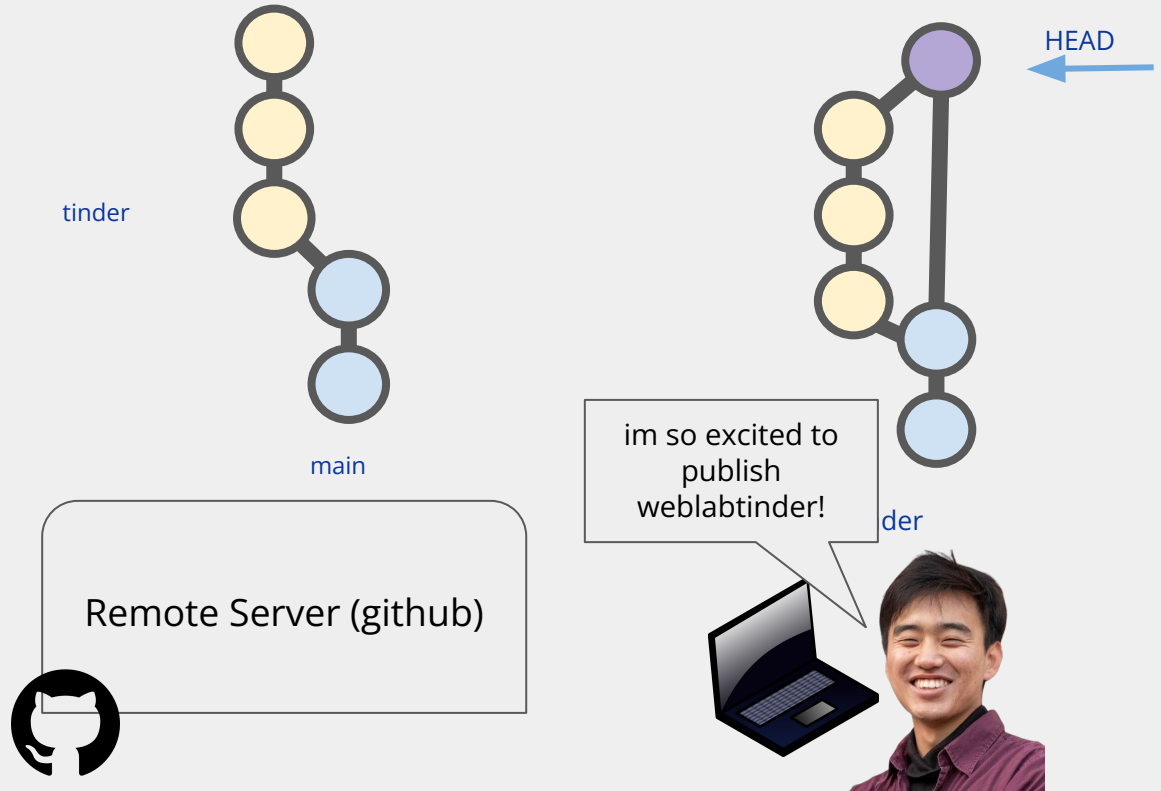
```
git checkout main
```



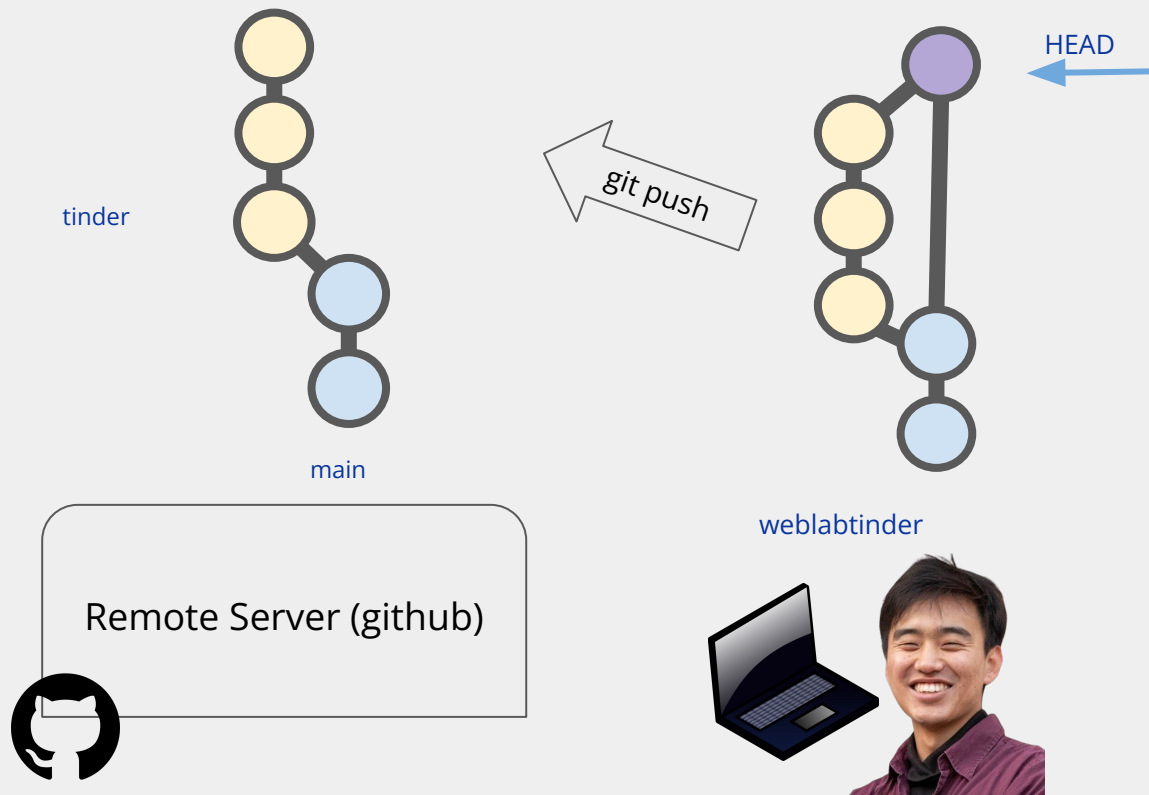
```
git checkout main  
git merge weblabtinder
```



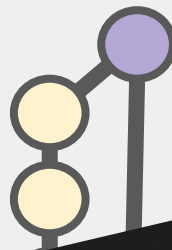
```
git checkout main  
git merge weblabtinder
```



```
git checkout main  
git merge weblabtinder  
git push
```



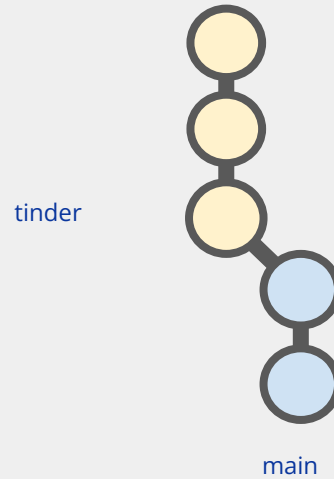
```
git checkout main  
git merge weblabtinder  
git push
```



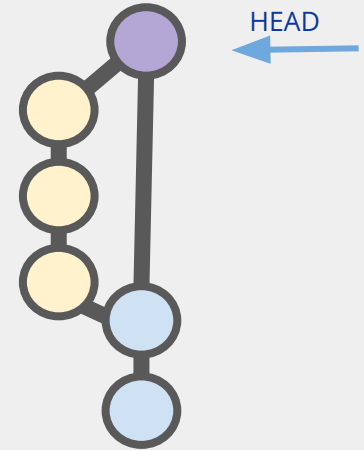
```
to https://github.com/tcui37/weblab-git-example.git  
! [rejected]  
error: failed to push some refs to 'https://github.com/tcui37/weblab-git-example.git'  
hint: Updates were rejected because the remote contains work that you do  
hint: not have locally. This is usually caused by another repository pushing  
hint: to the same ref. You may want to first integrate the remote changes  
hint: (e.g., 'git pull ...') before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```



