UNIVERSITY of WASHINGTON

Intermediate Git/Github

Valentina Staneva[1] and Bernease Herman[1]

[1]eScience Institute

November 2, 2018

Agenda

1. Fork a repository and setup
2. Review of introductory git
3. Navigating the git tree
4. Revert, reset, and checkout
5. Git branches and merging **+ Exercises**
6. Collaboration workflows
7. Pull requests **+ Exercises**
8. Review exercise

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)

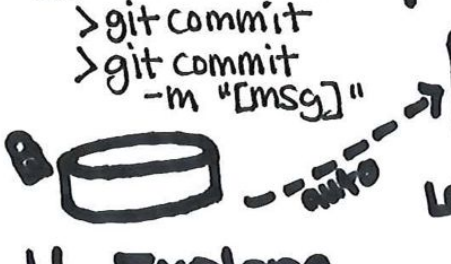
0. Set up
> git config [options]
> git init
> .gitignore

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)

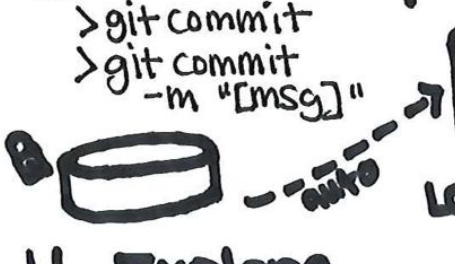
0. Set up
>git config [options]
>git init
>.gitignore

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)

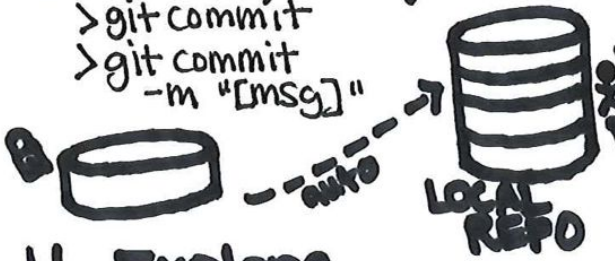
0. Set up
>git config [options]
>git init
>.gitignore

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)

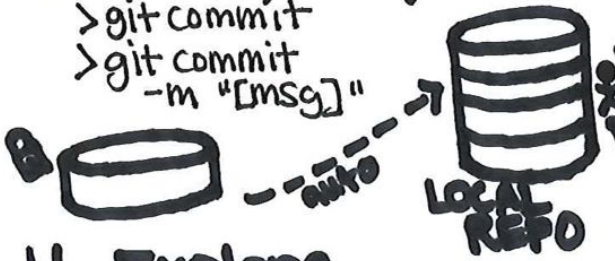
0. Set up
>git config [options]
>git init
>.gitignore

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]

Introduction To Git
*(and some GitHub)

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

1. Make changes

(use your preferred editor and tools.)

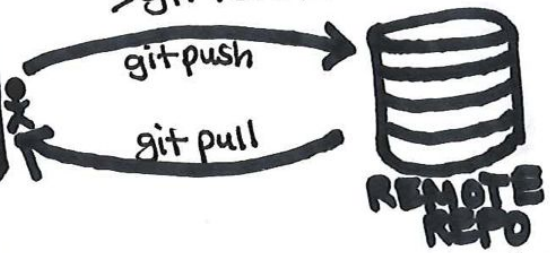
2. Stage changed files
>git add
>git add -A
>git rm [path]

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

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

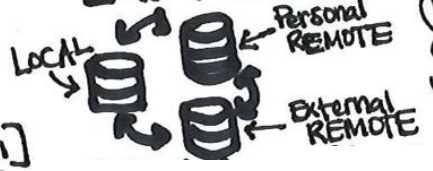
(Repeat 1-4 as desired.)

