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!

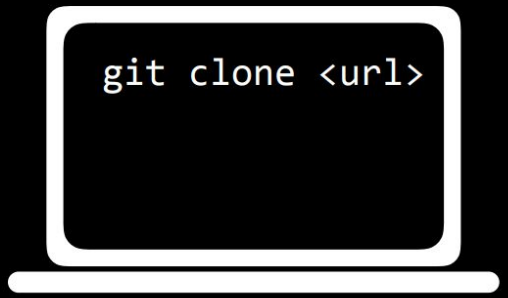
`git clone <url>`

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

git clone <url>

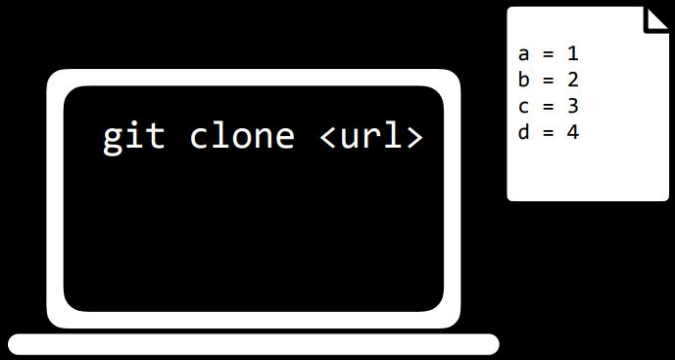
https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git clone <url>



https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git clone <url>

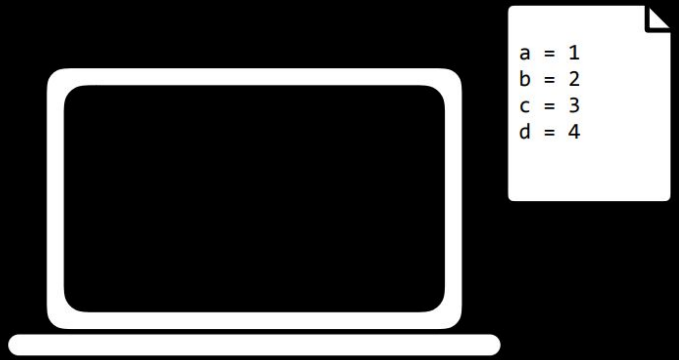


https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

`git add <filename>`

- Signals to git that the specified file should be monitored for changes
- Files not added in this way are essentially ignored by git
- `git add -A` signals to git that it should monitor all existing files

git add <filename>

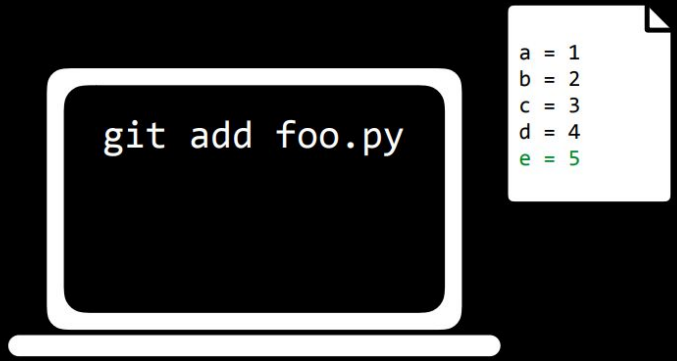


https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git add <filename>

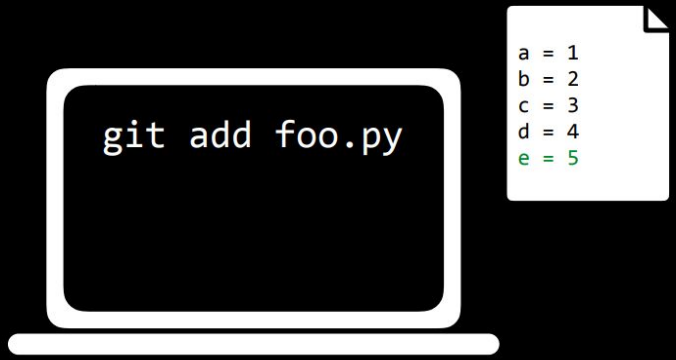
https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git add <filename>