```
git checkout main  
git merge weblabtinder  
git push
```



Remote Server (github)

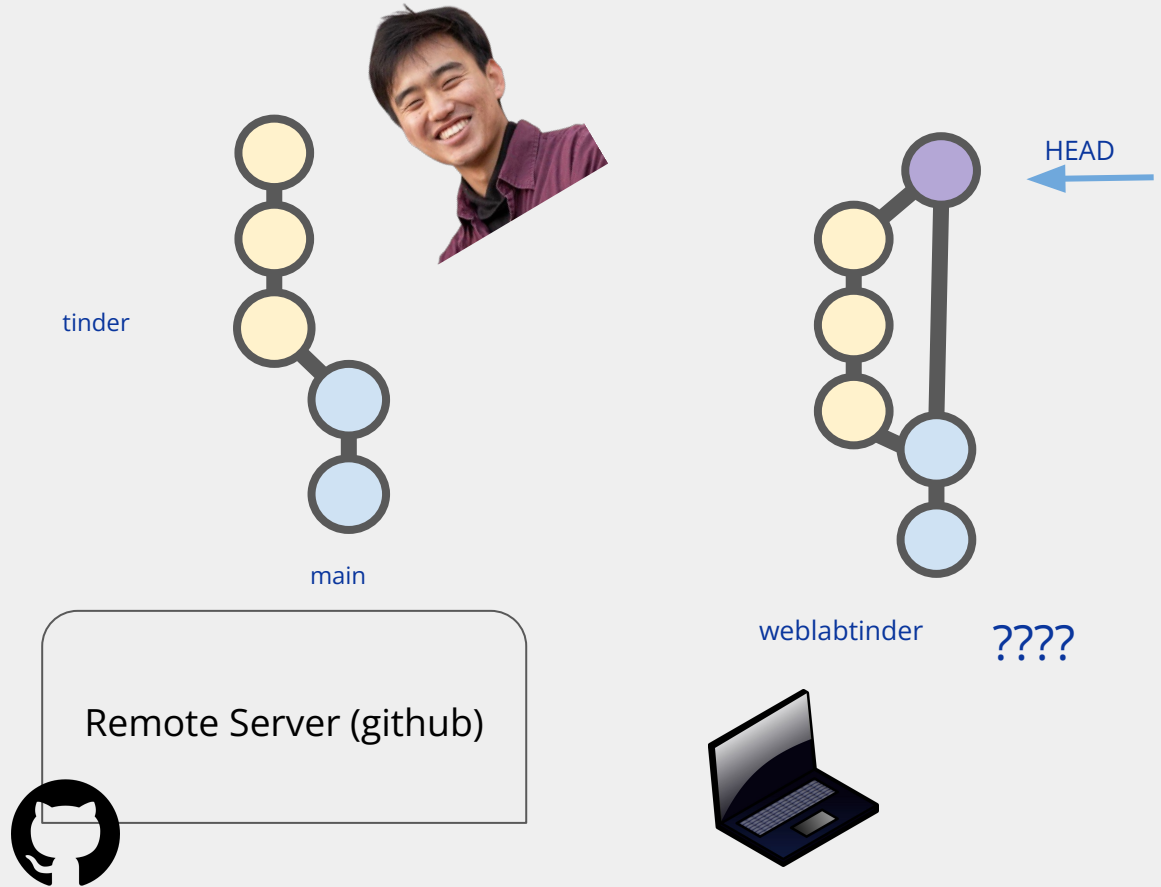


weblabtinder

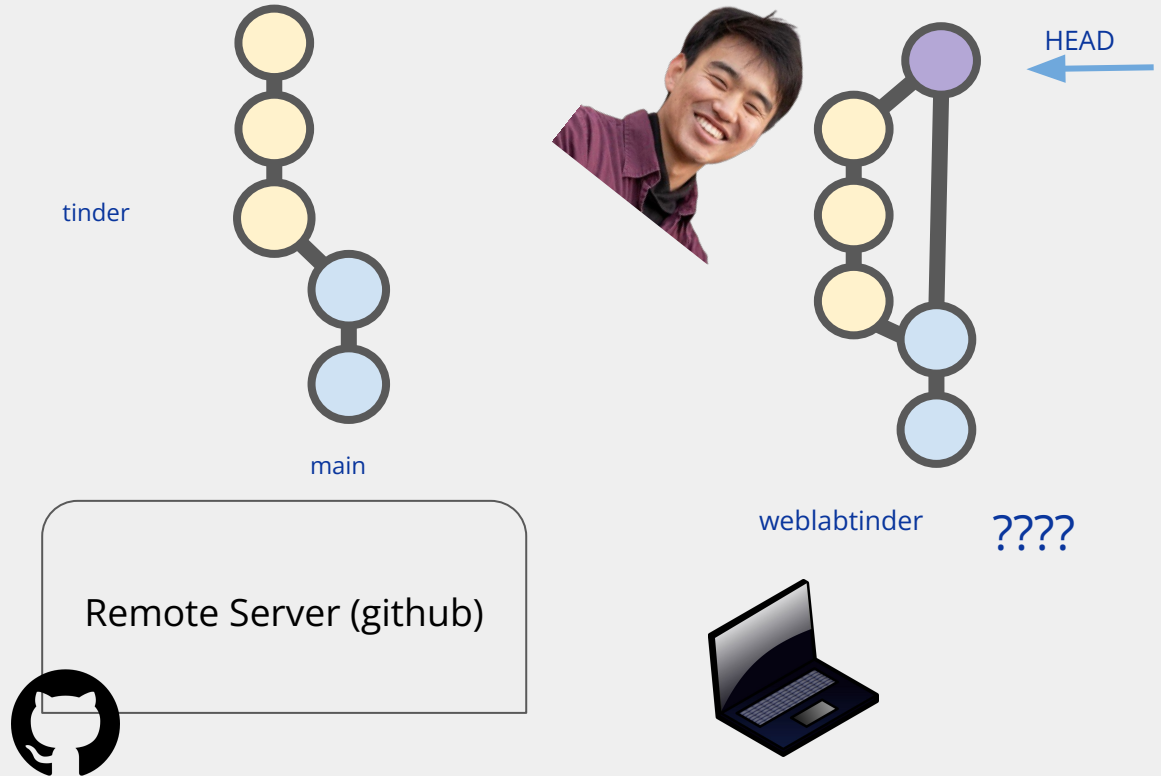
????



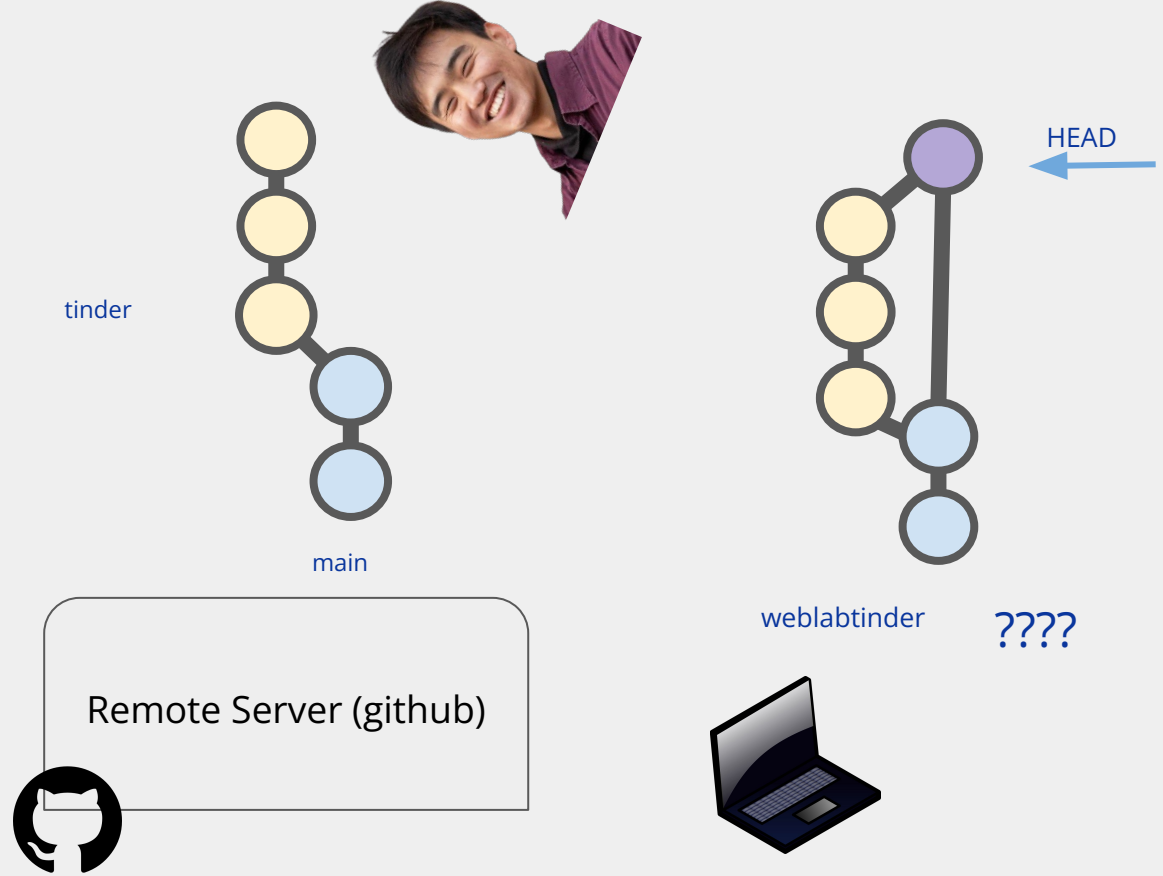
git fetch



git fetch



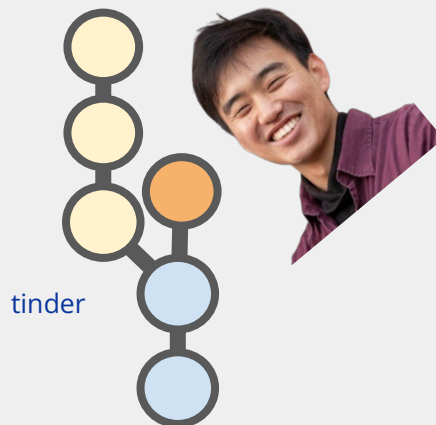
git fetch



git fetch

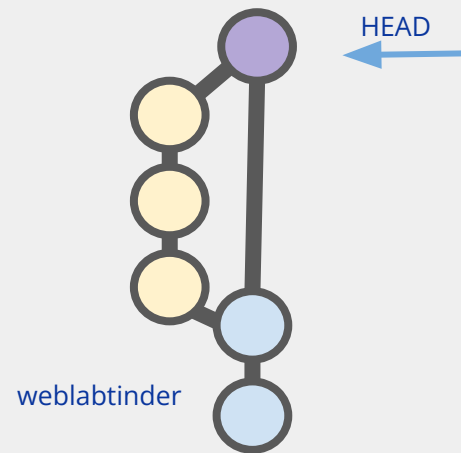


main



tinder

main



weblabtinder

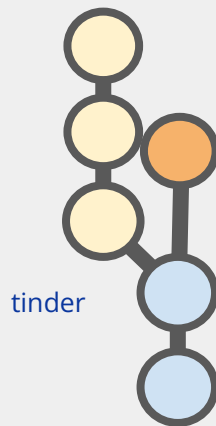
main

Remote Server (github)



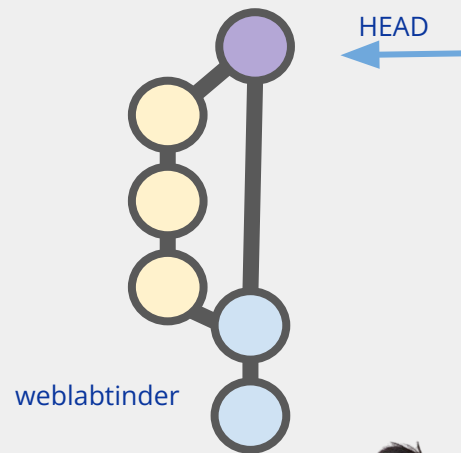
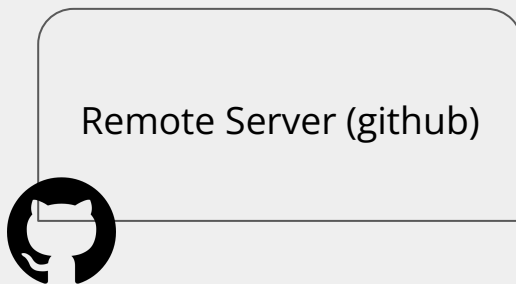


main



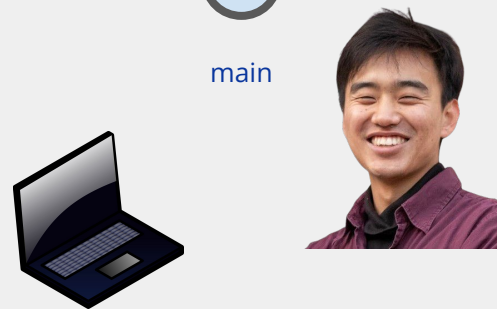
tinder

main

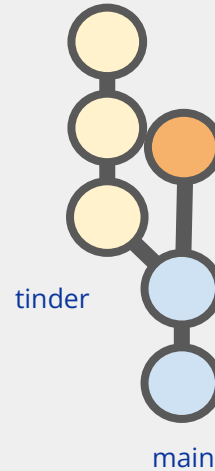


HEAD

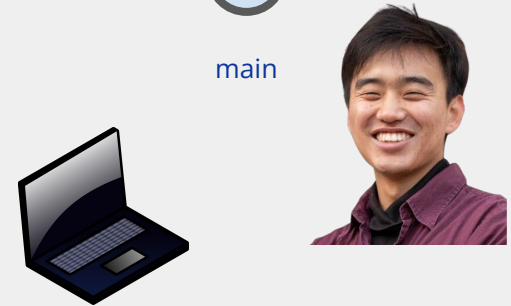
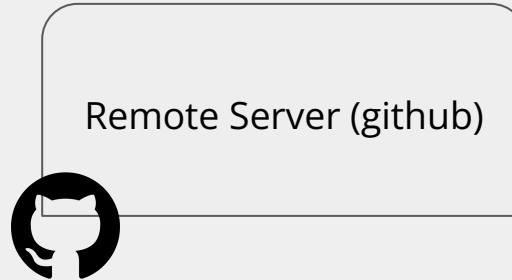
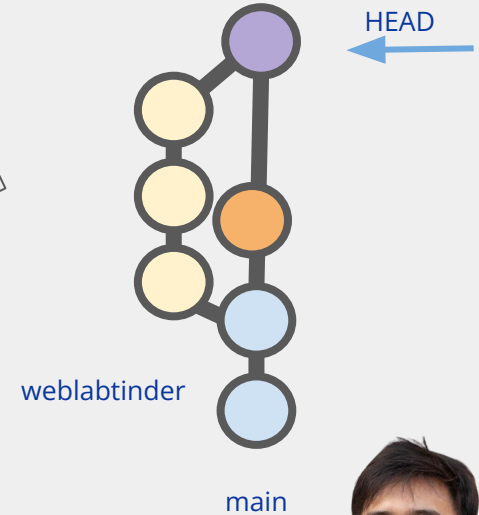
main