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]

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]

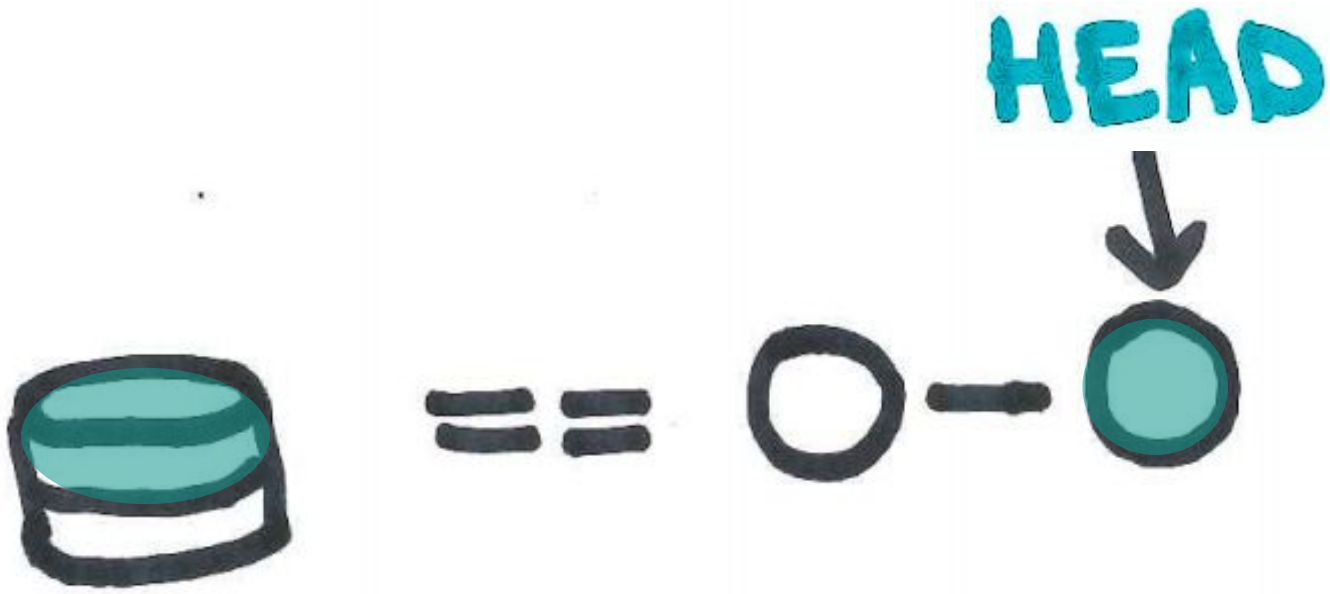
