

# Git and GitHub

# What is Git?

- Version Control System
  - Keep careful track of changes in your files
  - Collaborate with others on your projects more easily
  - Test changes without losing the original versions
  - Revert back to older versions when/if needed
- GitHub: web-based hosting service for git
  - Provides a "remote" location for storing your git workspaces
  - Useful if you lose/break your computer, etc.

# Using Git

- Installation
  - <https://github.com/join>
  - <https://help.github.com/articles/set-up-git/>
- How it works
  - Create a "repository" (workspace) for your project
  - Add/remove/save/edit files
  - Push local files online to GitHub / pull remote files from GitHub to your local workspace
  - And more!

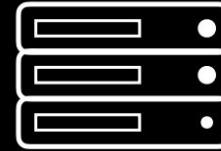
# git clone <url>

- Downloads an existing repository from GitHub
- Creates a synced, local copy

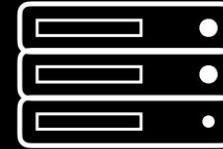
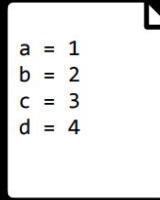
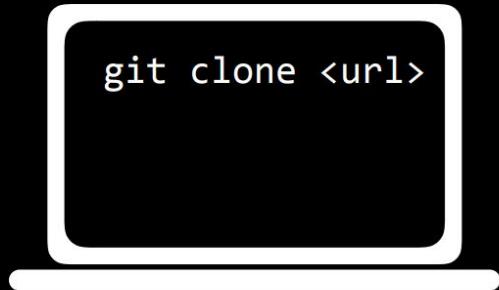
```
git clone <url>
```



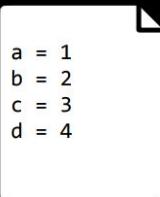
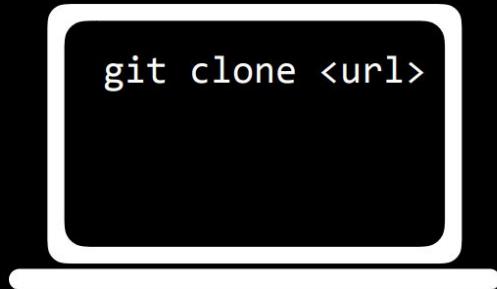
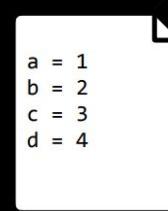
```
a = 1  
b = 2  
c = 3  
d = 4
```



```
git clone <url>
```



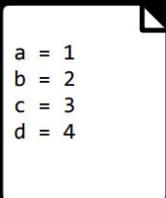
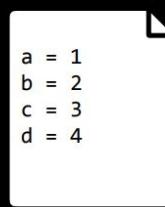
```
git clone <url>
```



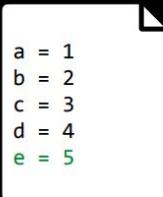
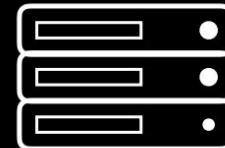
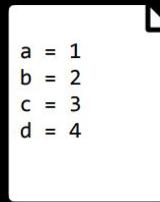
# git add <filename>

- Signals to git that the specified file should be “tracked” for changes
  - Places modified file(s) in the “staging area”
- Files not added in this way are essentially ignored by git
- `git add -A` signals to git that it should track all existing files

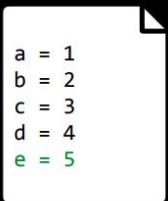
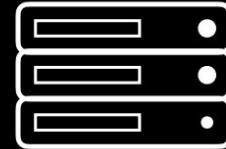
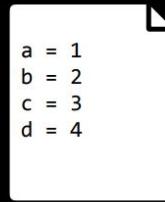
```
git add <filename>
```



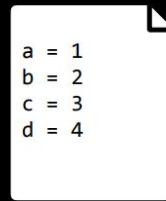
```
git add <filename>
```



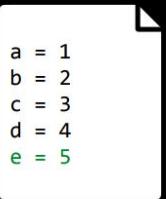
```
git add <filename>
```



```
git add <filename>
```



```
git add foo.py
```



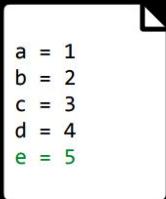
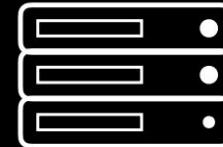
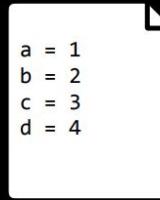
Changes to be committed:

modified: foo.py

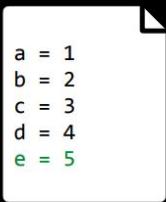
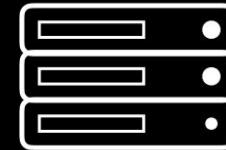
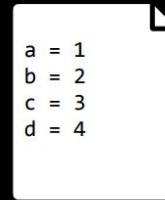
# git commit -m "message"

- Takes a "snapshot" of all files currently on the staging area and commits it to git's memory
- The "snapshot" is captioned with the given message as a brief description for the commit

