UNIVERSITY of WASHINGTON

CSE 583

Version Control II

David Beck[1,2,3], Joseph Hellerstein[1,3], **Bernease Herman[1]**, Colin Lockard[3]

[1]eScience Institute
[2]Chemical Engineering
[3]Computer Science

November 1, 2018

Agenda

1. Questions on homework 4 or previous lectures
2. Version Control, Git, and GitHub
3. Hands on practice with Git & GitHub

Quick review from October 4th version control lecture

0. Set up Git and project

```
> git config [options]
> git config --global color.ui "auto"
> git init
```

Explore

```
> git status
> git log [options]
> git show [sha1]
```

1. Make changes

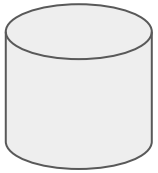
Use your preferred editor and tools.

2. Stage files and changes

```
> git add [path]
> git add -A
> git add .
```

3. Create snapshot

```
> git commit
> git commit
    -m "[message]"
```

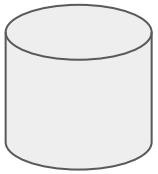


Local repo

4. Add remote repo

```
> git remote add
     [name] [url]
```

Remote repo



5. Pull from remote

```
> git fetch [remote] [branch]
> git pull [remote] [branch]
```

6. Push to remote

```
> git commit
> git commit -m "[message]"
```

Time to remember your GitHub logins

Introduction To Git
#(and some GitHub)

0. Set up

> git config [options]
> git init
> .gitignore

Introduction To Git
#(and some GitHub)

0. Set up

> git config [options]
> git init
> .gitignore

1. Make Changes

(use your preferred editor and tools.)

Introduction To Git
#(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

Introduction To Git
#(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
  -m "[msg]
```

Introduction To Git
#(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
  -m "[msg]"
```

Introduction To Git
*(and some GitHub)

0. Set up
> git config [options]
> git init
> .gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
> git add
> git add -A
> git rm [path]

3. Create snapshot
> git commit
> git commit -m "[msg]"

4. Explore
> git status
> git log [options]
> git show [sha1]

(Repeat 1-4 as desired.)

Introduction To Git
*(and some GitHub)

0. Set up
>git config [options]
>git init
>.gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
>git add
>git add -A
>git rm [path]

3. Create snapshot
>git commit
>git commit -m "[msg]"

4. Explore
>git status
>git log [options]
>git show [sha1]

(Repeat 1-4 as desired.)

Introduction To Git
*(and some GitHub)

0. Set up
> git config [options]
> git init
> .gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
> git add
> git add -A
> git rm [path]

3. Create snapshot
> git commit
> git commit -m "[msg]"

4. Explore
> git status
> git log [options]
> git show [sha1]

(Repeat 1-4 as desired.)

5. Add remote
> git remote add [name][url]
> git remote -v

Introduction To Git
*(and some GitHub)

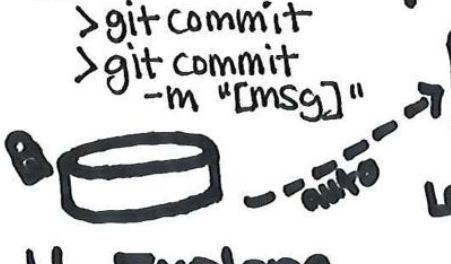
0. Set up
> git config [options]
> git init
> .gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
> git add
> git add -A
> git rm [path]

3. Create snapshot
> git commit
> git commit -m "[msg]"



5. Add remote
> git remote add [name][url]
> git remote -v

4. Explore
> git status
> git log [options]
> git show [sha1]

(Repeat 1-4 as desired.)

Introduction To Git
*(and some GitHub)

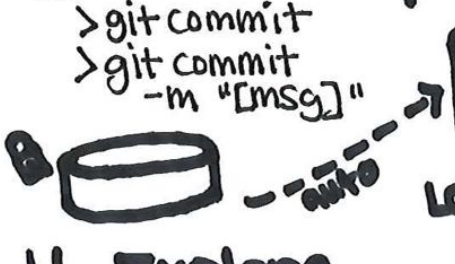
0. Set up
>git config [options]
>git init
>.gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
>git add
>git add -A
>git rm [path]

3. Create snapshot
>git commit
>git commit -m "[msg]"



5. Add remote
>git remote add [name][url]
>git remote -v

4. Explore
>git status
>git log [options]
>git show [sha1]

6. Pull from remote
>git fetch [remote][branch]
>git pull [remote][branch]

(Repeat 1-4 as desired.)