https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git add <filename>



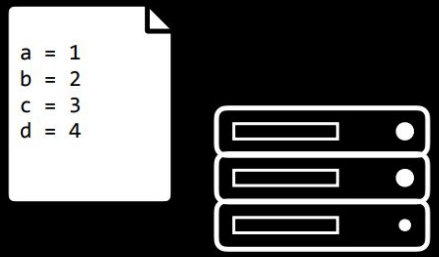

```
Changes to be committed:

    modified: foo.py
```

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

`git commit -m "message"`

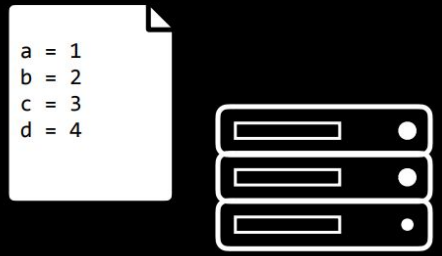
- Takes a "snapshot" of all files currently being monitored and commits it to git's memory
- The "snapshot" is captioned with the given message as a brief description for the commit

git commit -m "message"



https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

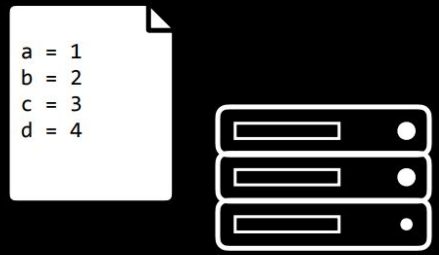
git commit -m "message"

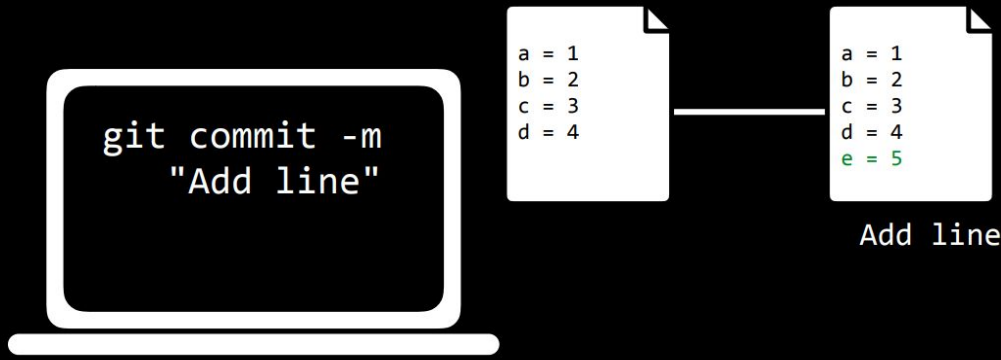



```
a = 1
b = 2
c = 3
d = 4
e = 5
```

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git commit -m "message"