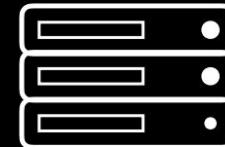
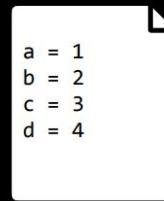
```
git commit -m "message"
```



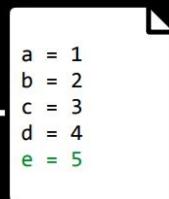
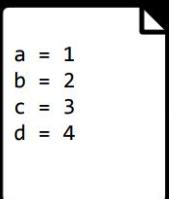
```
git commit -m "message"
```



```
git commit -m "message"
```



```
git commit -m  
"Add line"
```



Add line

```
git commit -am "message"
```

- Nearly identical to previous command, with the added step of applying `git add` to all existing tracked files first
  - Ignores untracked files

# git status

- Displays useful information about your repository (e.g., current branch, tracked/untracked files, differences between local and remote versions)

# git status



```
a = 1  
b = 2  
c = 3  
d = 4
```

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

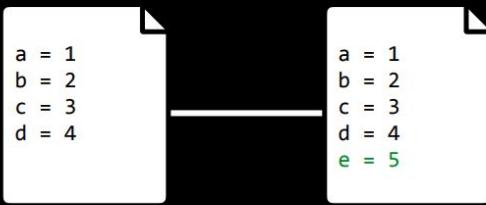
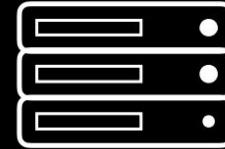
Add line

```
a = 1  
b = 2  
c = 3  
d = 4
```



# git status

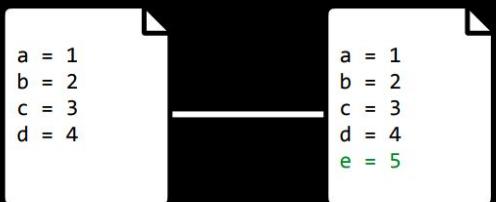
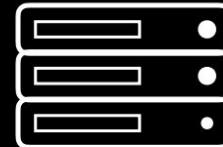
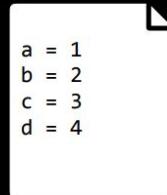
```
a = 1  
b = 2  
c = 3  
d = 4
```



Add line



# git status



Add line

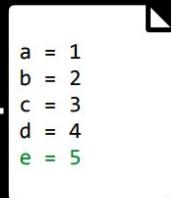
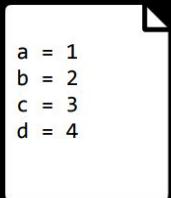
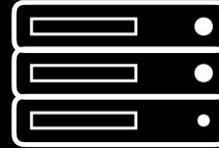
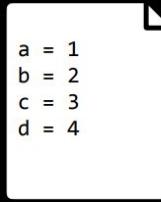
git status

On branch master  
Your branch is ahead of 'origin/master' by 1 commit.  
(use "git push" to publish your local commits)

# git push

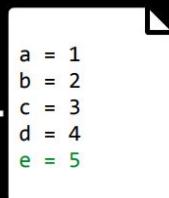
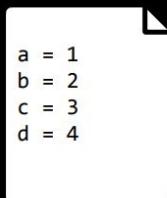
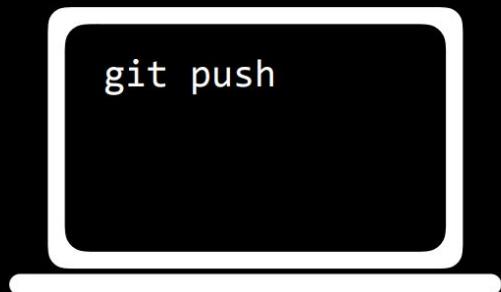
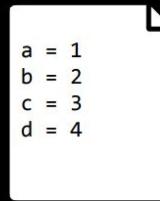
- Uploads local commits to the remote repository (i.e., from your computer to GitHub)

git push



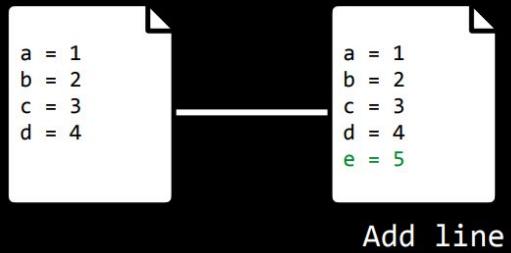
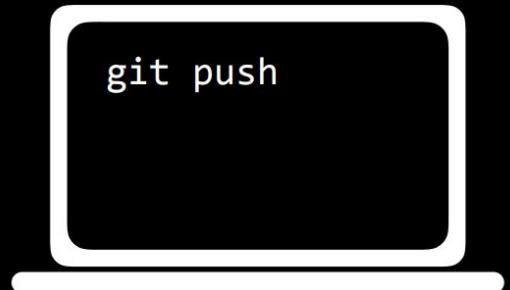
Add line