Introduction To Git
*(and some GitHub)

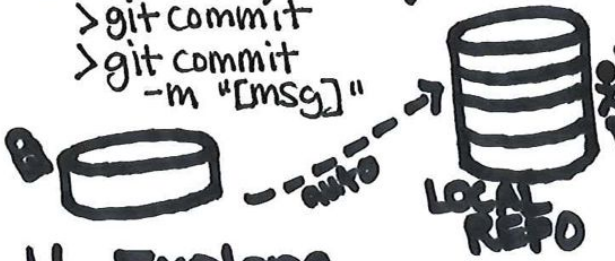
0. Set up
>git config [options]
>git init
>.gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
>git add
>git add -A
>git rm [path]

3. Create snapshot
>git commit
>git commit -m "[msg]"



5. Add remote
>git remote add [name][url]
>git remote -v

4. Explore
>git status
>git log [options]
>git show [sha1]

(Repeat 1-4 as desired.)

6. Pull from remote
>git fetch [remote][branch]
>git pull [remote][branch]

7. Push to remote
>git push [remote][branch]

Introduction To Git
*(and some GitHub)

0. Set up
> git config [options]
> git init
> .gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
> git add
> git add -A
> git rm [path]

3. Create snapshot
> git commit
> git commit -m "[msg]"

4. Explore
> git status
> git log [options]
> git show [sha1]

(Repeat 1-4 as desired.)

5. Add remote
> git remote add [name] [url]
> git remote -v

6. Pull from remote
> git fetch [remote] [branch]
> git pull [remote] [branch]

7. Push to remote
> git push [remote] [branch]

BONUS: Conflicts
TIP: pull before commit to minimize conflicts!
> git merge

Done with review, on to new material

Introduction To Git
*(and some GitHub)

0. Set up
> git config [options]
> git init
> .gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
> git add
> git add -A
> git rm [path]

3. Create snapshot
> git commit
> git commit -m "[msg]"

4. Explore
> git status
> git log [options]
> git show [sha1]

(Repeat 1-4 as desired.)

5. Add remote
> git remote add [name] [url]
> git remote -v

6. Pull from remote
> git fetch [remote] [branch]
> git pull [remote] [branch]

7. Push to remote
> git push [remote] [branch]

BONUS: Conflicts
TIP: pull before commit to minimize conflicts!
> git merge

12. Forks and PRs


(Done on GitHub website.)

Introduction To Git
*(and some GitHub)

0. Set up
>git config [options]
>git init
>.gitignore

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files
>git add
>git add -A
>git rm [path]

3. Create snapshot
>git commit
>git commit -m "[msg]"

4. Explore
>git status
>git log [options]
>git show [sha1]

(Repeat 1-4 as desired.)

5. Add remote
>git remote add [name][url]
>git remote -v

6. Pull from remote
>git fetch [remote][branch]
>git pull [remote][branch]

7. Push to remote
>git push [remote][branch]

8. Undoing changes
>git reset [options]
>git revert [sha1]

9. Rewriting history
(Not to be used on public commits!)
>git commit --amend
>git rebase [-i]
>git reflog

TIP: pull before commit to minimize conflicts!

10. Climbing the Git tree

>git checkout
Detached HEAD state!