https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

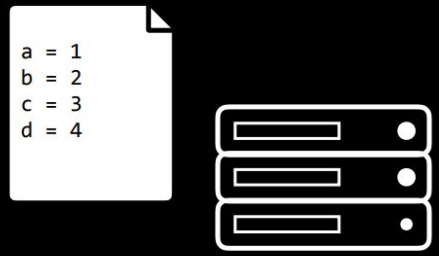
`git commit -am "message"`

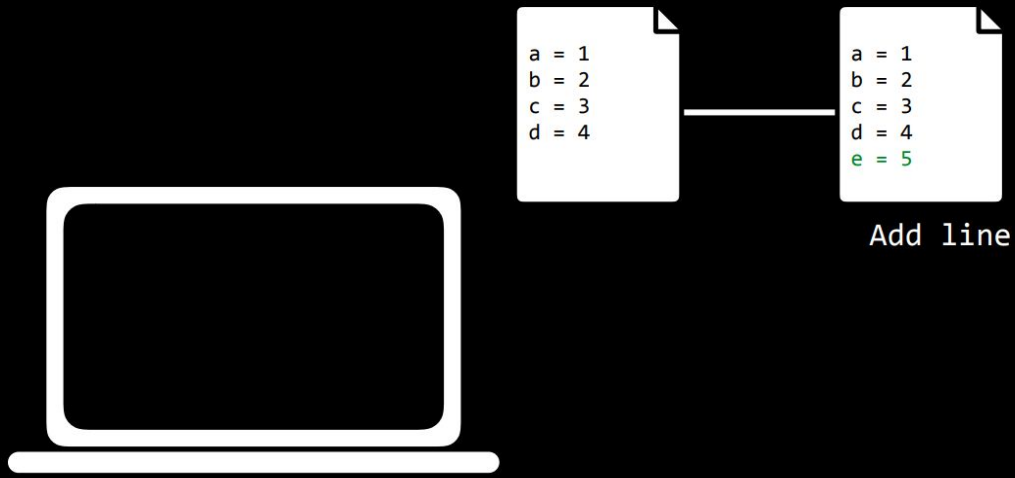
- Nearly identical to previous command, with the added step of applying `git add` to all existing files first