git push

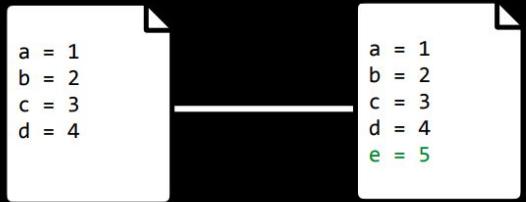


Add line

git push



Add line



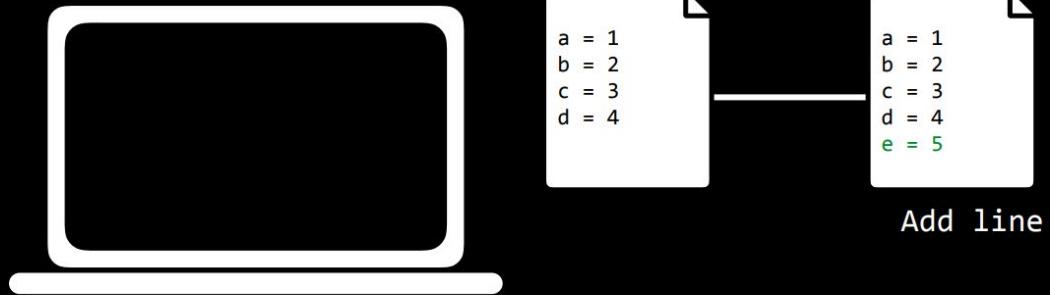
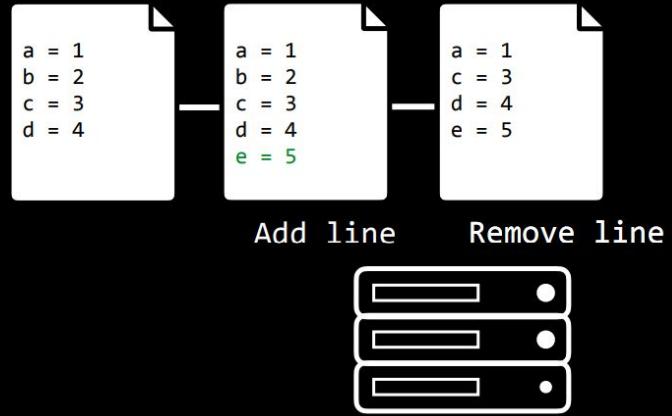
Add line



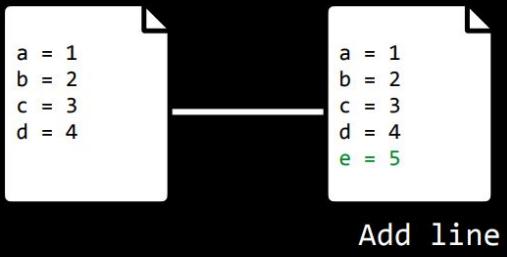
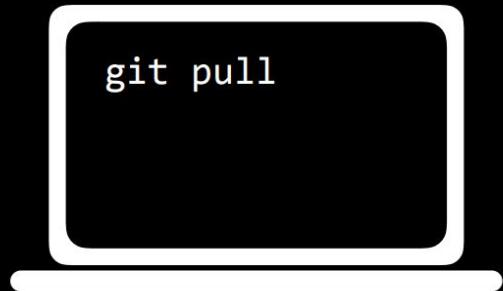
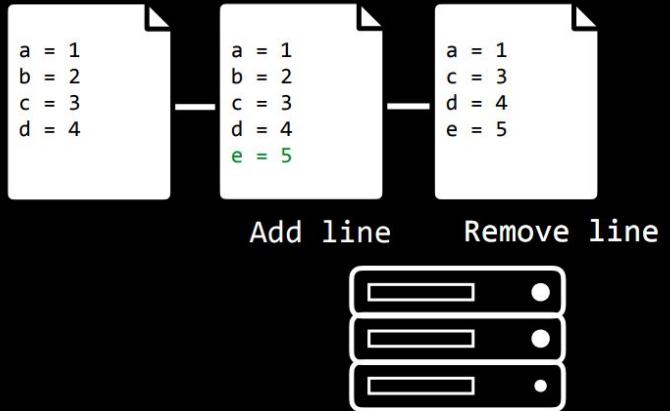
# git pull

- Downloads remote commits to the local repository (i.e., from GitHub to your computer)

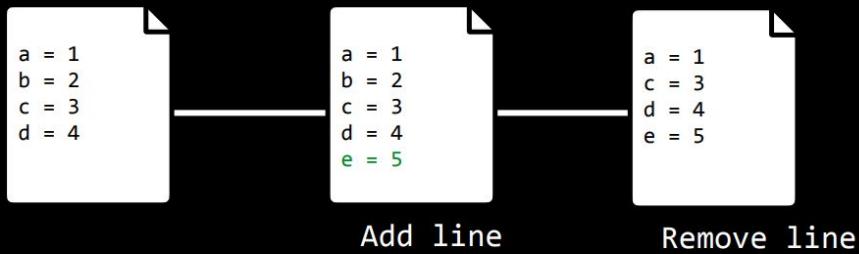
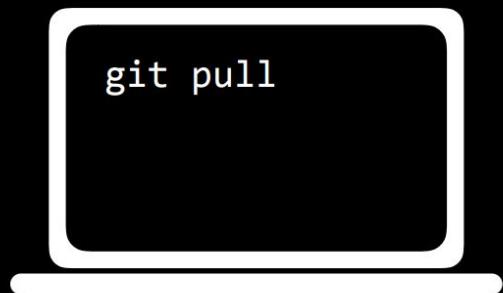
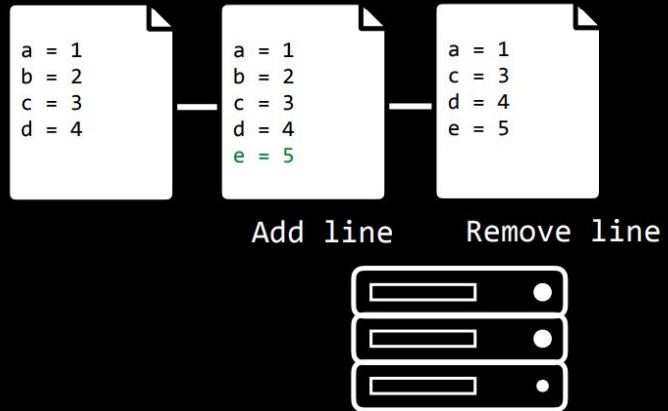
# git pull