Bernease Herman 10/4/18

A single commit

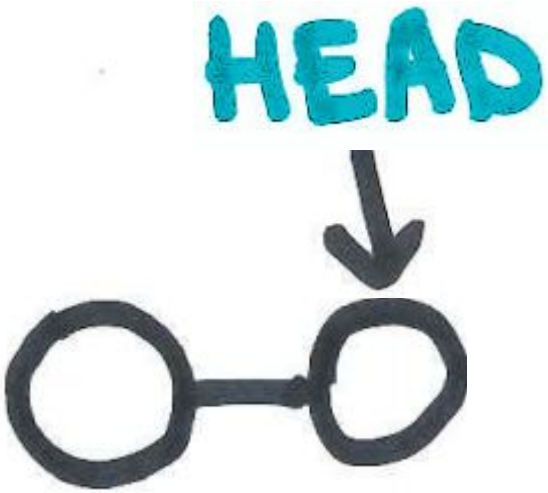
In tree representation

Multiple commits represented

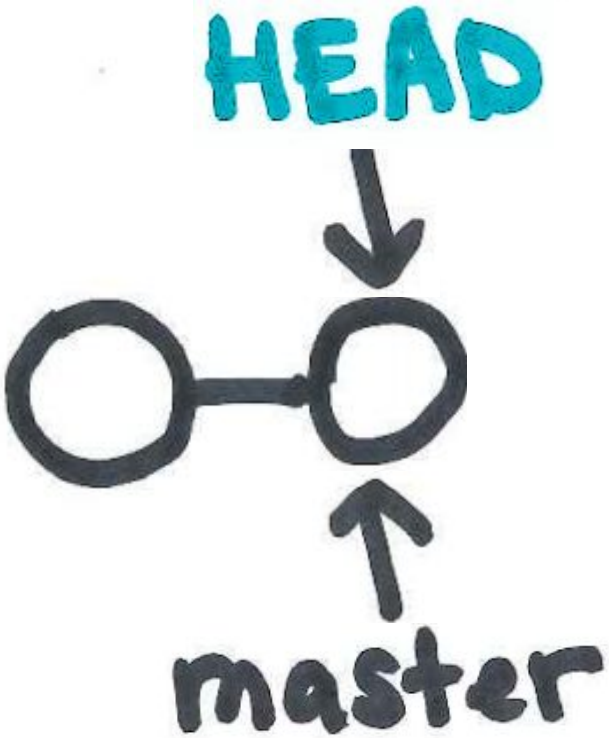