BONUS: Conflicts
>git merge
>git rebase

11. Branches
>git branch [options]
>git checkout

>git merge

12. Forks and PRs
(Done on GitHub website.)

13. Workflows and Tags and More
>git tag [options]

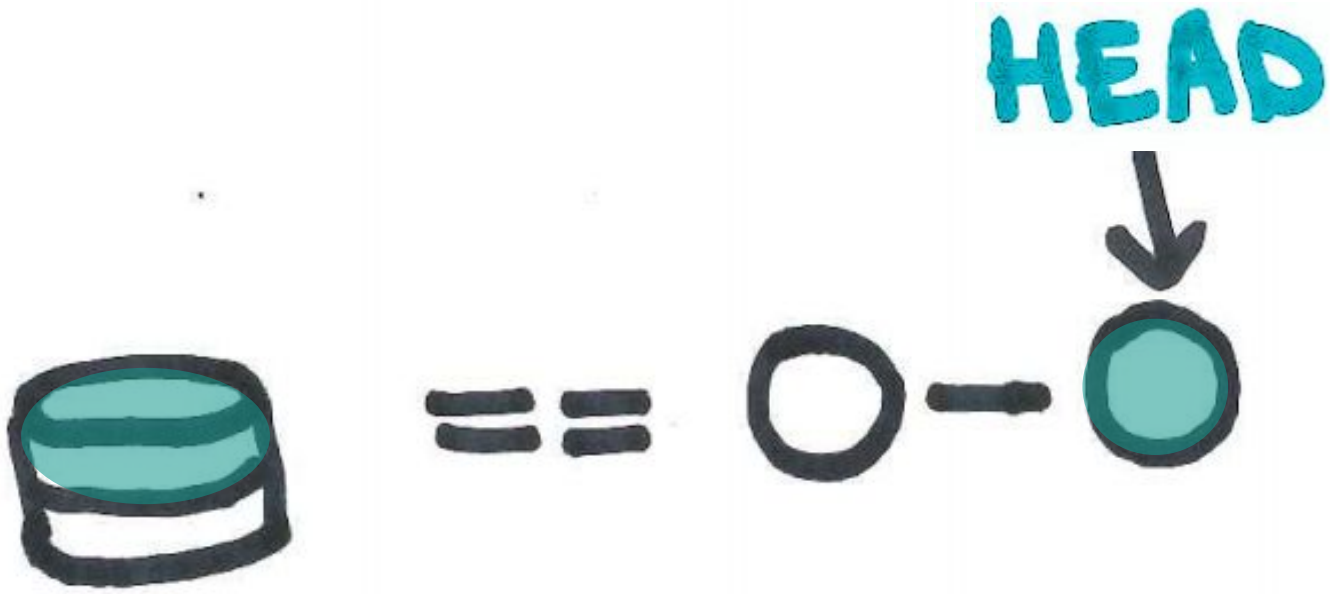
Bernease Herman 10/4/18

A single commit

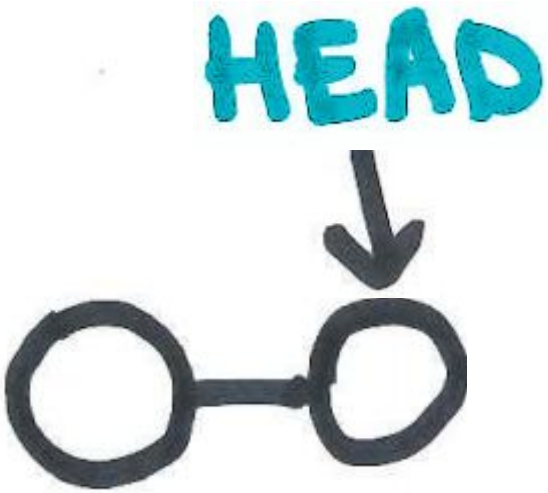
In tree representation

Multiple commits represented

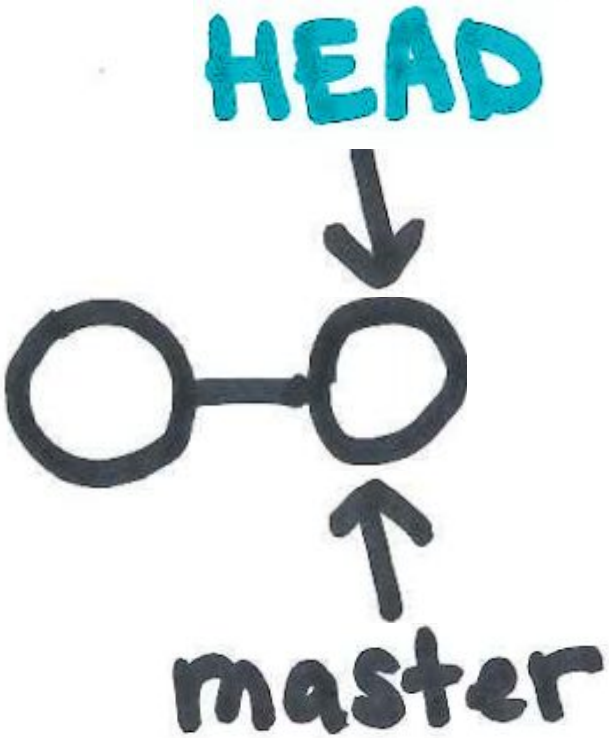
Your working directory and files