# git pull



# git pull



# Merge Conflicts

- When two collaborators make conflicting changes to the same file, a merge conflict may arise
- Git will complain when you attempt to `git pull` and you will need to manually resolve the conflict

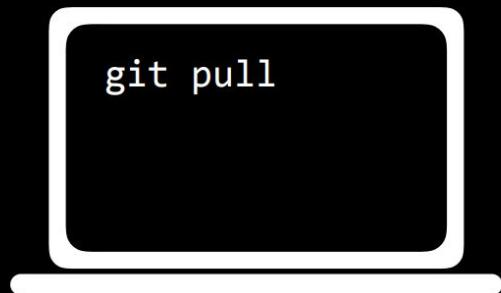
# Merge Conflicts



# Merge Conflicts



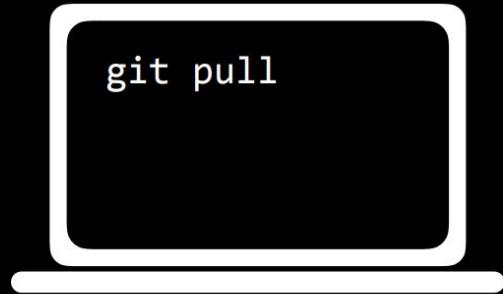
# Merge Conflicts



```
git pull
```

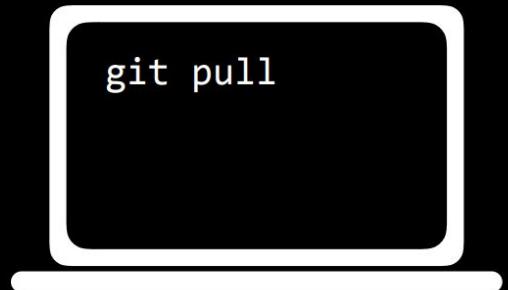
CONFLICT (content): Merge conflict in foo.py  
Automatic merge failed; fix conflicts and then  
commit the result.

# Merge Conflicts



```
a = 1
<<<< HEAD
b = 2
=====
b = 0
>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```

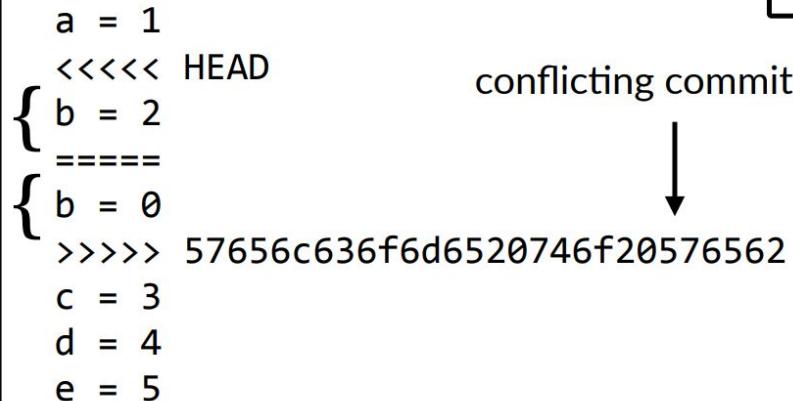
# Merge Conflicts



your  
changes  
  
remote  
changes

```
a = 1
<<<< HEAD
{
b = 2
=====
{
b = 0
>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```

conflicting commit

A large white rectangular box contains the code. At the top right, there is a small icon of a document with a folded corner. A vertical black arrow points downwards from the word "conflicting commit" towards the merge conflict marker "=====".