Your working directory and files



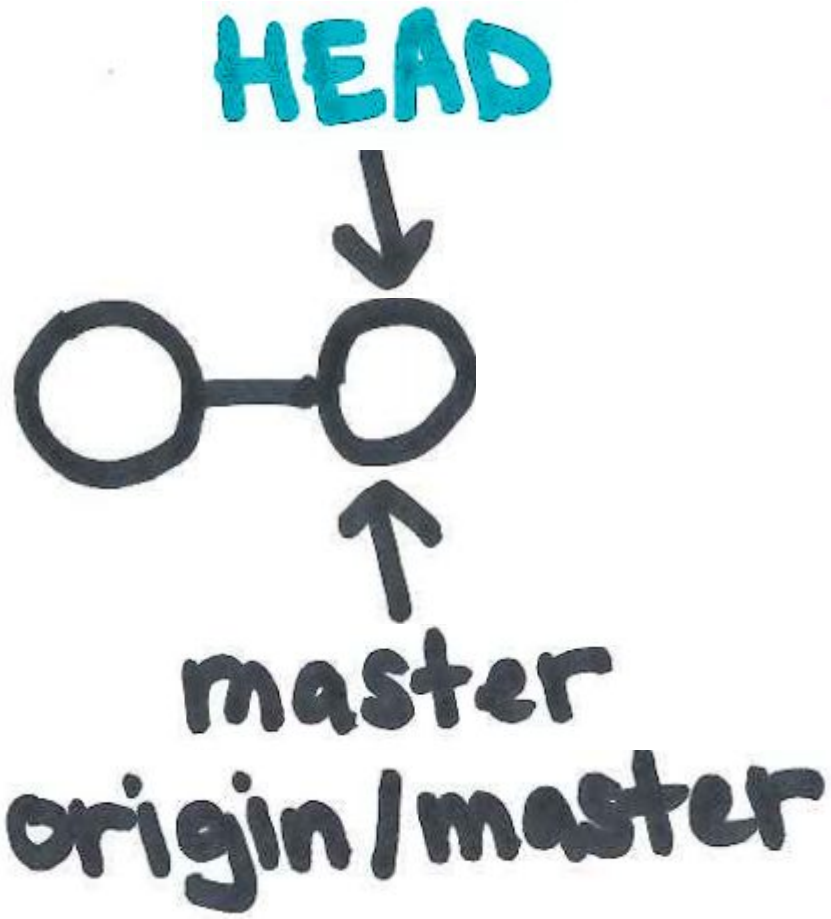
HEAD pointer on our tree



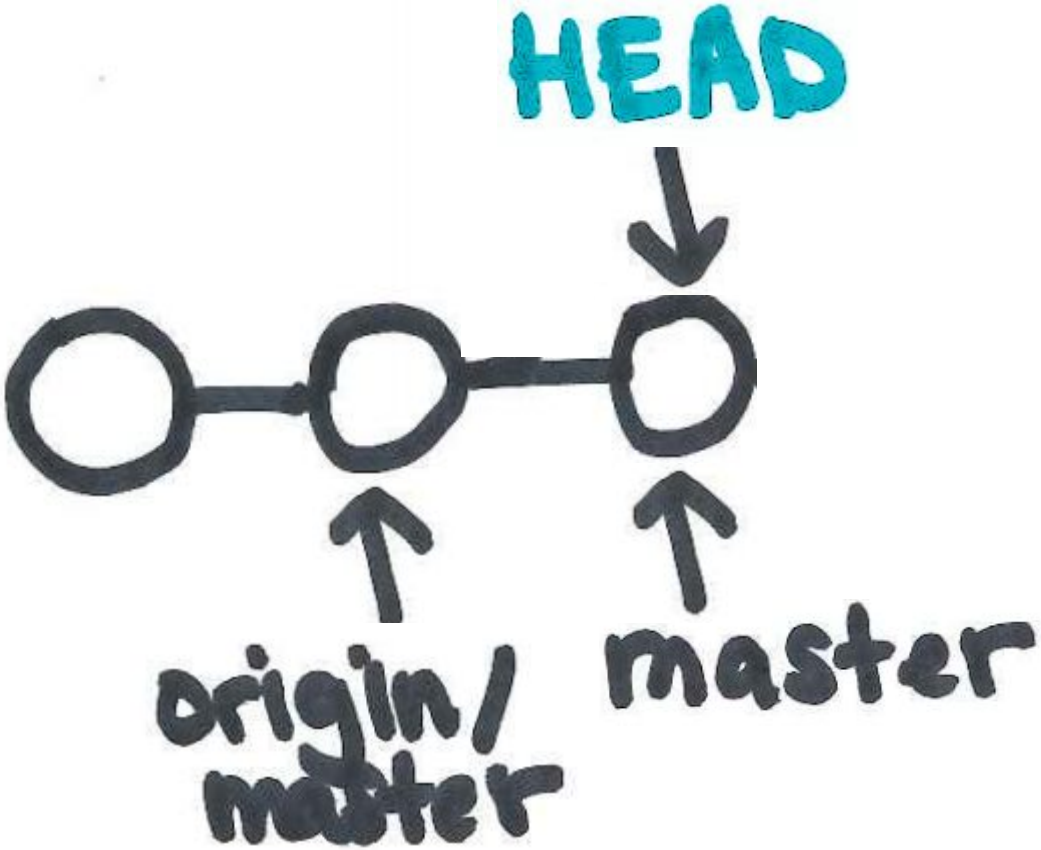