```
git checkout main  
git merge weblabtinder  
git push  
*on local main* git pull
```



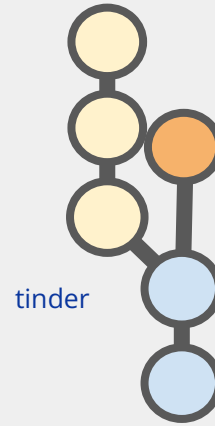
git pull



```
git checkout main  
git merge weblabtinder  
git push  
*on local main* git pull
```

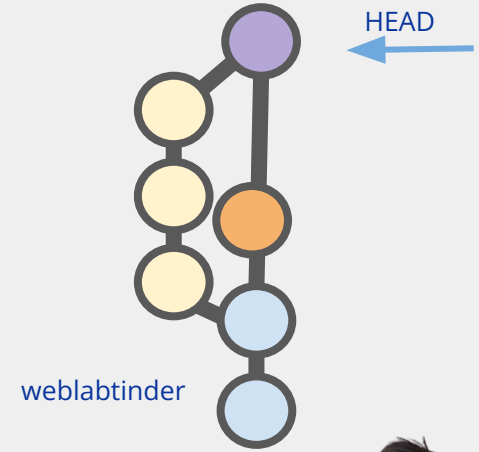


main



main

Remote Server (github)



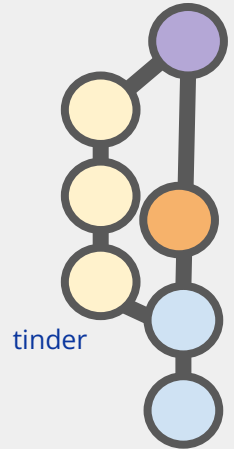
main



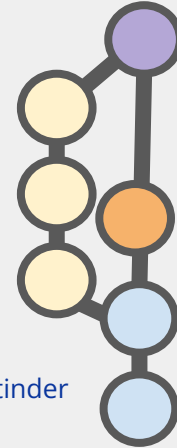
```
git checkout main
git merge weblabtinder
git push
*on local main* git pull
git push
```



main



main



main

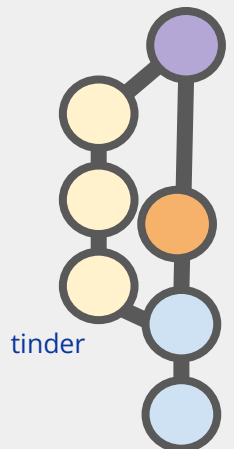
Remote Server (github)



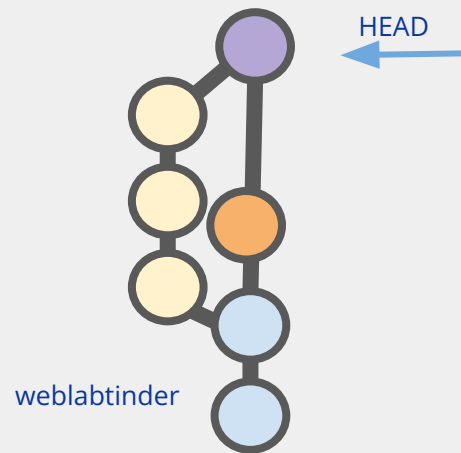
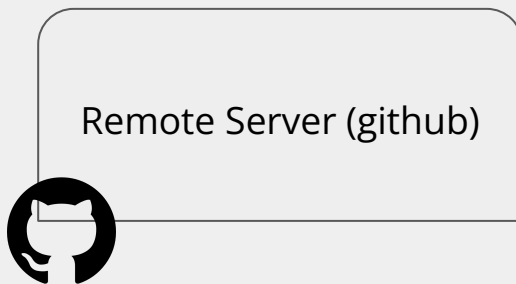

```
git checkout main
git merge weblabtinder
git push
*on local main* git pull
git push
```



main



main



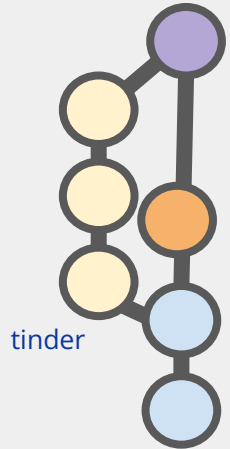
main