# Merge Conflicts



```
a = 1
<<<< HEAD
b = 2
=====
b = 0
>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```

# Merge Conflicts



```
a = 1  
  
b = 2  
  
c = 3  
d = 4  
e = 5
```

# Merge Conflicts



```
git pull
```

```
a = 1  
b = 2  
c = 3  
d = 4  
e = 5
```

# git log

- Displays history of commits made in the repository from newest to oldest

git log



git log



# git log



```
commit 436f6d6d6974204d73672048657265
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Mon Jan 22 14:06:28 2018 -0400
```

Remove a line

```
commit 57656c636f6d6520746f20576562
Author: Brian Yu <brian@cs.harvard.edu>
Date:   Mon Jan 22 14:05:28 2018 -0400
```

Add a line

# Branching

- Each repository by default has a "master" branch where all your work lives
- Sometimes useful to create separate branches in your repository (to test new features, separate work among collaborators, etc.)



first  
commit



first  
commit

changes

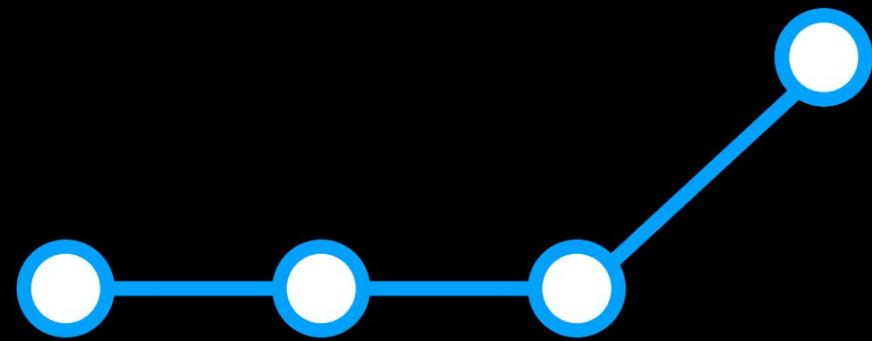


first  
commit

changes

more  
changes

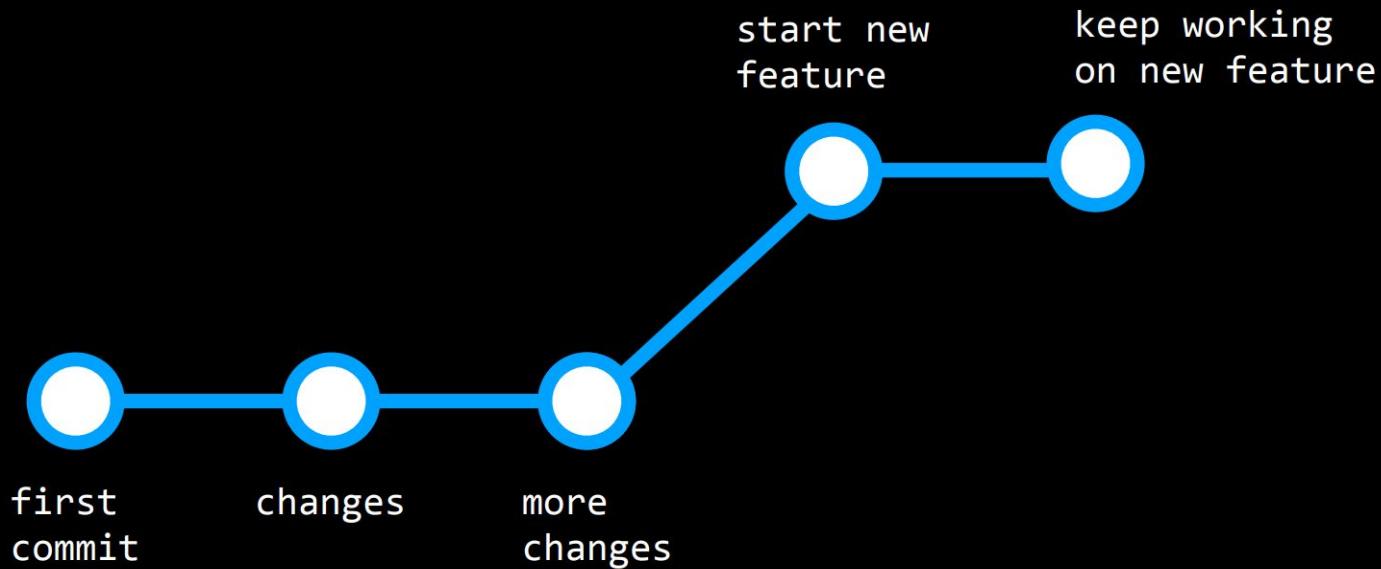
start new  
feature