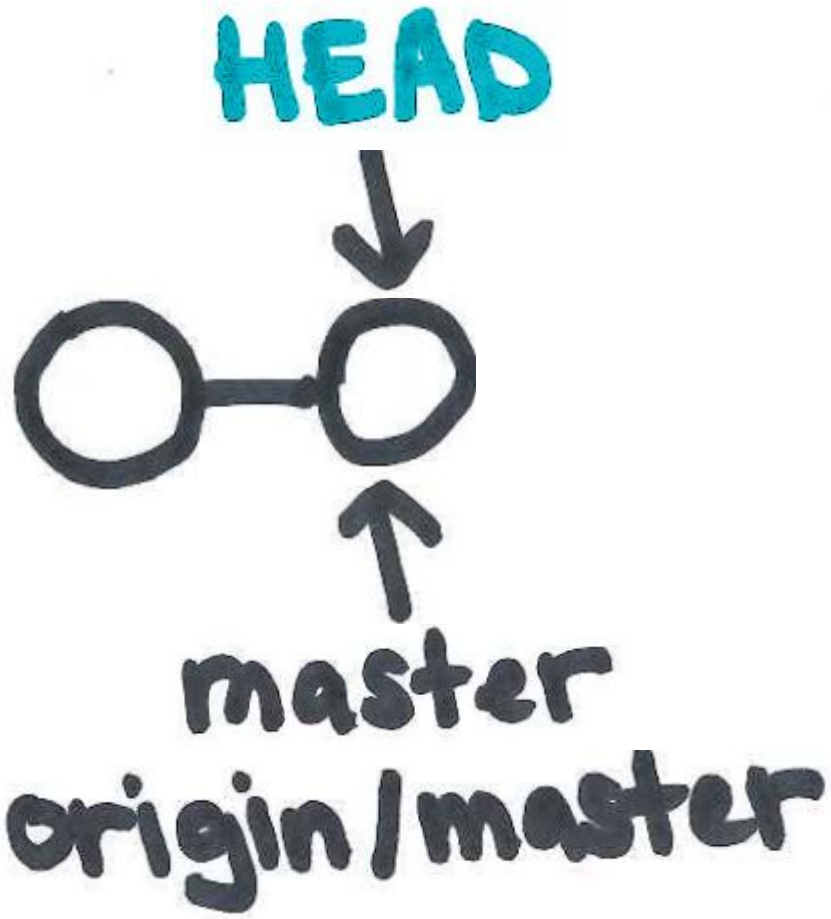
HEAD pointer on our tree



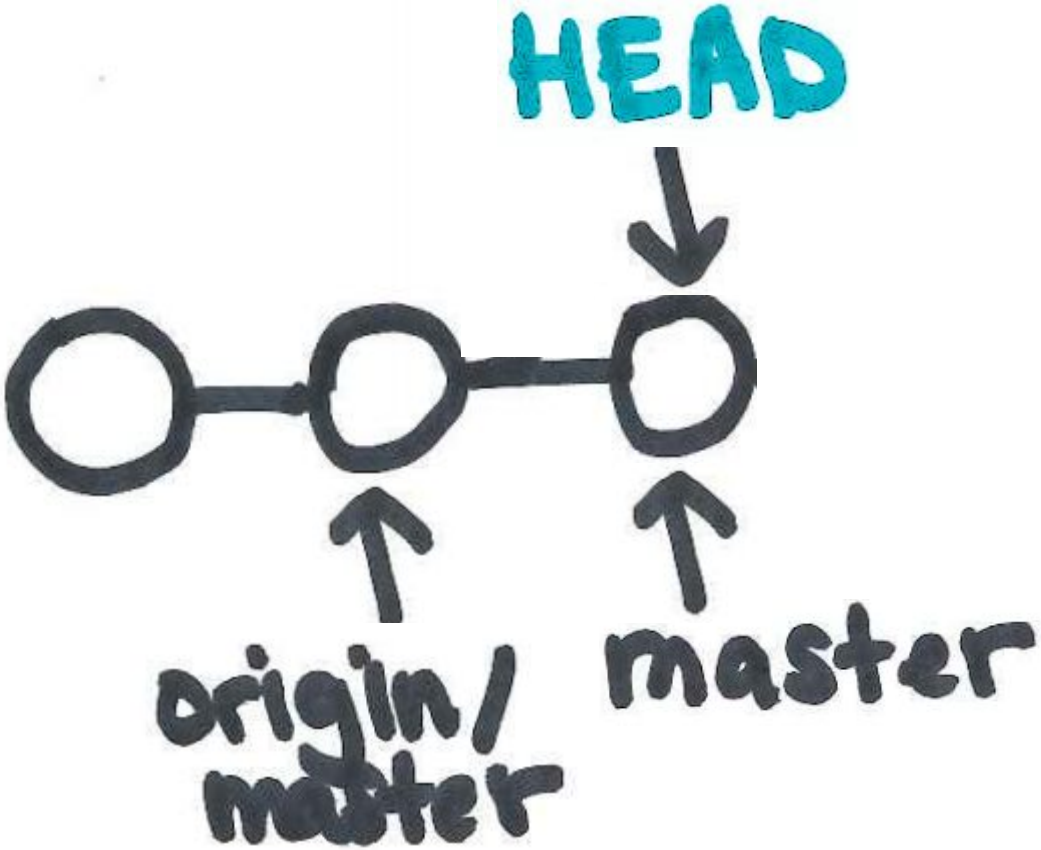
Think of branches as a pointer, as well



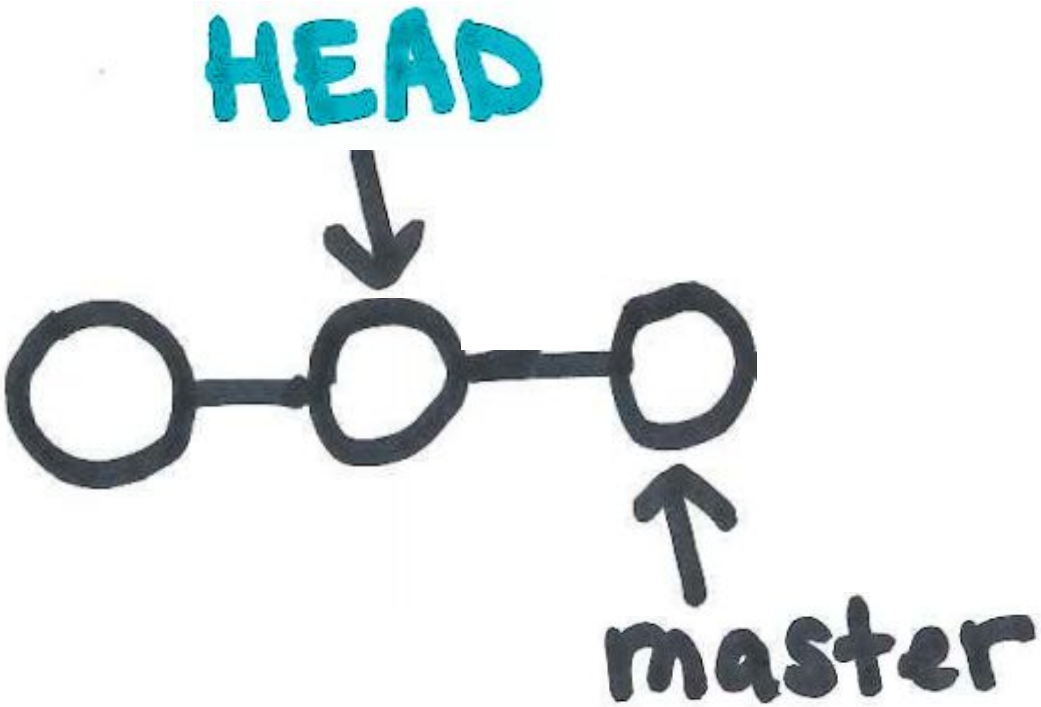