Think of branches as a pointer, as well



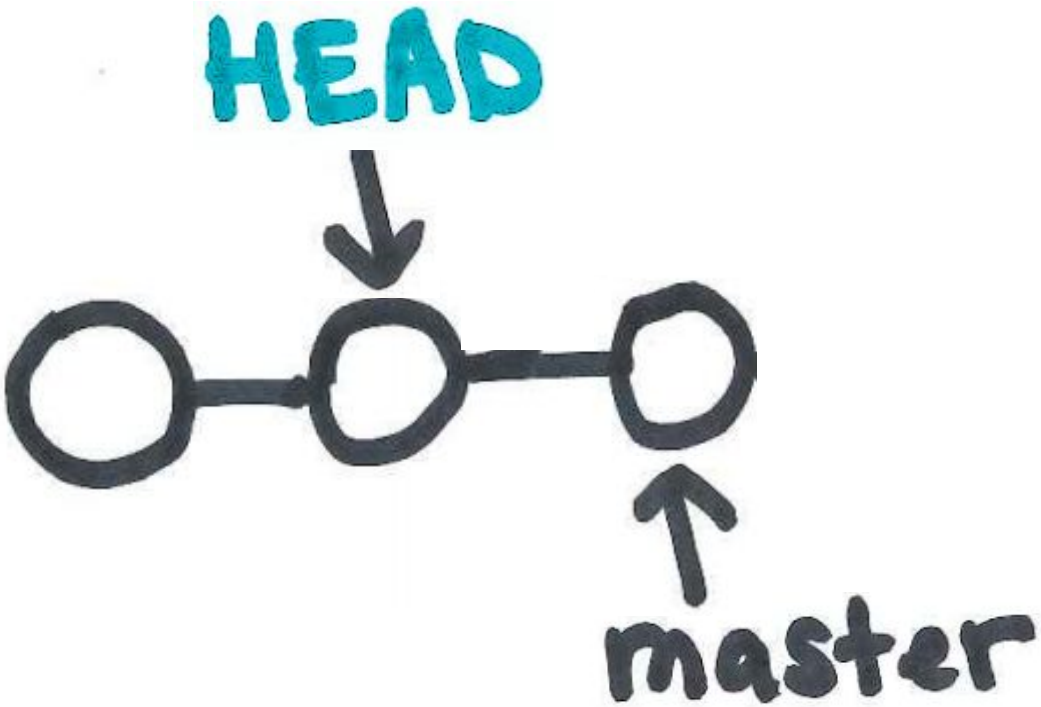
Remote branches are included



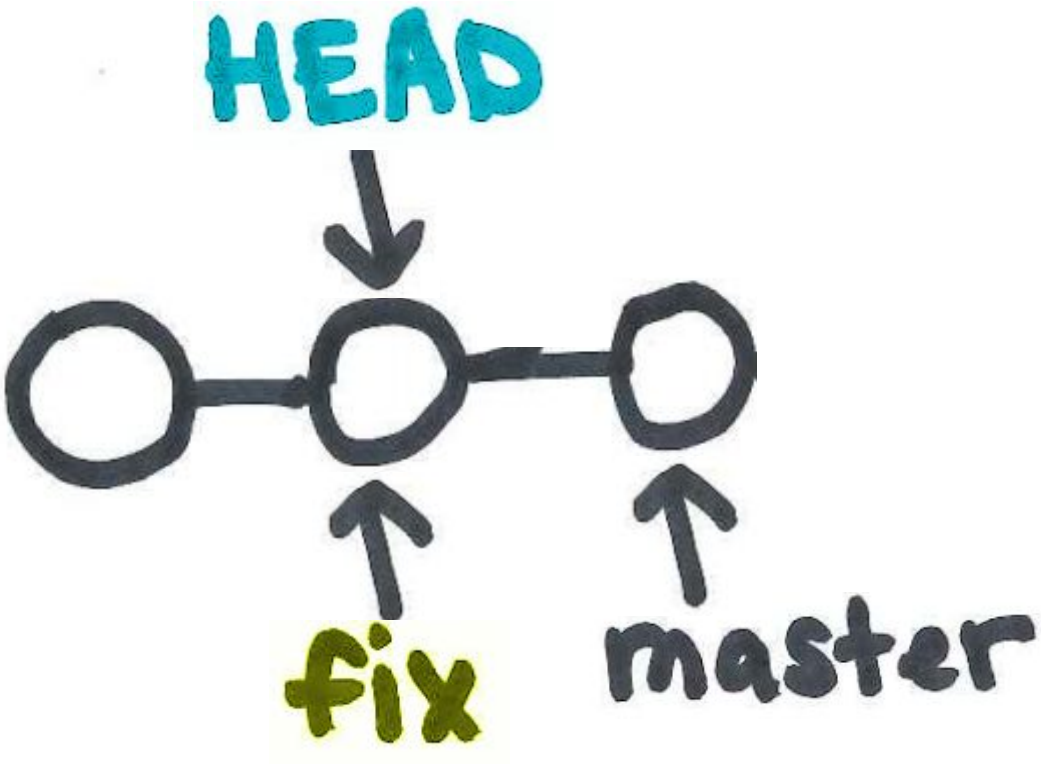