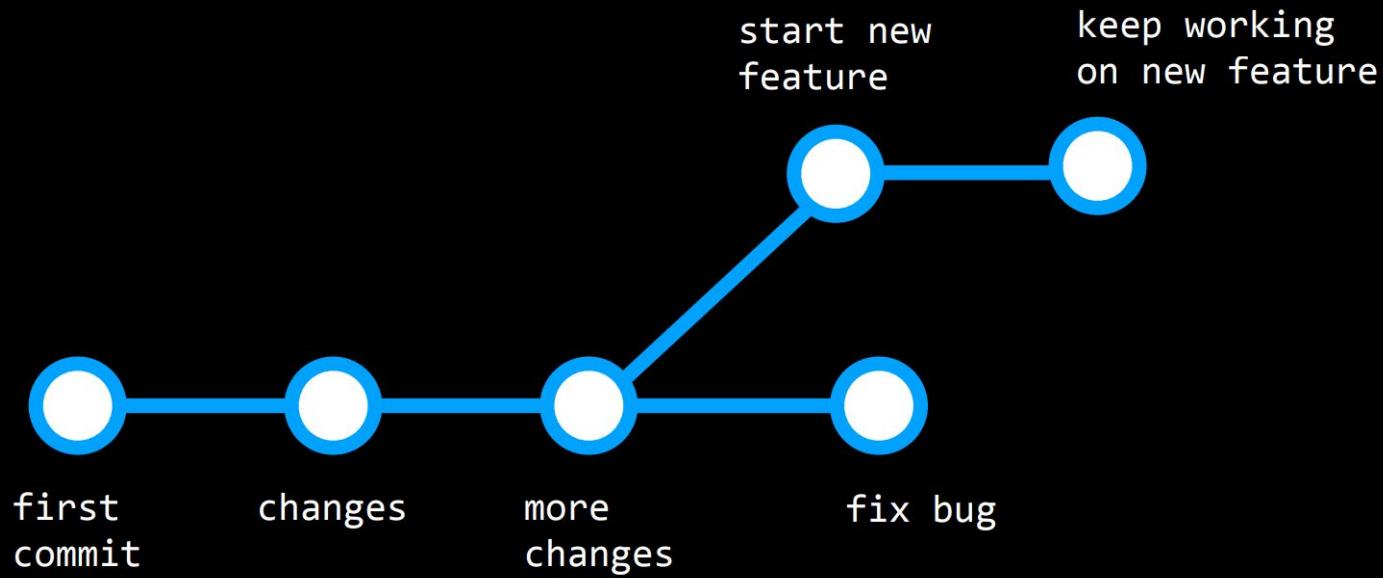


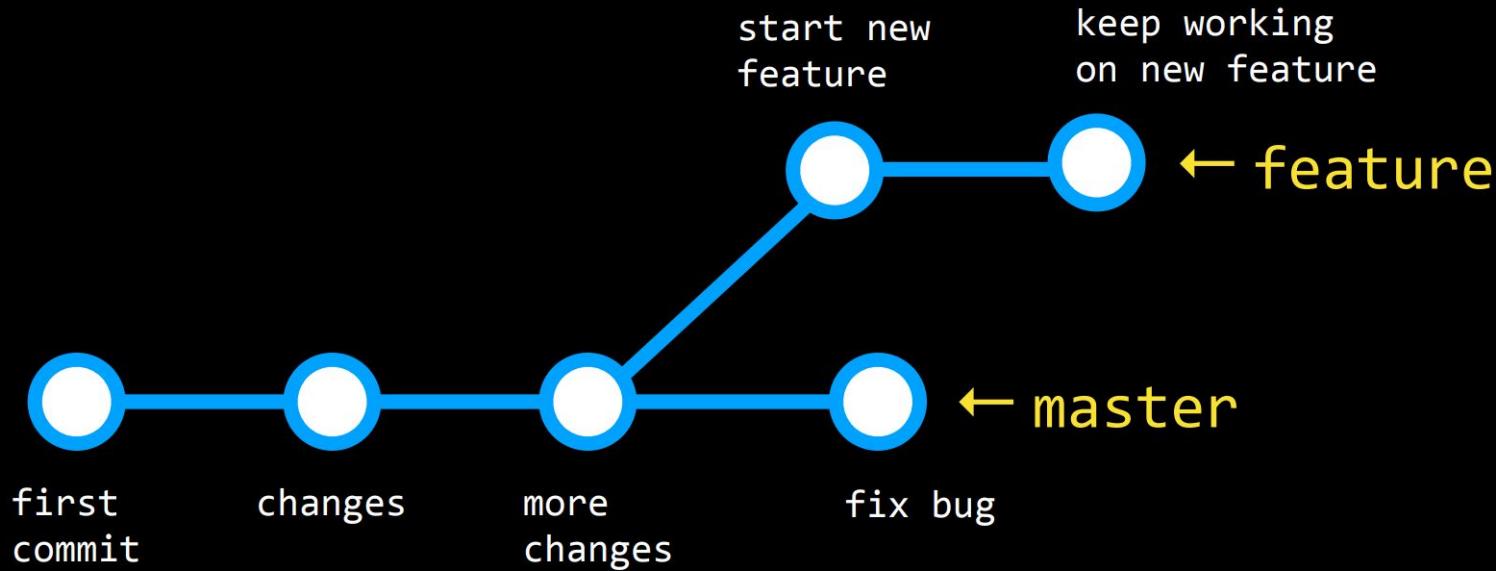
first  
commit

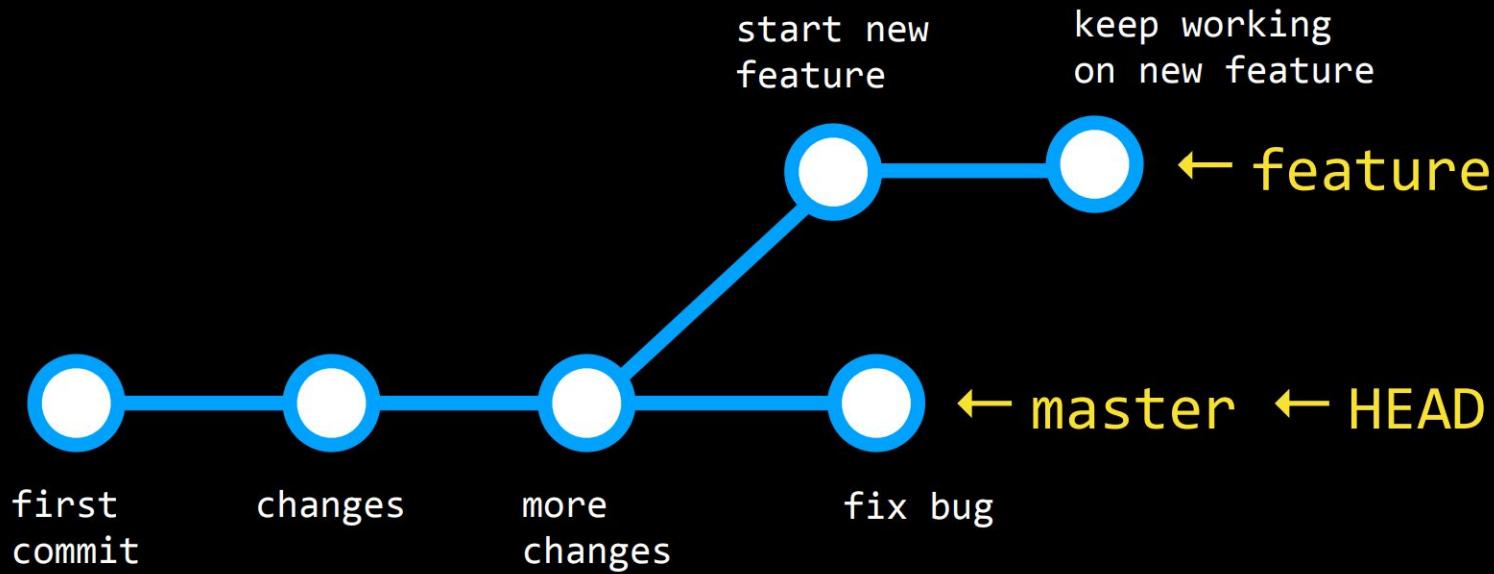
changes

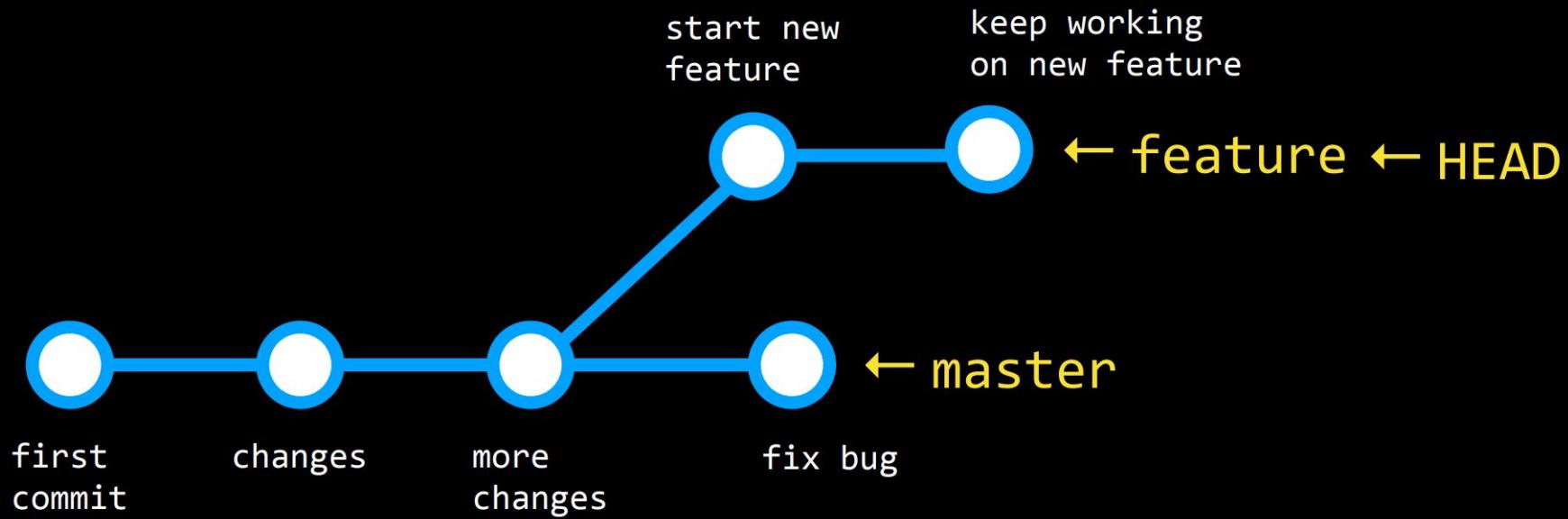
more  
changes

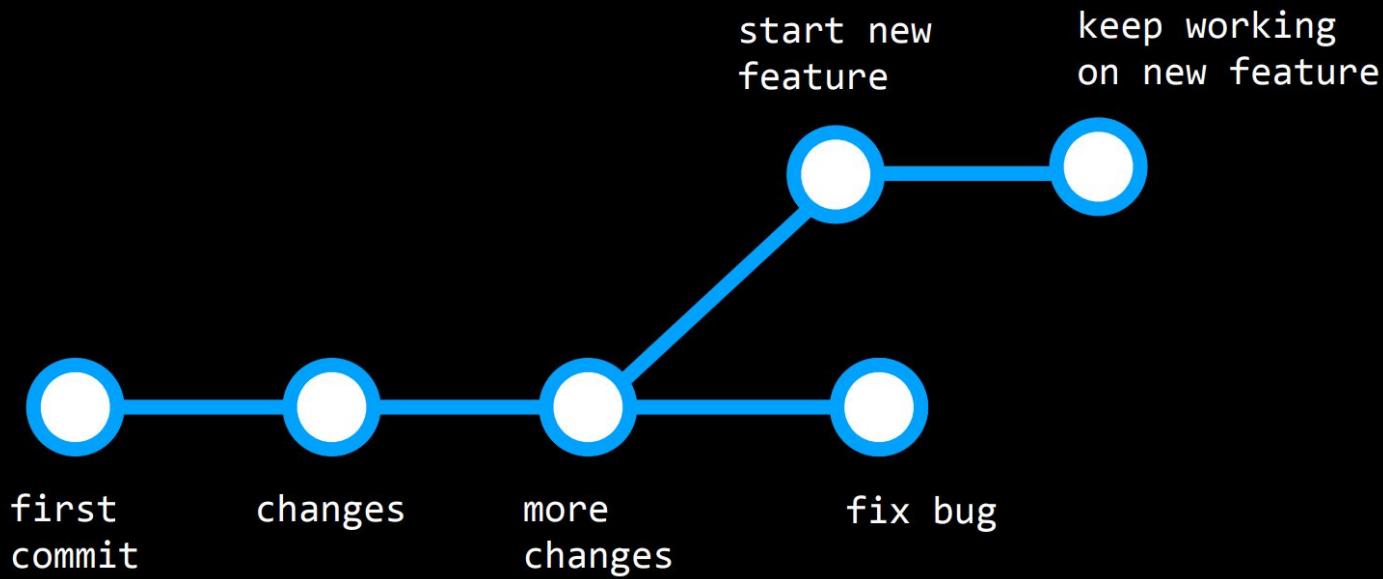


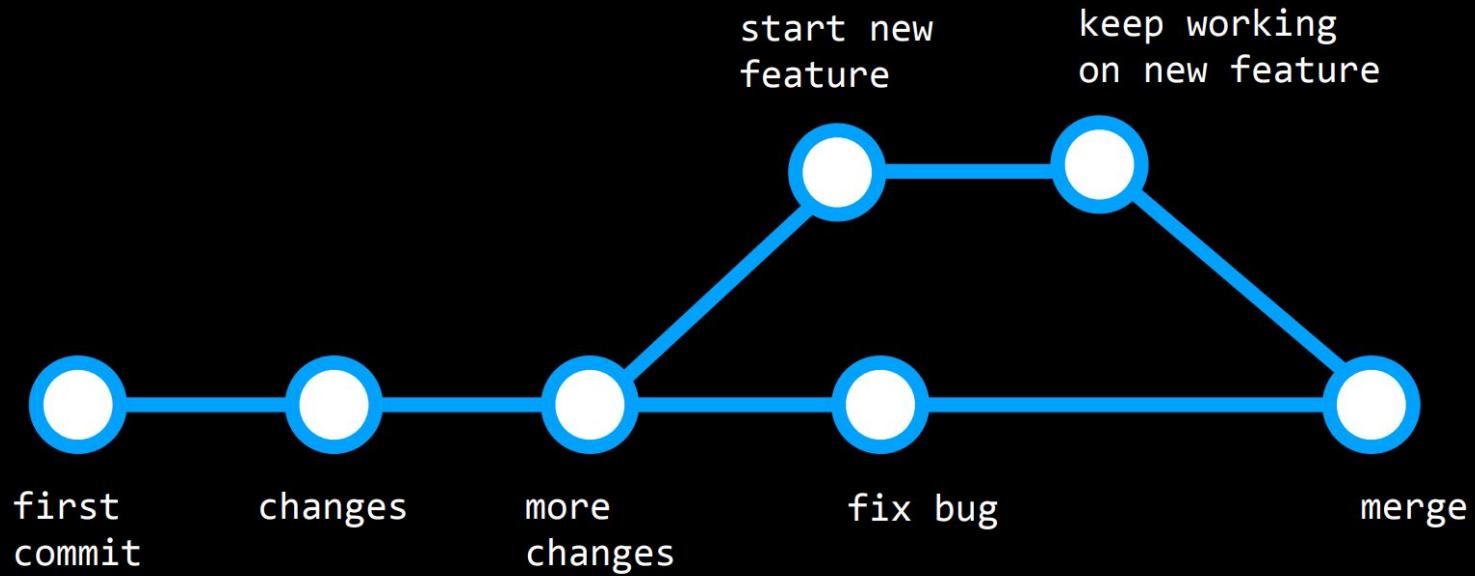












# git branch

- By default, lists all of the branches in your repository, but has a few other variations:
- `git branch <branch>`
  - Creates a new branch with the given name
- `git branch -d <branch>`
  - Deletes the specified branch

# git checkout <branch>

- Switches from the current branch to the specified branch (must already exist)

# git checkout -b <branch>

- First creates a new branch with the given name, then switches to it

# git merge <branch>

- Merges the specified branch to the current branch

# Open Source

- GitHub is a popular home for "open source" projects (i.e., projects whose source code is freely available online and may be redistributed and modified).

# Forking

- Create a copy of someone else's repository on your profile so that you can contribute to their project

Final questions?