```
git checkout main
git merge weblabtinder
git push
*on local main* git pull
git push
git branch -d weblabtinder
```

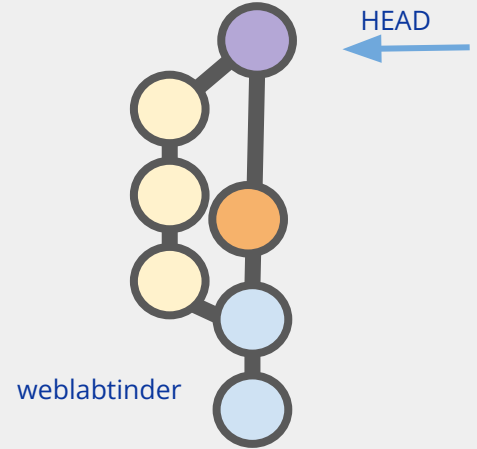


main



main

Remote Server (github)



main

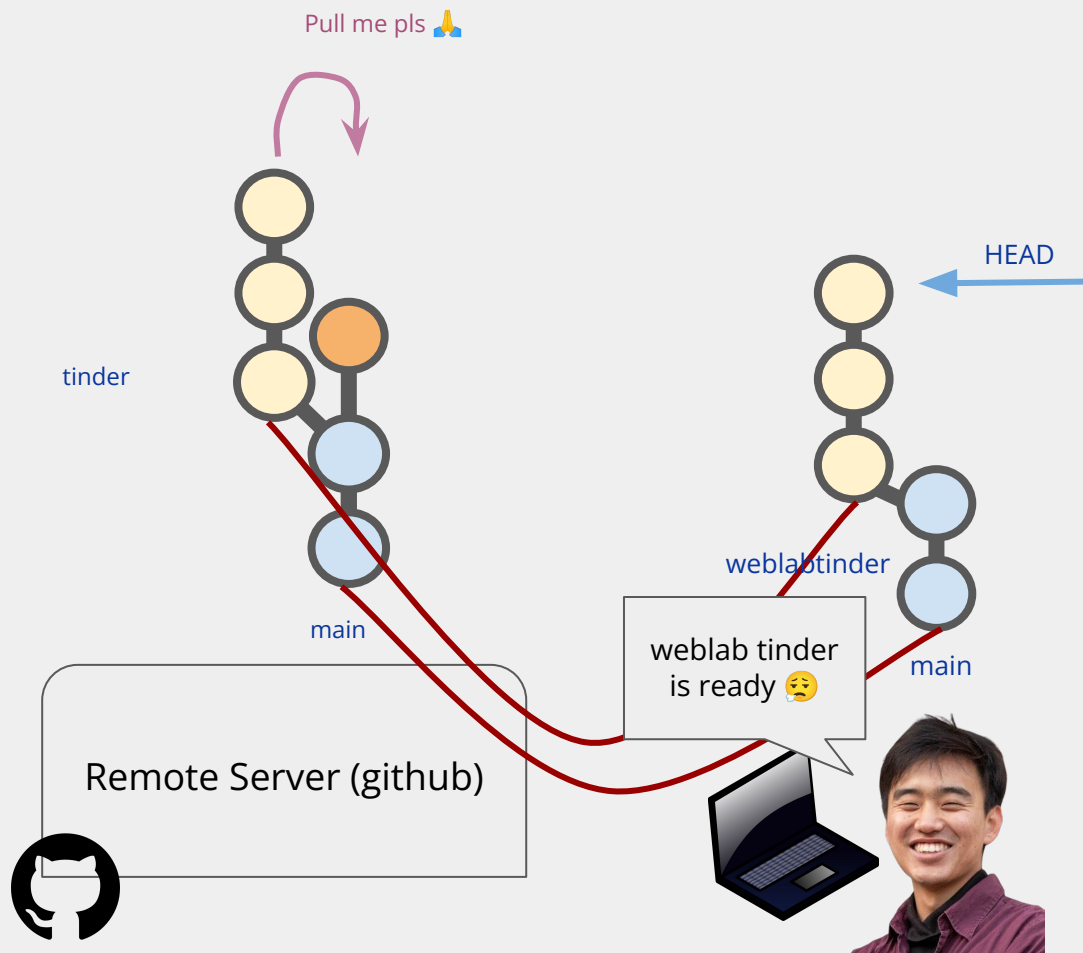


Workflow with Local + Remote Branch

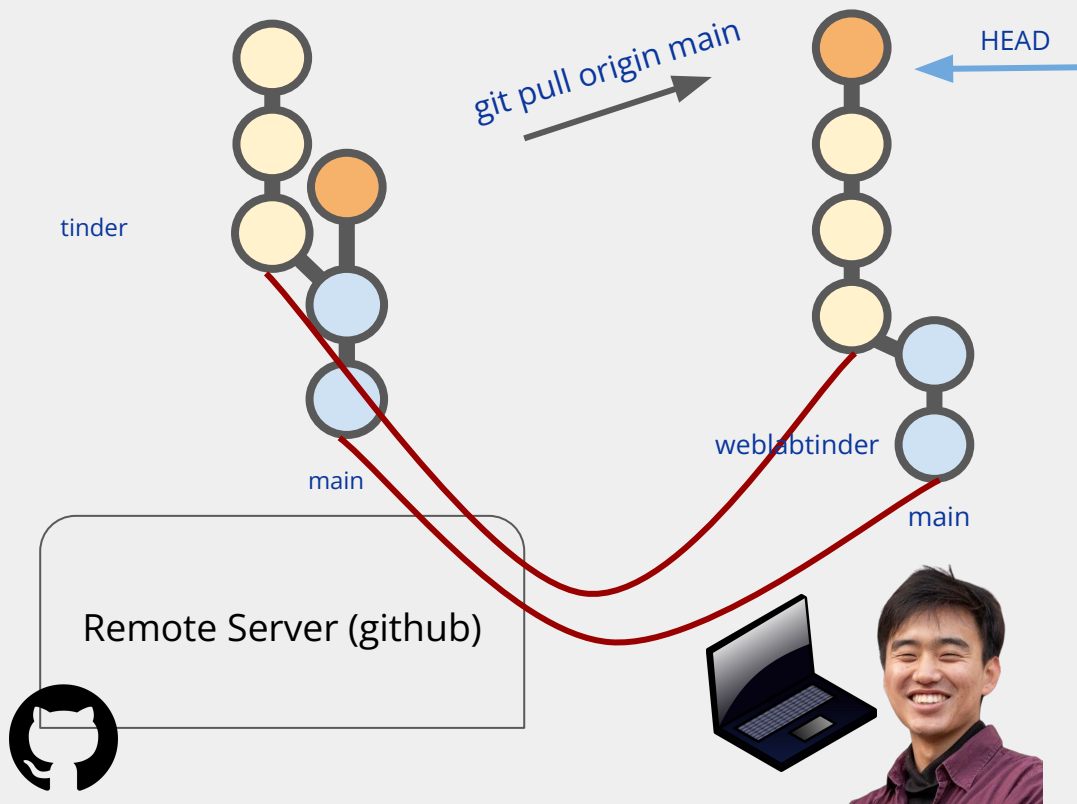
To add a new feature to your weblab project...

1. Create a local branch: `git checkout -b <branchname>`
2. Do some work, make some commits on your local branch
3. Create a remote branch that corresponds to your local branch: `git push -set-upstream origin <what you want to name remote branch>`
4. Continue doing work + making commits on your local branch, using `git push` to update the corresponding remote branch
5. When you're ready to merge your branch into main, checkout to local main and `git pull` to pull in any new commits from main (and resolve merge conflicts if necessary)
6. Still on local main, `git push` your new merge commit to remote main

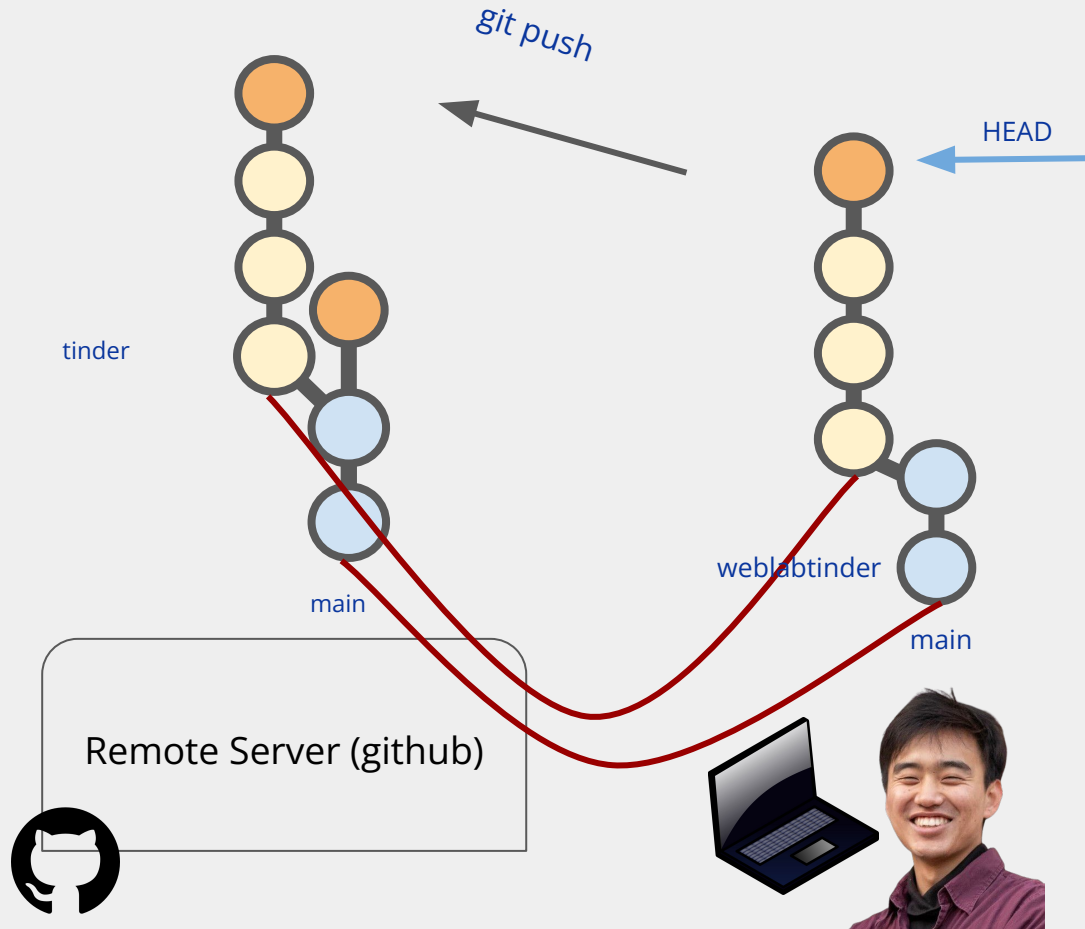
*Pull Requests



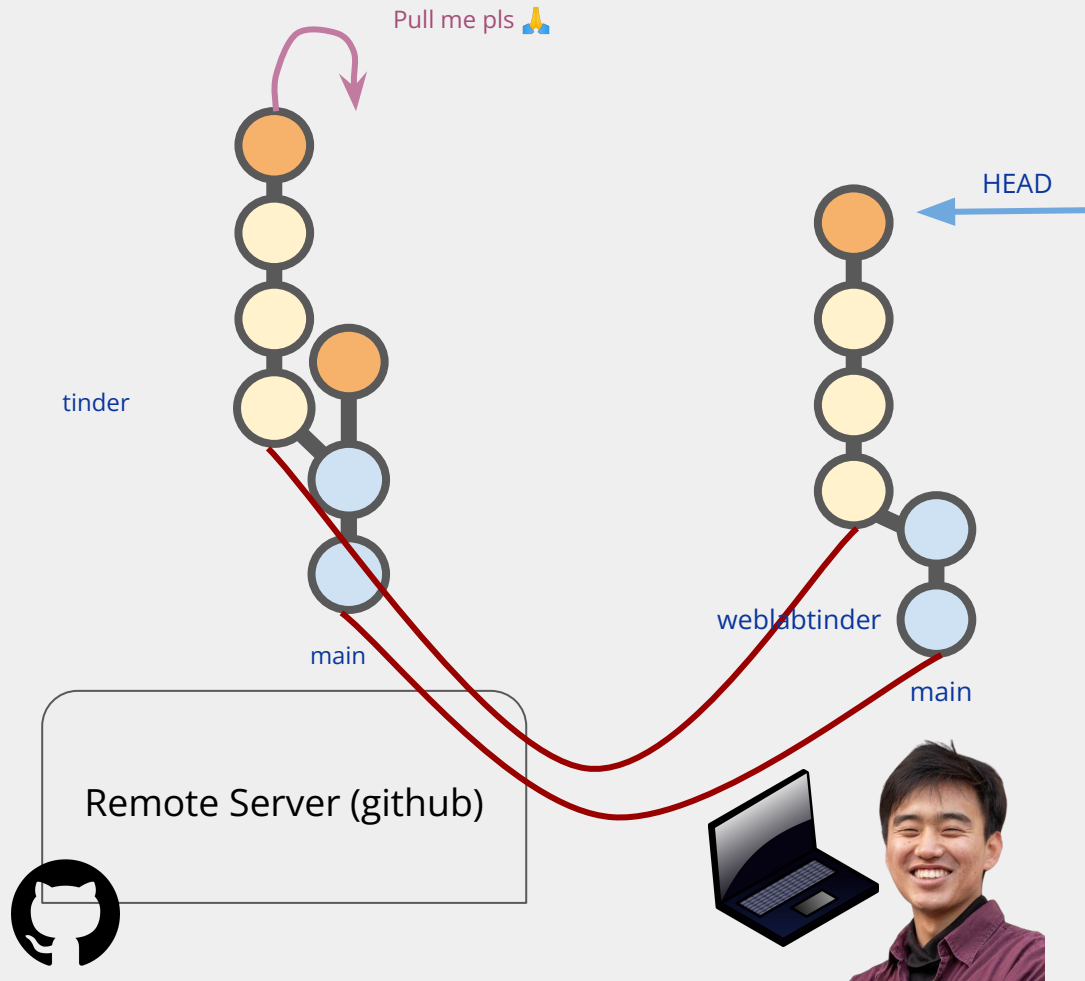
git pull origin main