Local commit, before pushing to remote



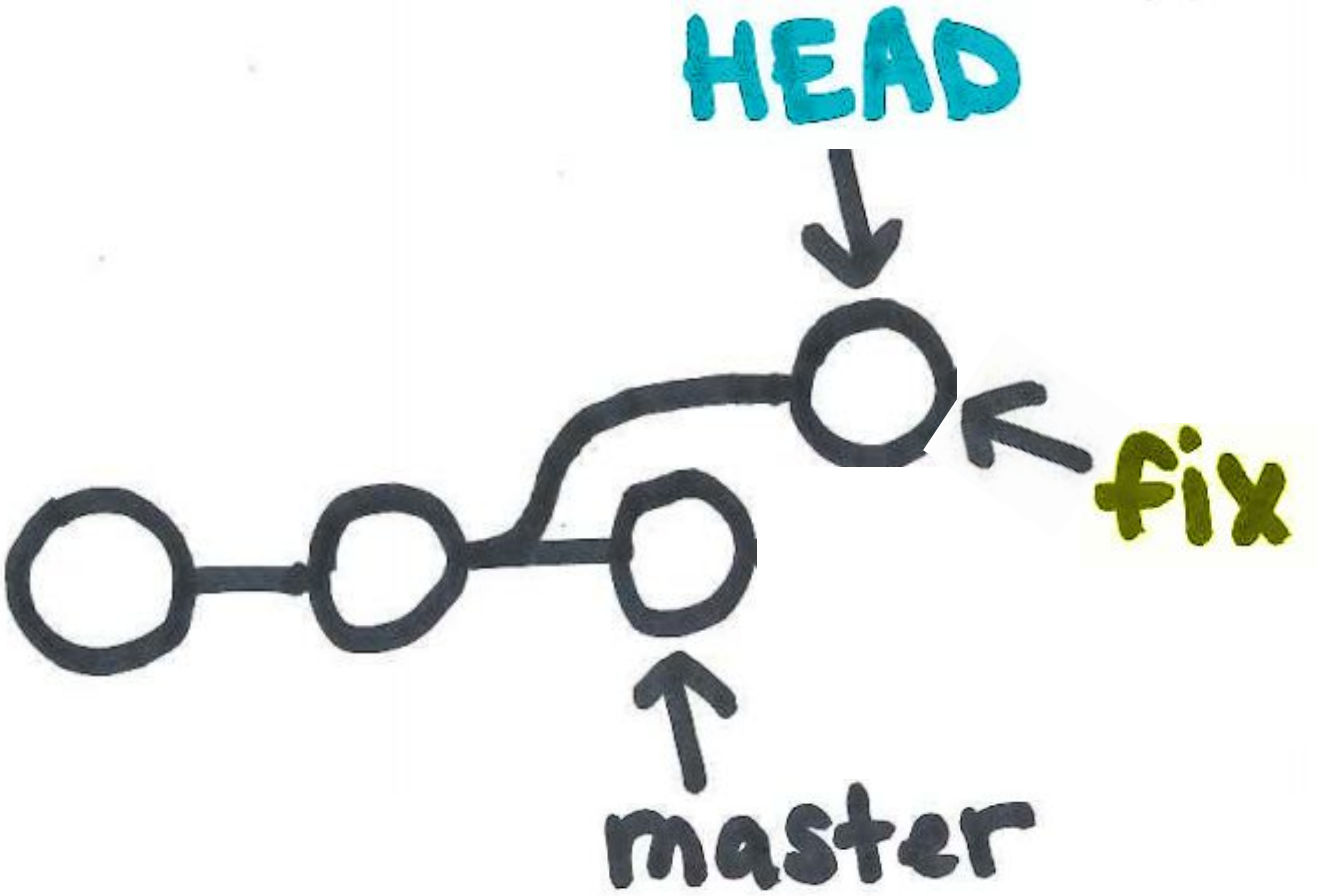
Checkout an earlier commit (hiding `origin/master` for simplicity)