`git status`

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

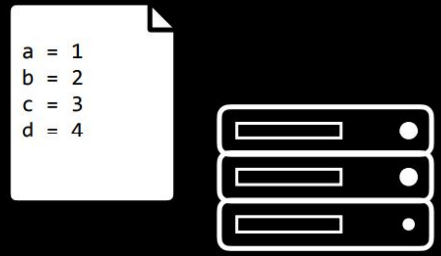
git status

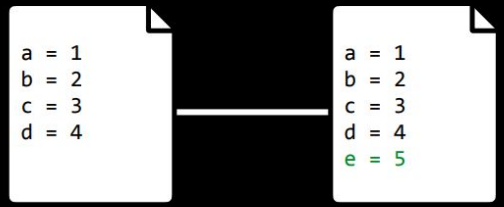




https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

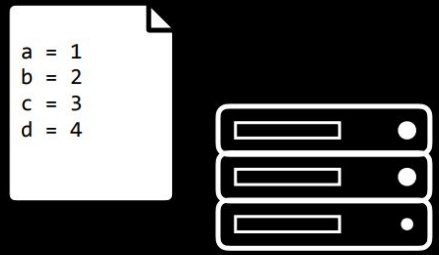
git status





https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git status



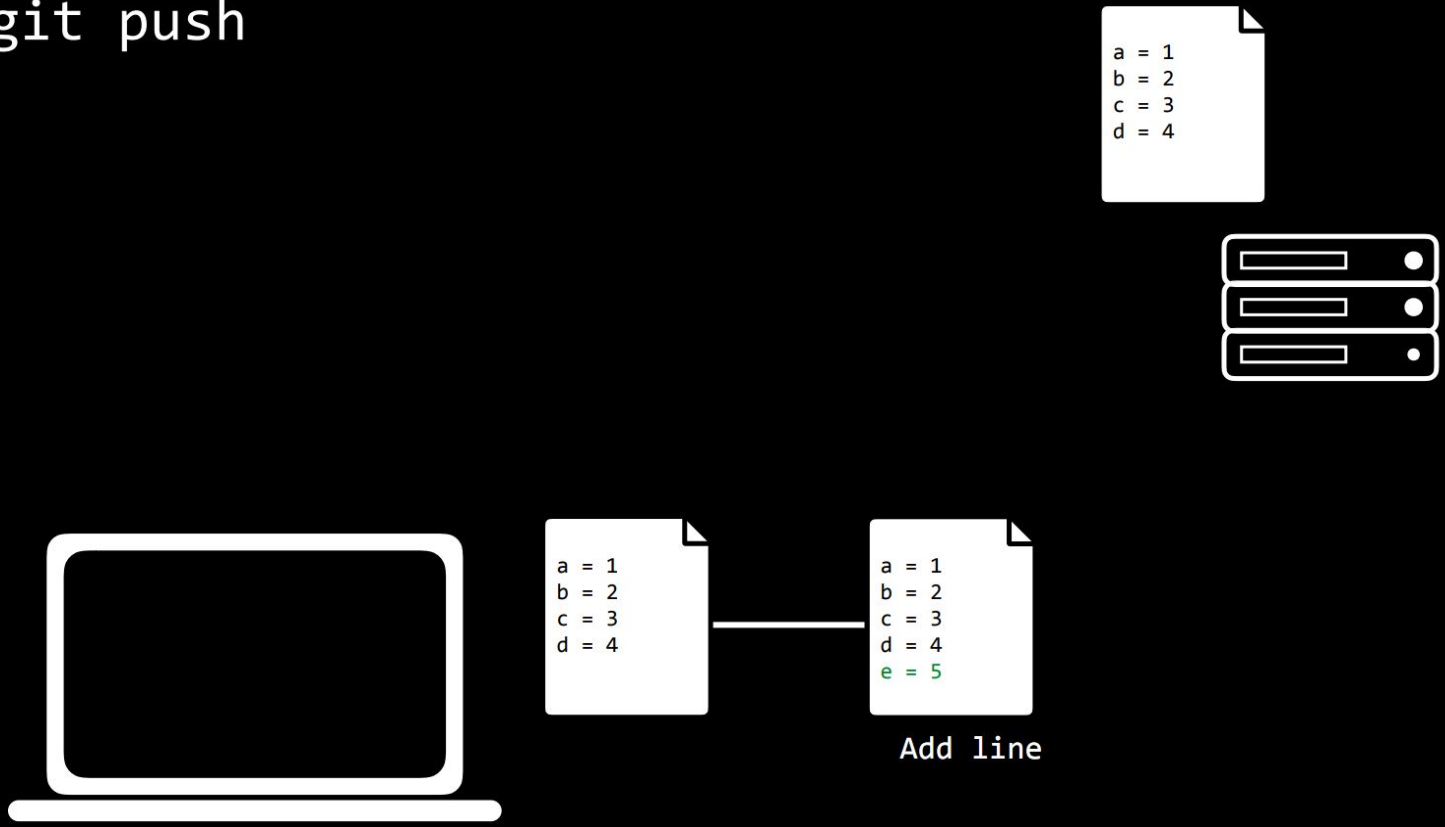

```
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
```

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

`git push`

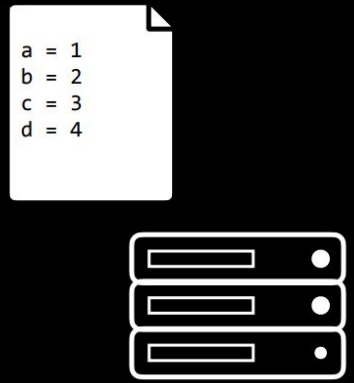
- Uploads local commits to the remote repository (i.e., from your computer to GitHub)