```
git pull origin main  
git push
```



```
git pull origin main  
git push  
*go online and make a PR*
```

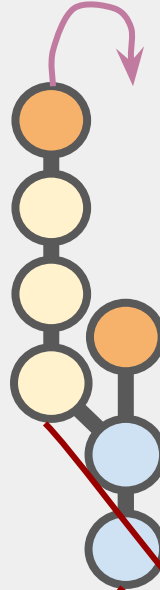


git pull origin main
git push
*go online and pull



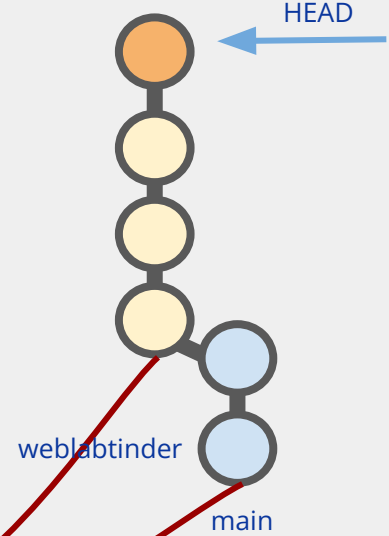
tinder

Pull me pls 🙏



main

Remote Server (github)

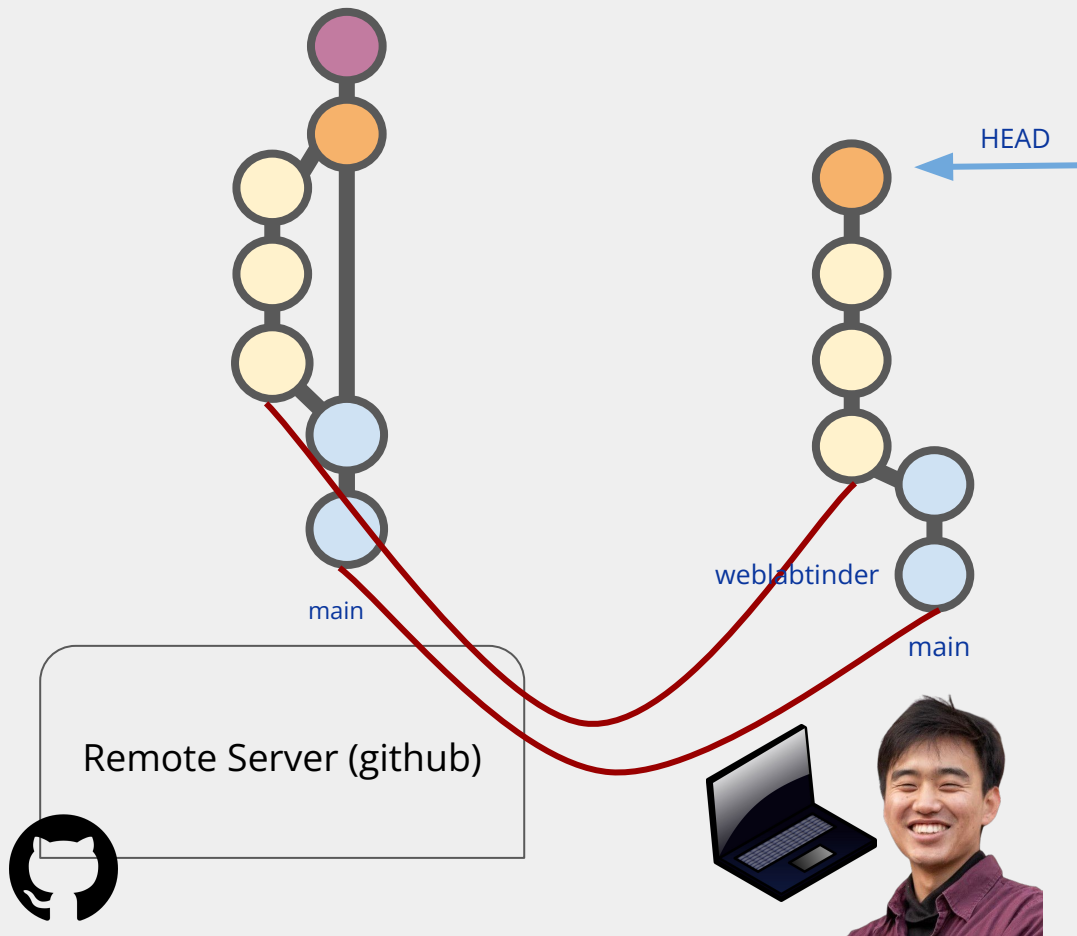


weblabtinder

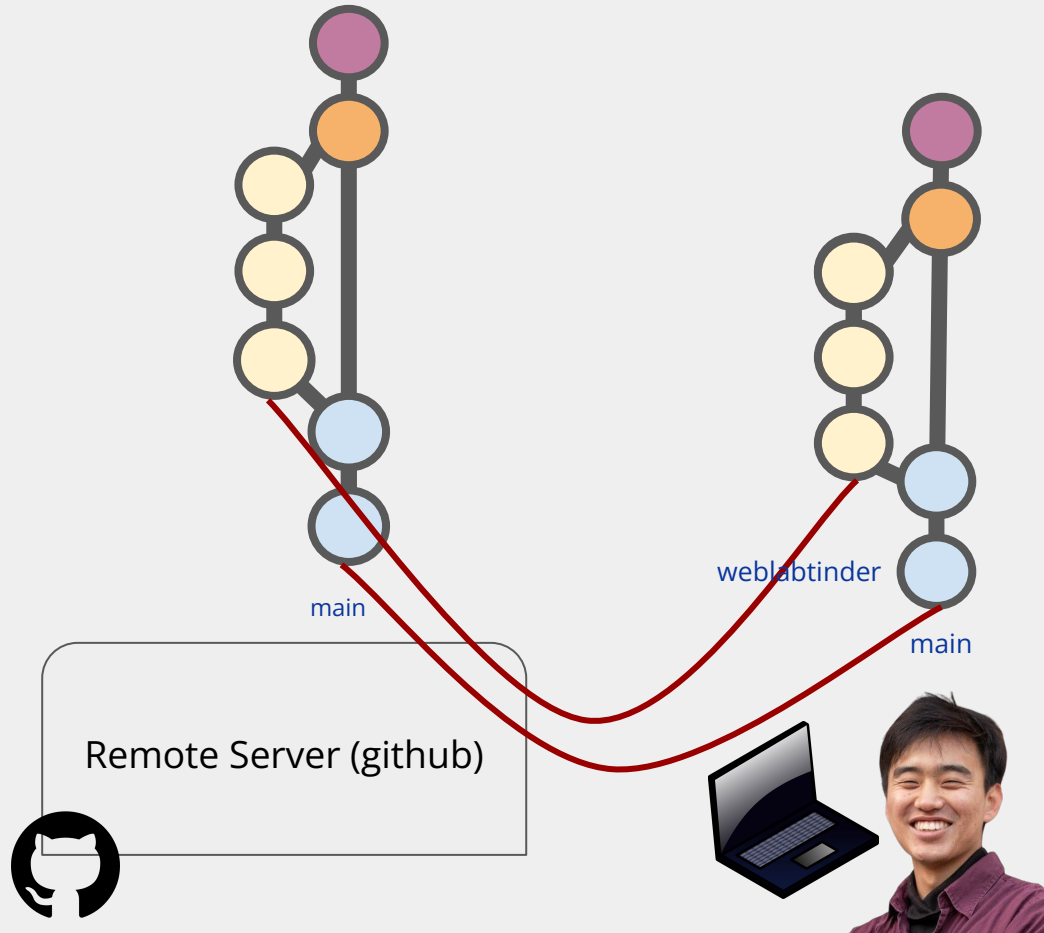
main



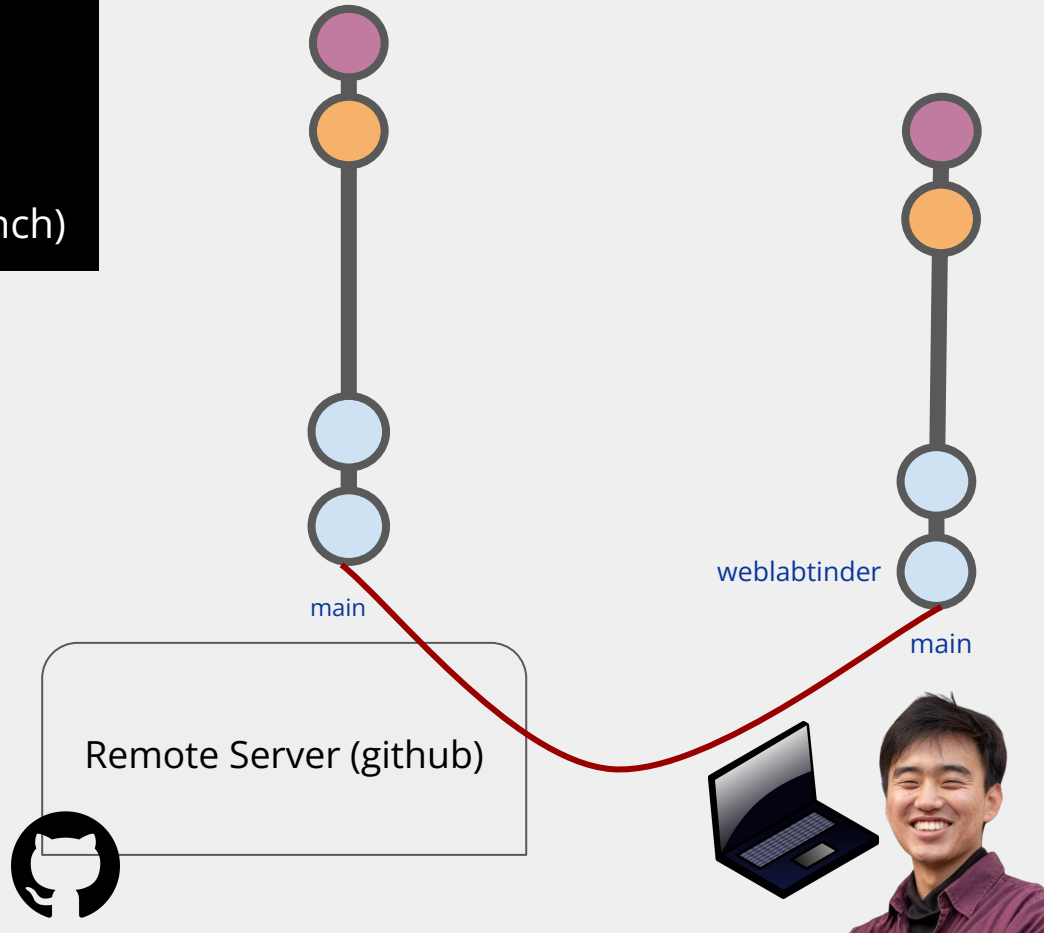
```
git pull origin main  
git push  
*go online and make a PR*
```