Remote branches are included



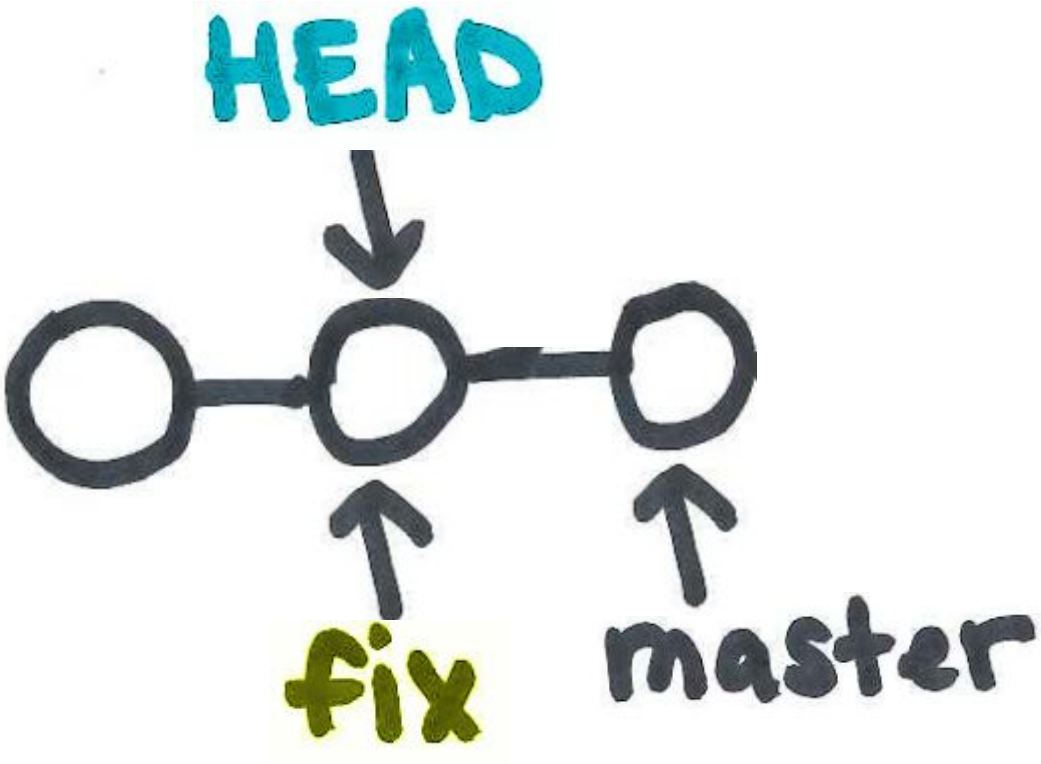
Local commit, before pushing to remote



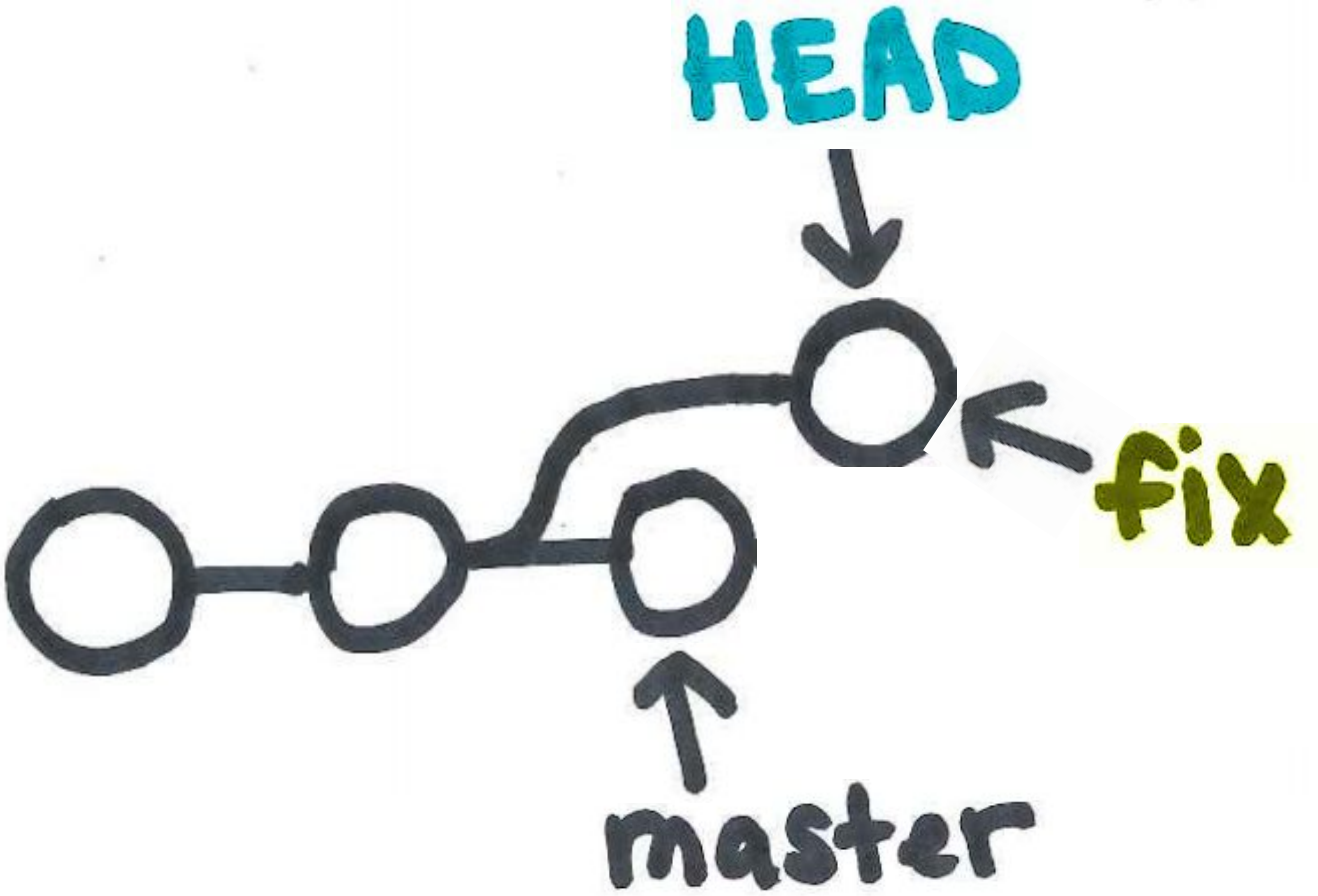
Checkout an earlier commit (hiding `origin/master` for simplicity)