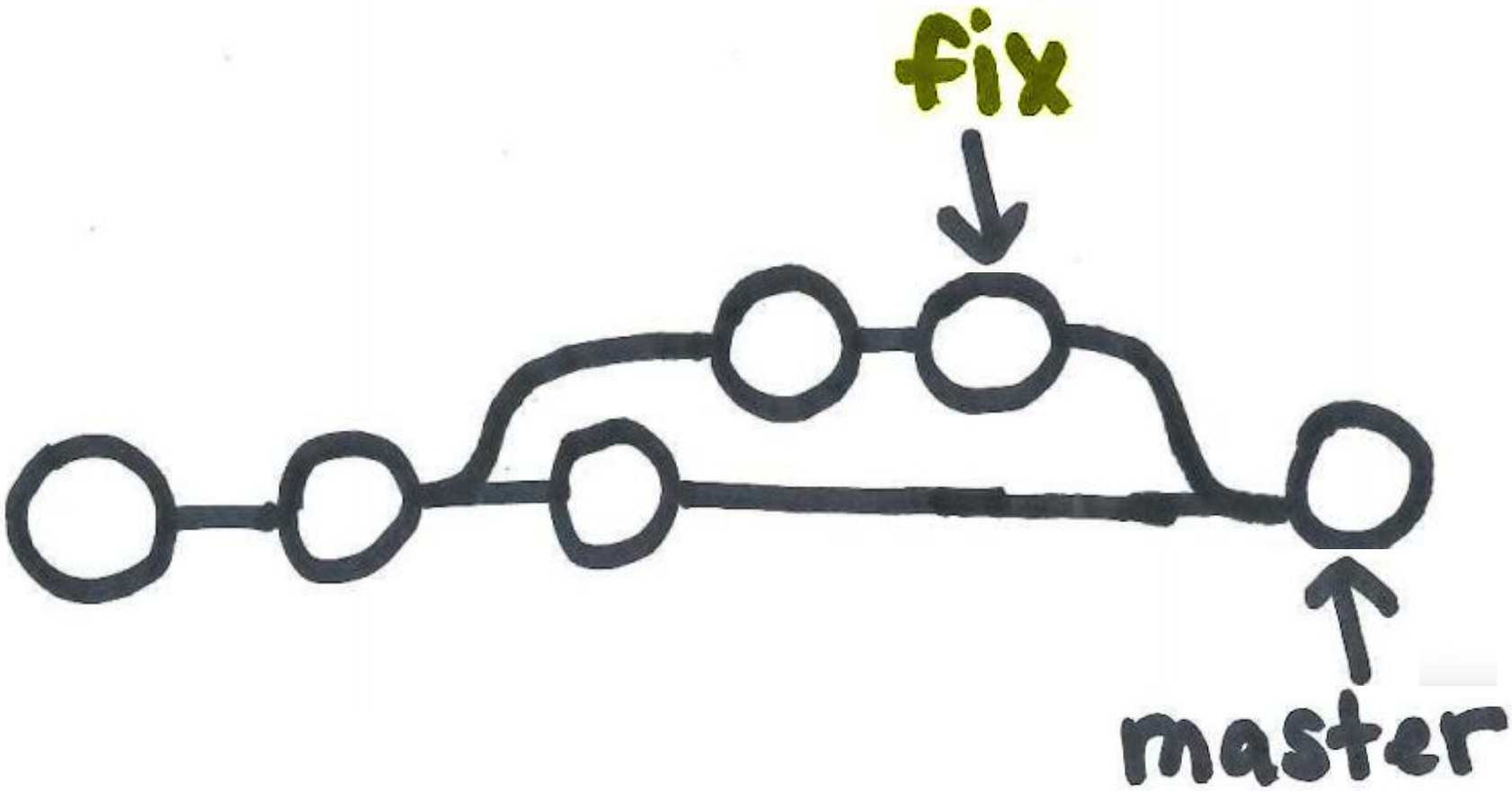
Creating a new branch



Making changes along this branch



Merging commits to another branch



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
```

Collaboration workflows

Do you need a stable version of the code at all times?

Do you have a regular schedule for releases?

Do you trust changes from the collaborators equally?

Single shared repository, Forking workflow, Feature branching, Git Flow strategy

See notes in **uwescience/git-intermediate repo**
More info at **Atlassian's git tutorials**

origin/fix

origin/master

origin/fix

origin/master

fix

origin/master

master

HEAD

Questions?