```
git pull origin main  
git push  
*go online and make a PR*  
*on local main branch* git pull
```



```
git pull origin main  
git push  
*go online and make a PR*  
*on local main branch* git pull  
git branch -d weblabtinder  
*go online and delete tinder branch)
```

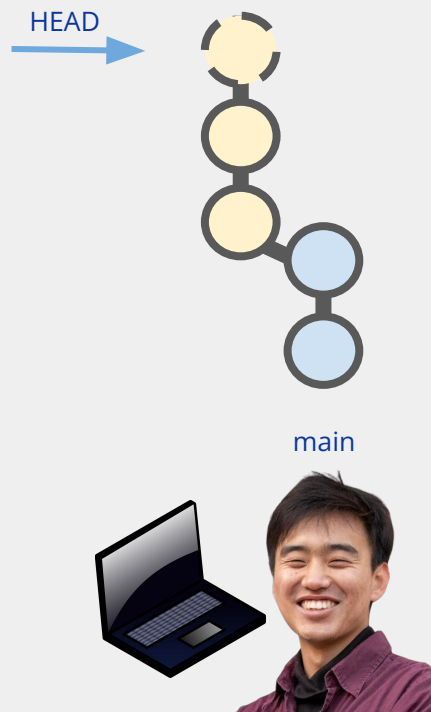
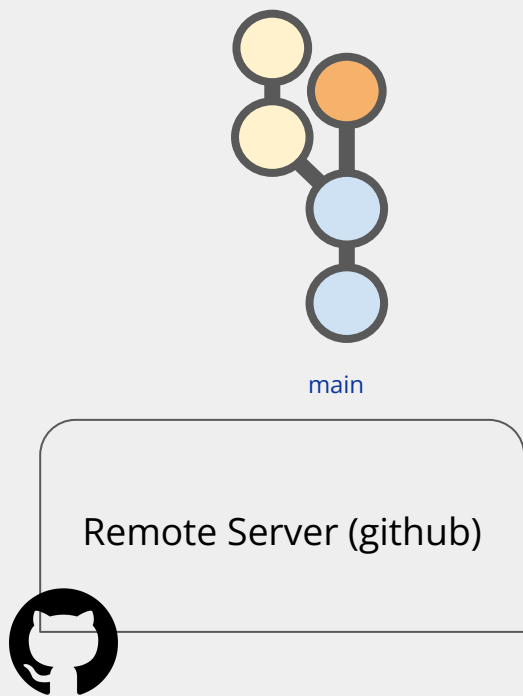
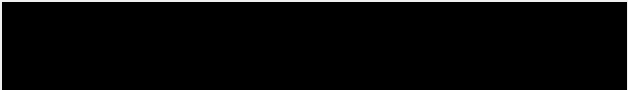


Workflow In Industry

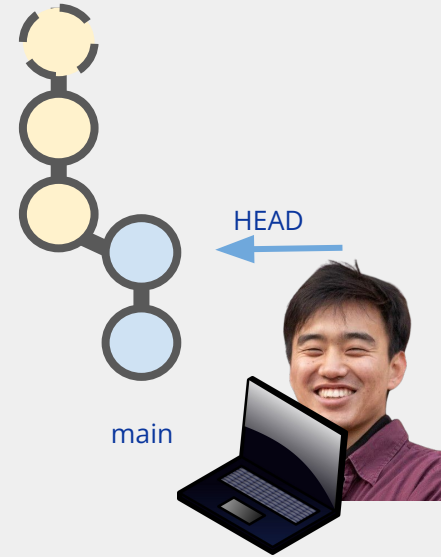
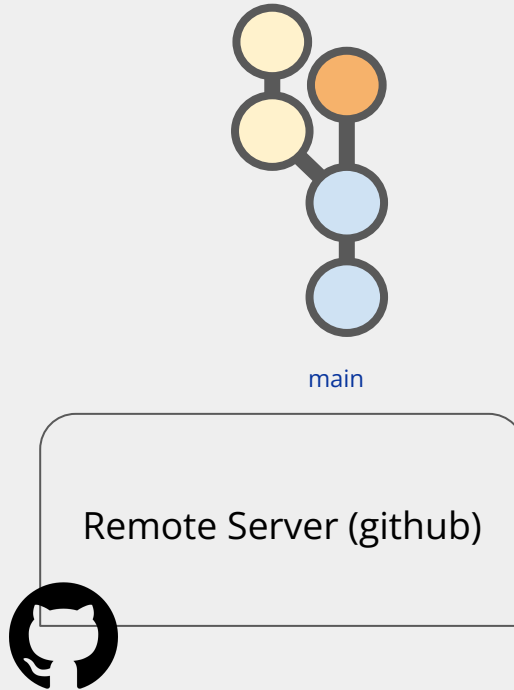
To add a new feature in industry...

1. Create a local branch: `git checkout -b <branchname>`
2. Make some commits on your local branch
3. Create a remote branch that corresponds to your local branch: `git push -set-upstream origin <what you want to name remote branch>`
4. Continue doing work + making commits on your local branch, using `git push` to update the corresponding remote branch