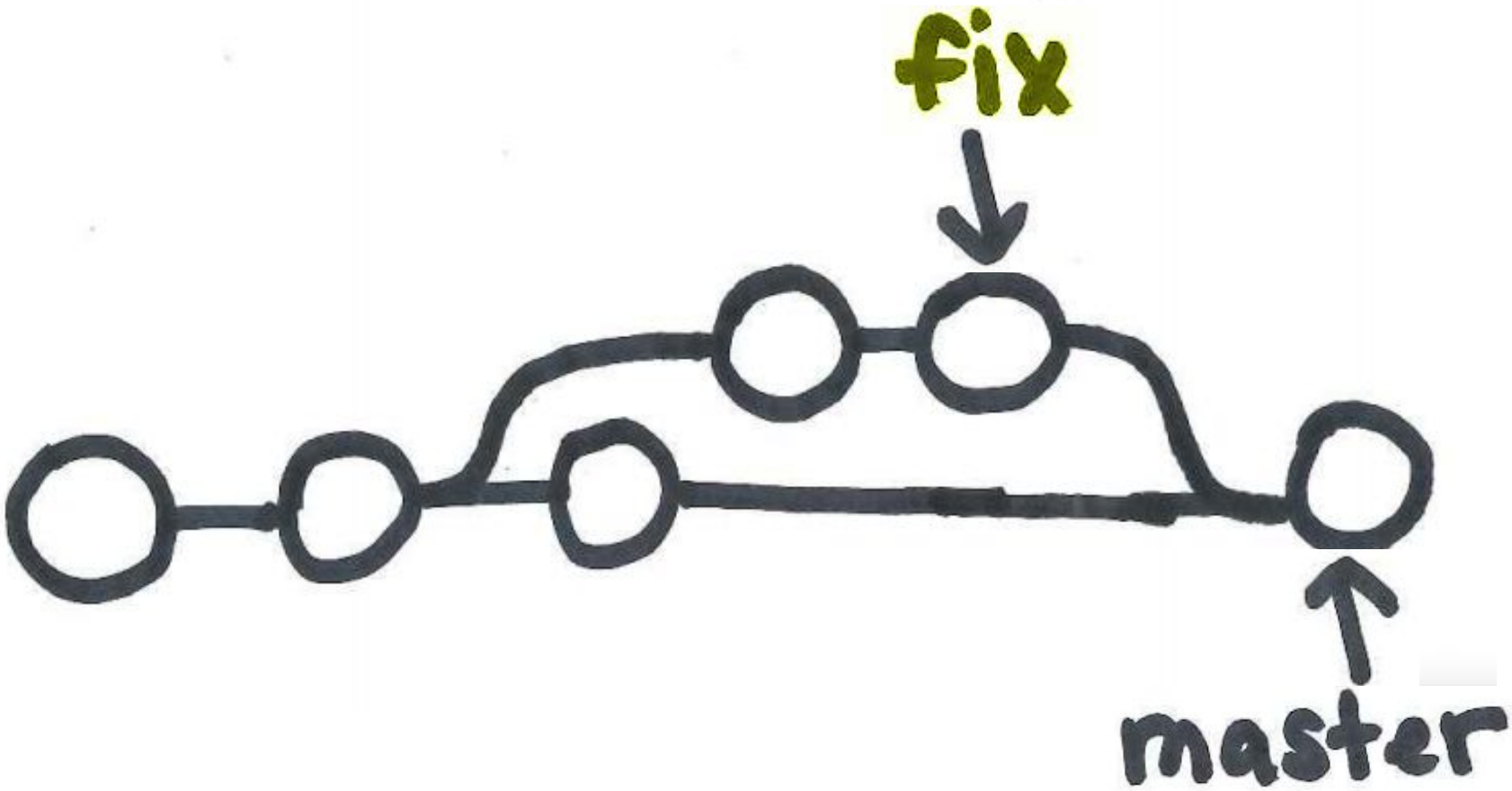
Creating a new branch

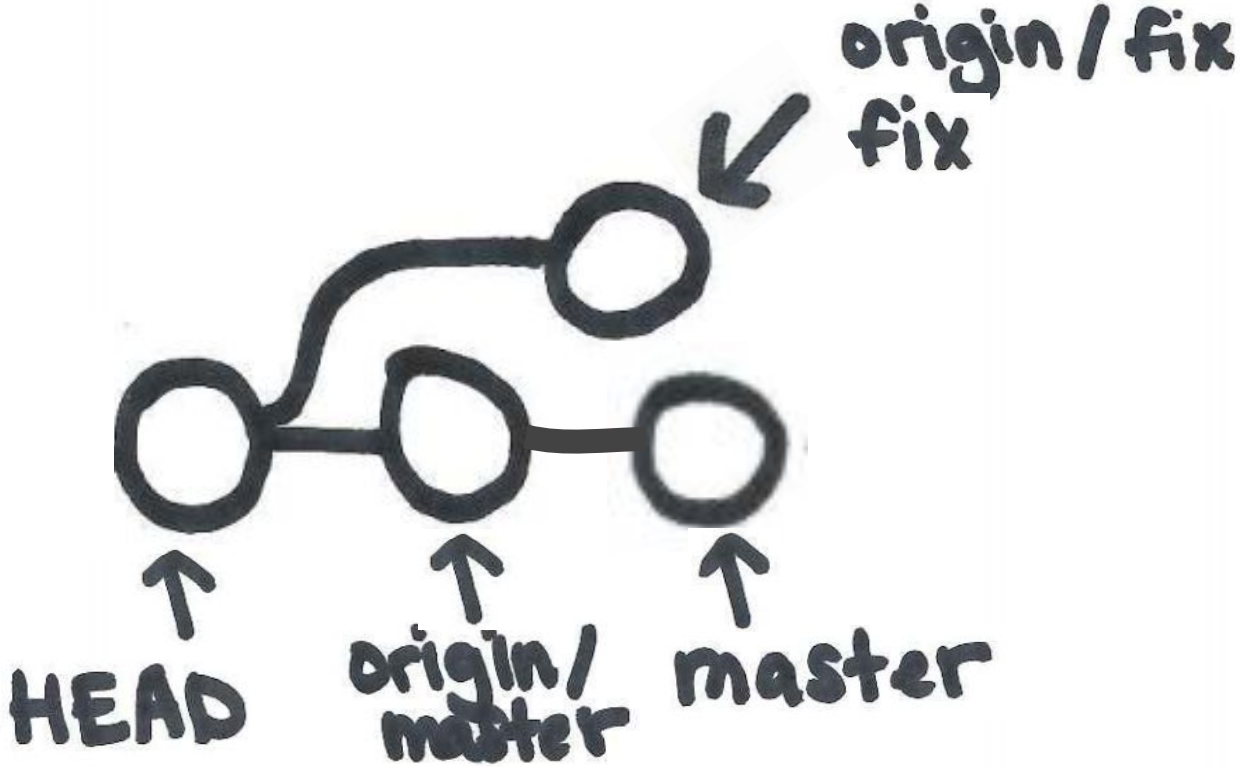


Making changes along this branch



Merging commits to another branch




origin / fix
fix
HEAD
origin/
master
master

Exercise: Tracing the Git Tree

With a partner (or groups of 3), walk through how the following commands would change your git tree. Draw a diagram with the final tree that includes labels for HEAD, all local branches, and all remote branches (origin/*).

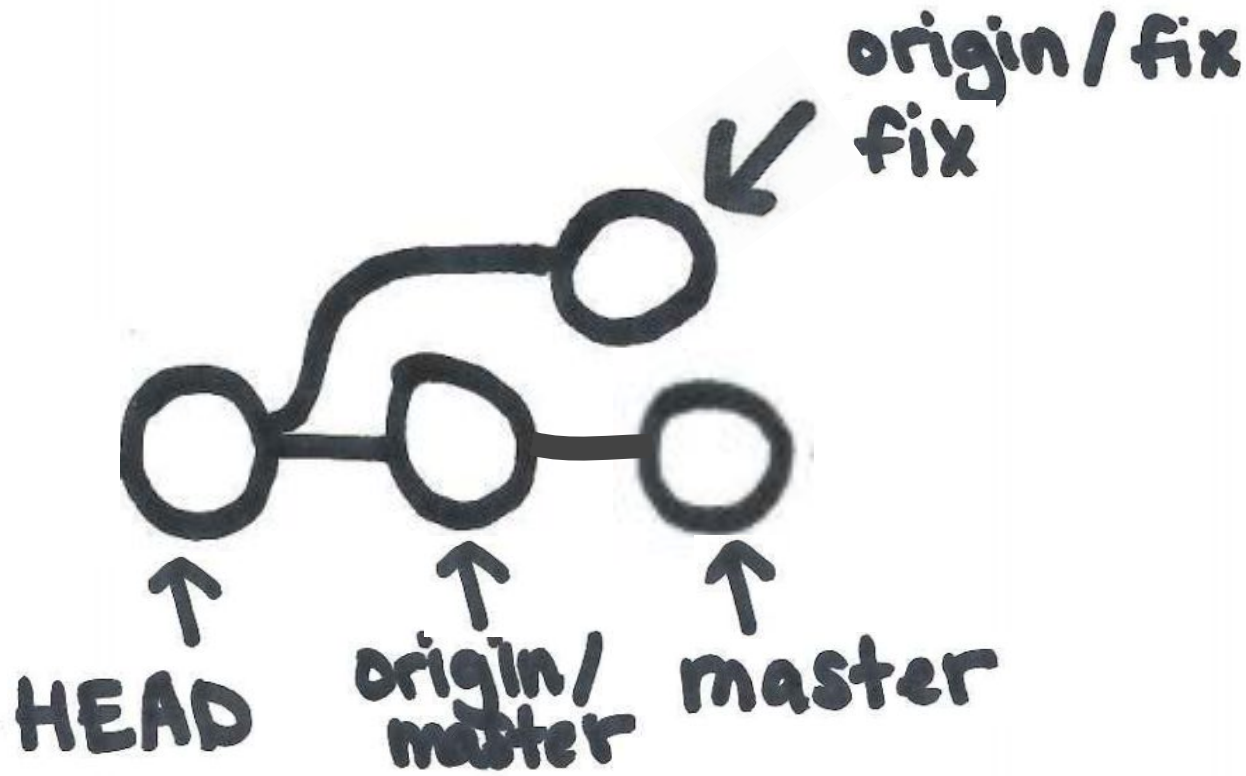
Assume that all add/commit combinations has changes and creates a commit.

```
git init

git commit -a -m "First commit"

git commit -a -m "Second commit"

git remote add origin <url>
```

(Assume remote has an empty repository.)

```
git push origin master

git checkout HEAD~1

git branch fix

git checkout fix

git commit -a -m "Third commit"

git push origin fix

git checkout master

git commit -a -m "Fourth commit"

git checkout HEAD~2
```

Exercise answer (amended from video)



Questions?