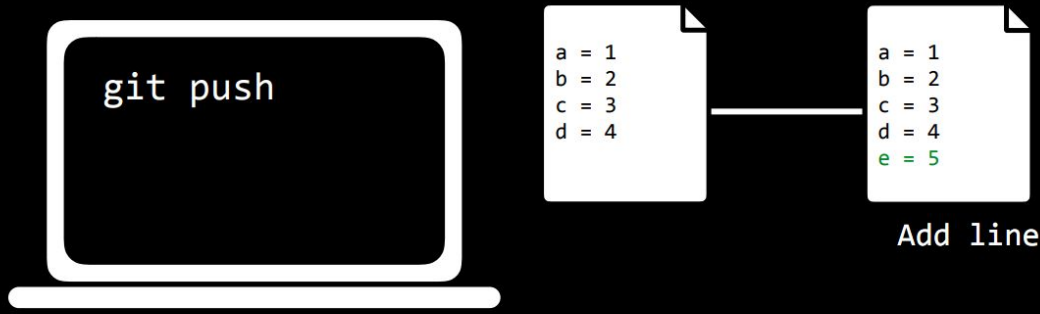
git push



https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

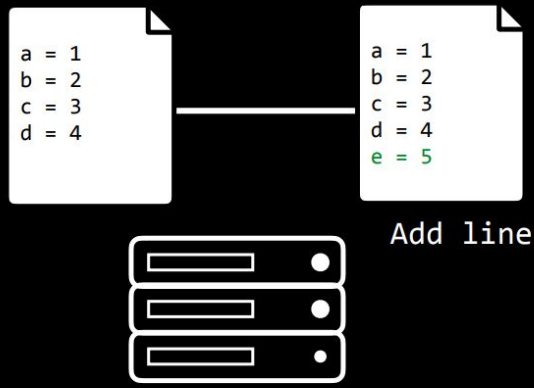
git push

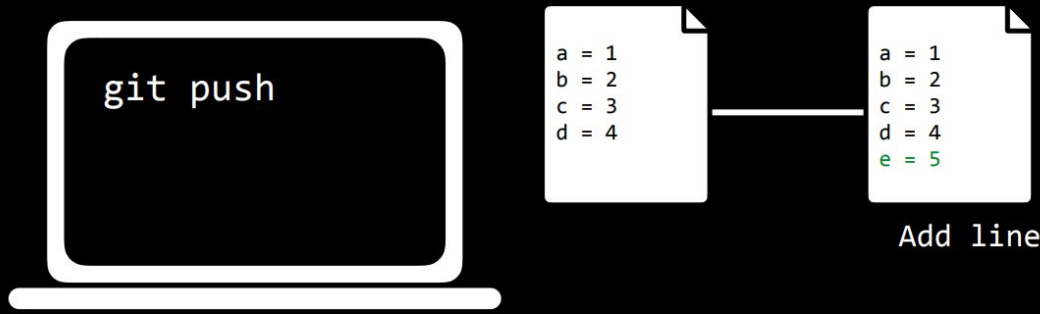




https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git push



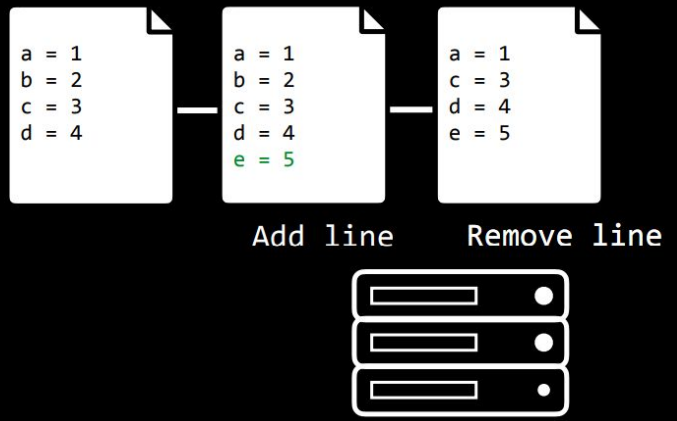


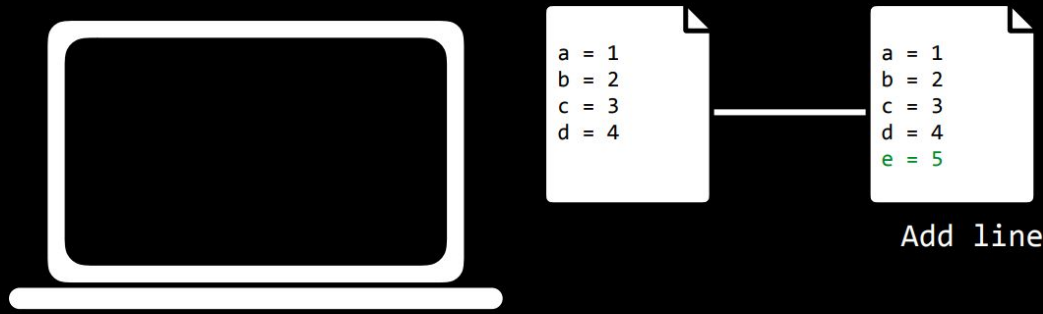
https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