5. If needed, pull commits from remote main into your branch with `git pull origin main` (on your local branch) and then `git push` the pulled commits to your remote branch
6. Create a Pull Request online

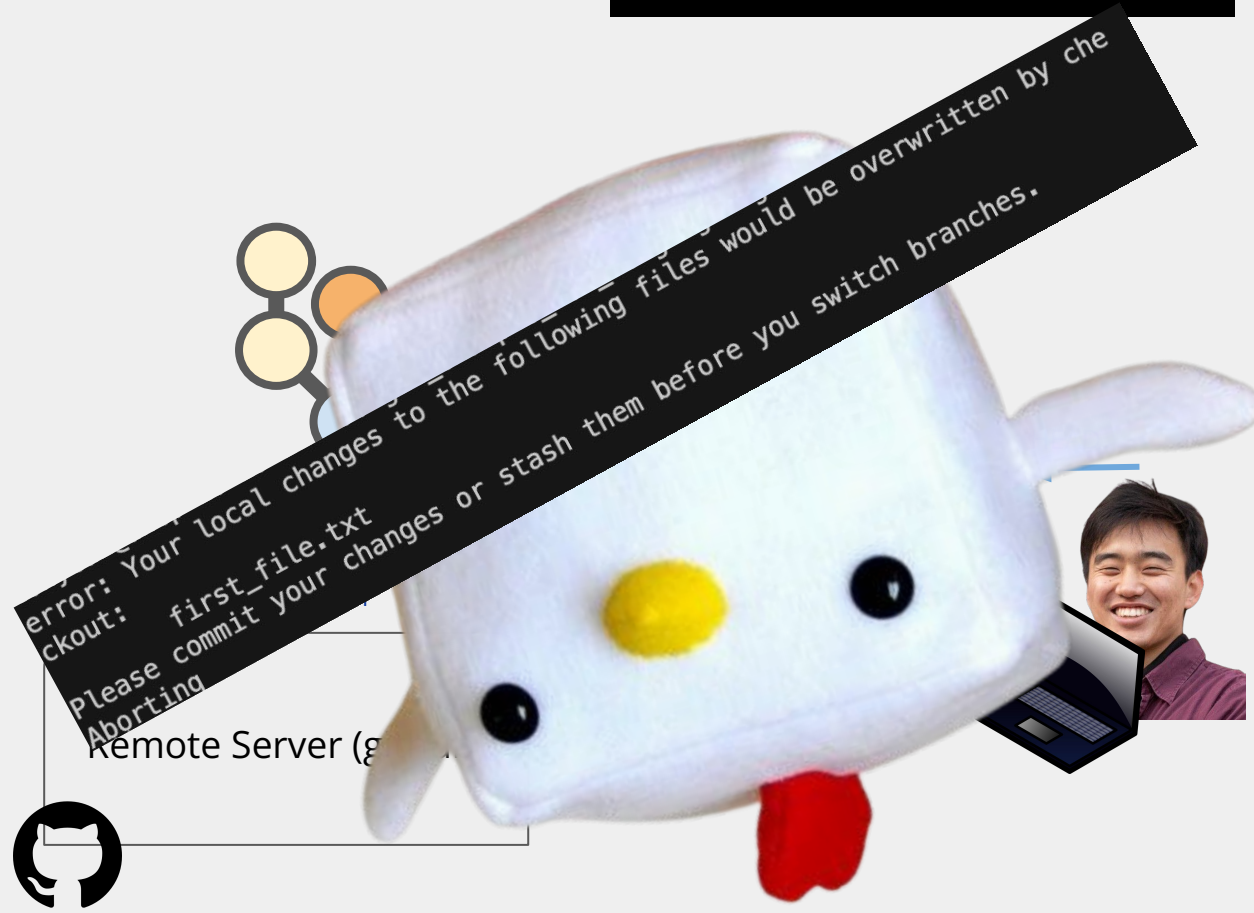
Useful tool: stash!



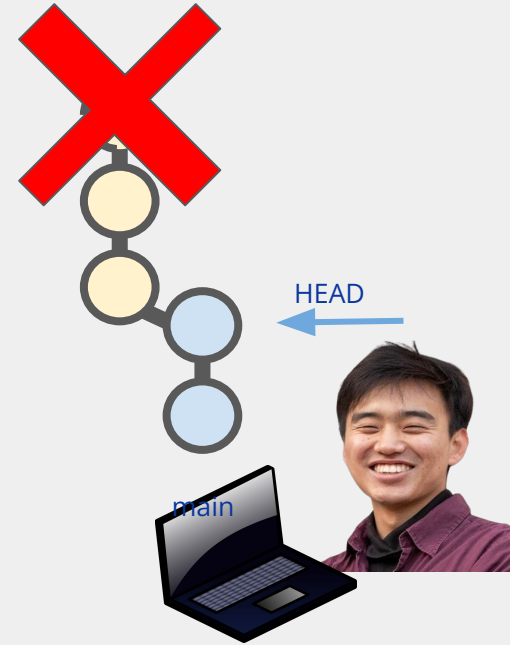
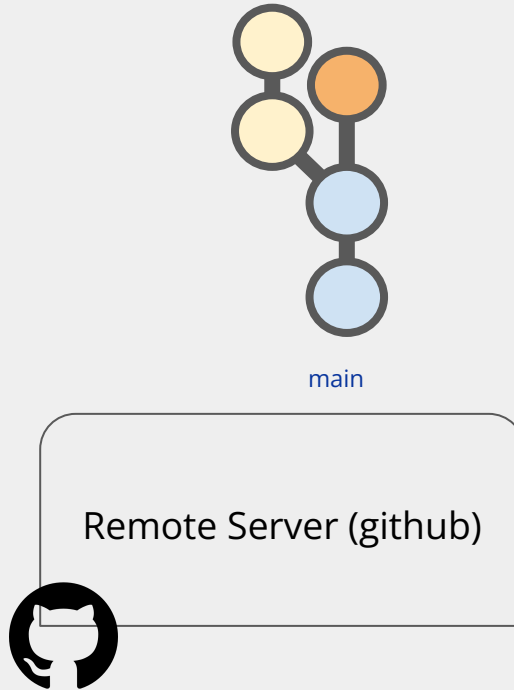
git checkout main



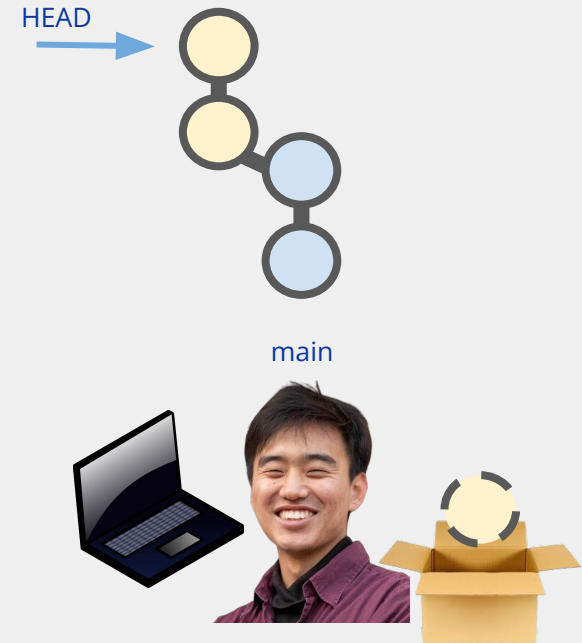
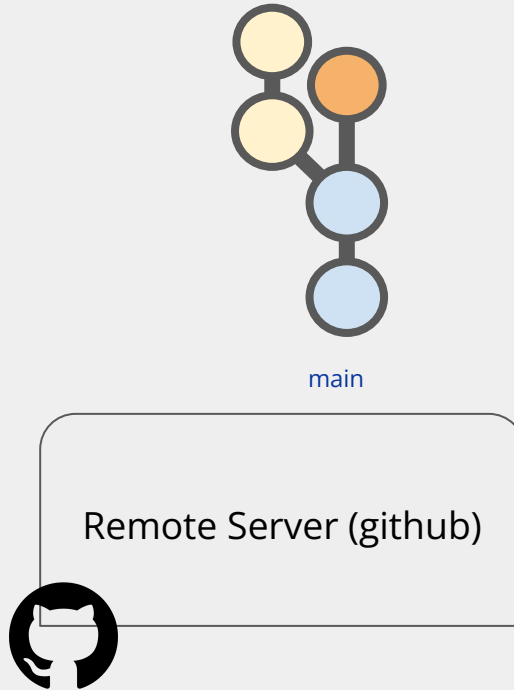
git checkout main



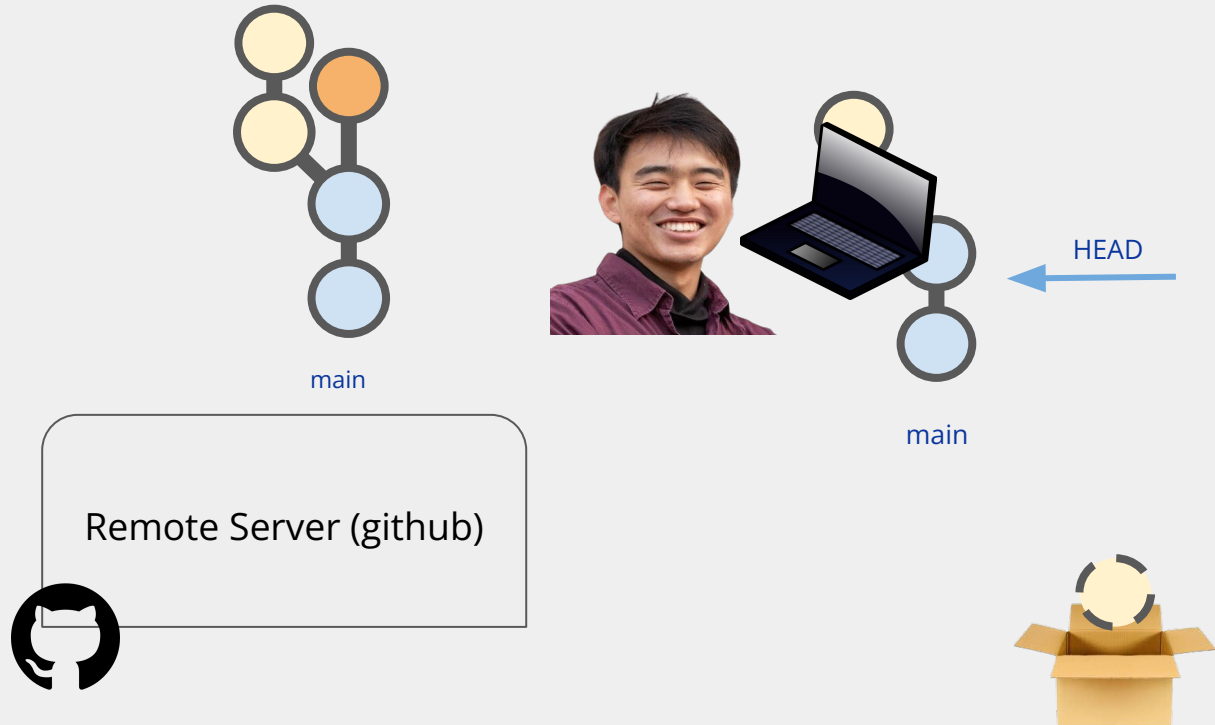
git checkout



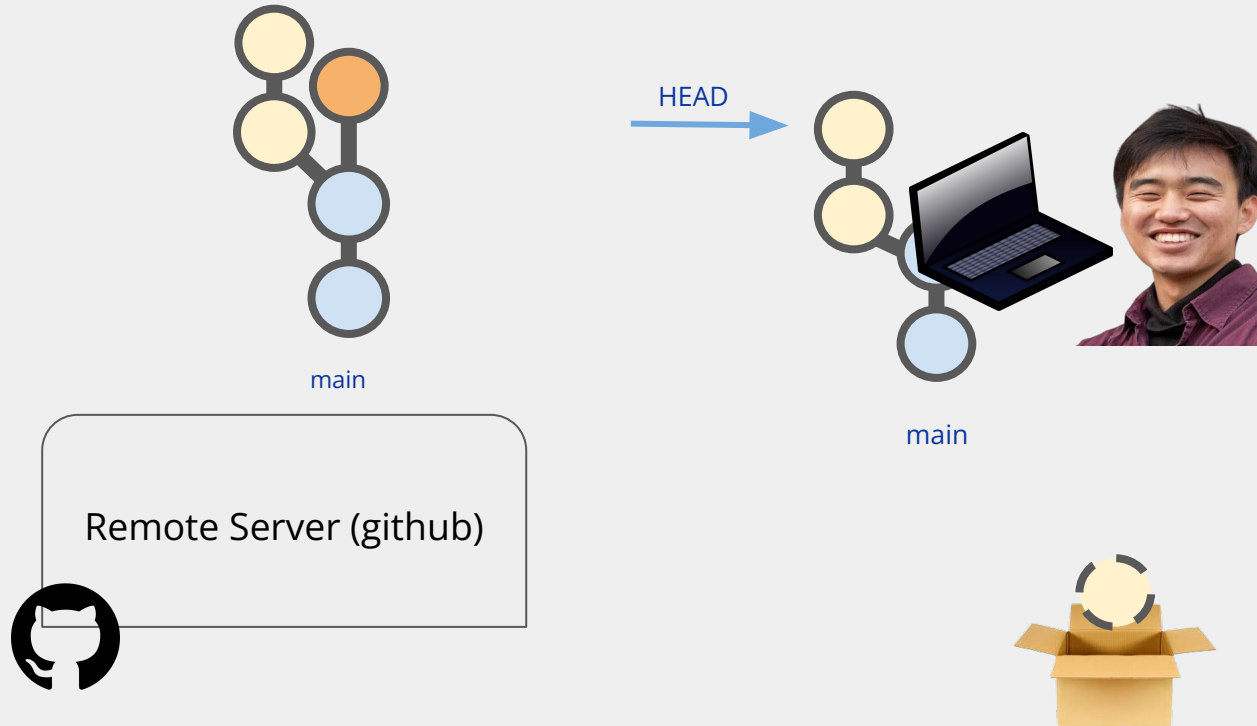
git stash



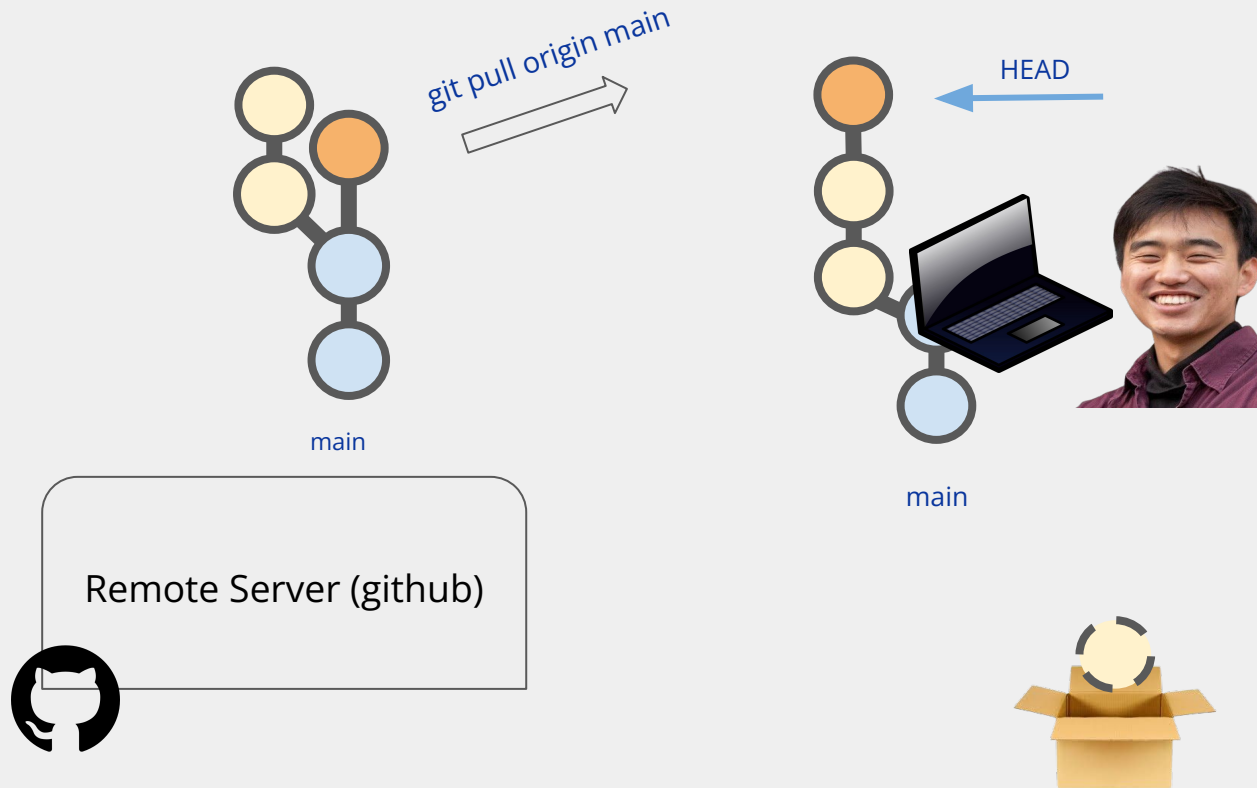
git checkout main



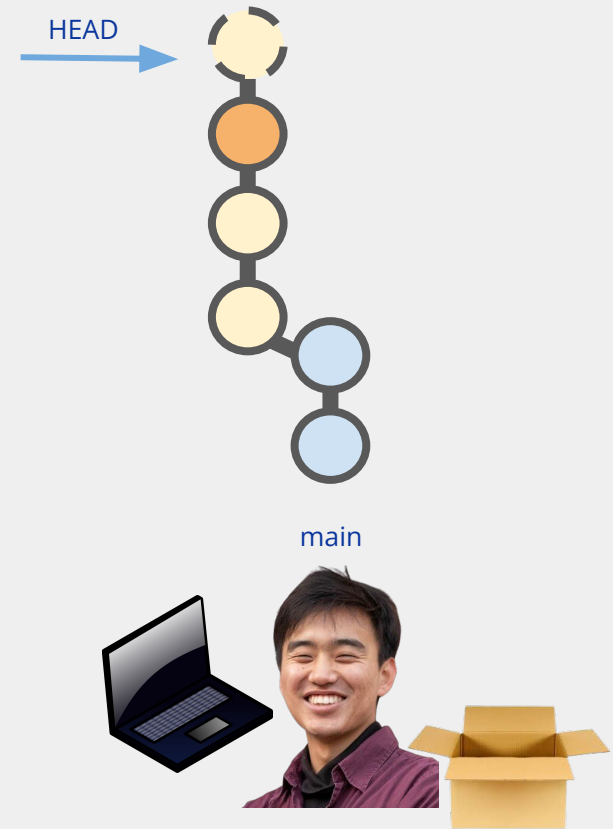
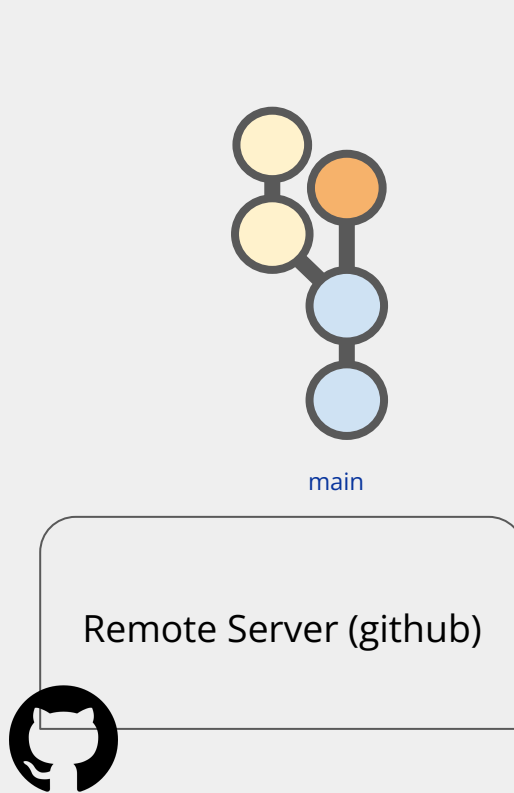
git checkout otherbranch



```
git pull origin main
```



```
git stash pop
```



Workflow with stash

1. `git stash`
2. Do anything (checkout with other branches, `git pull`)
3. `git stash pop`

Recap

Terminology

- Staging : Indicating you want to commit a set of changes in a file.
- Commit : a group of code changes that correspond to a certain task
- Branch : making a commit history diverge, often to group commits together for developing features
- Remote/Origin : The version of your repo that is on the internet (Github)
- Upstream : The corresponding remote branch for a local branch (aka tracking)

General Workflows





- Create new local branch
- Set upstream (Create corresponding remote branch)
- Create commits, pull & push often
- Merge into main (either directly or Pull Request)

Git resources

Good Git practice: [Learn Git Branching](#)

Git Cheat sheet: [Cheatsheet](#)

Announcements

- Please give us feedback!   weblab.is/feedback
 - Thank you all for the really helpful feedback both here and on milk and cookies!
- Informal Office Hours/Recitation now till 5pm 
- **Milestone 1 signups due tonight!**
- Workshop 3 re-record coming soon :))
-  Project skeleton at weblab.is/skeleton
 - Includes some stuff we haven't learned yet (google auth)