`git pull`

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

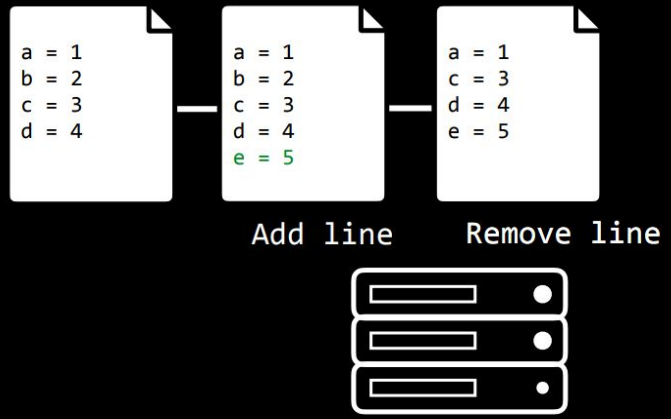
git pull

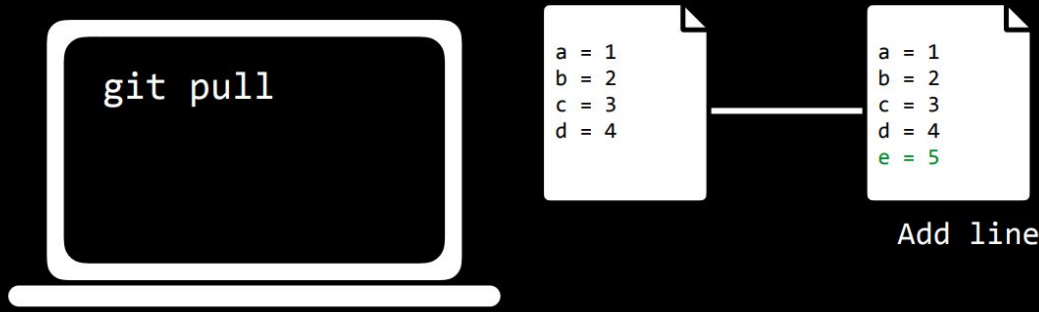




https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

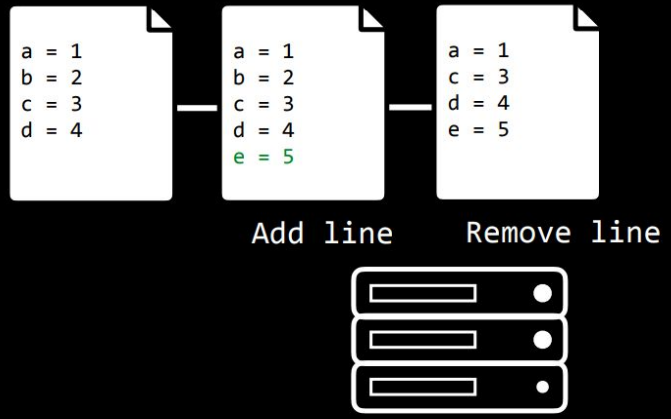
git pull

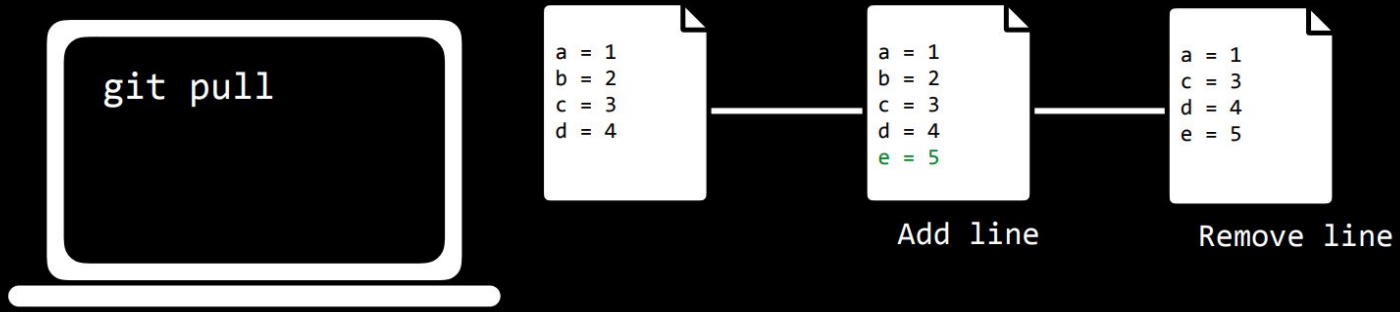




https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git pull





https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

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

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts


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

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts


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

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts

your changes

remote changes

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

conflicting commit

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts


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

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts


```
a = 1

b = 2



c = 3
d = 4
e = 5
```

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

Merge Conflicts


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

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

`git log`

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

git log

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

git log

https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

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
```

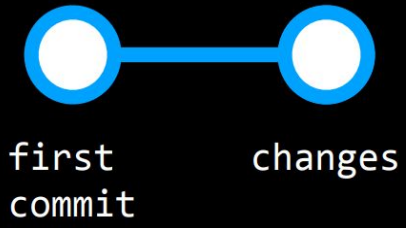
https://cs50.harvard.edu/web/2018/spring/resources/0/lecture0.pdf

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.)

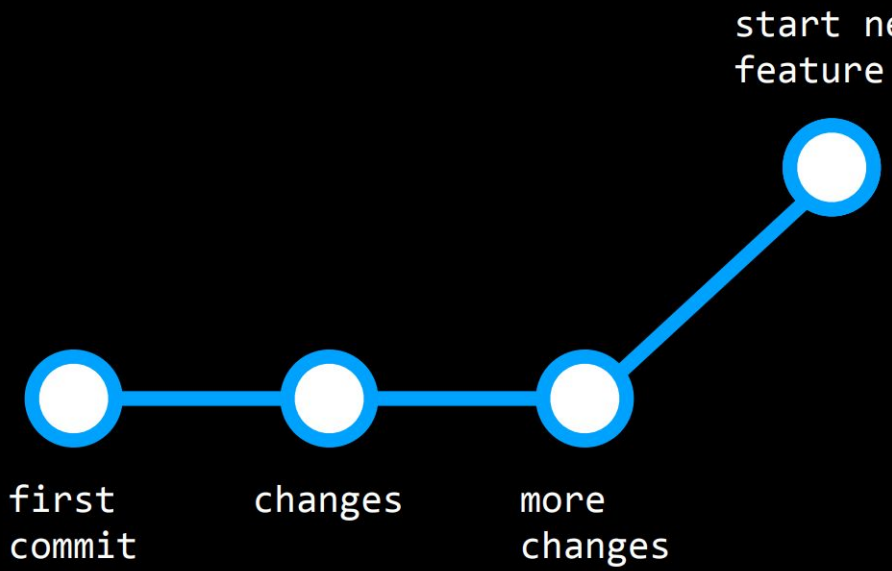
first
commit

https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf

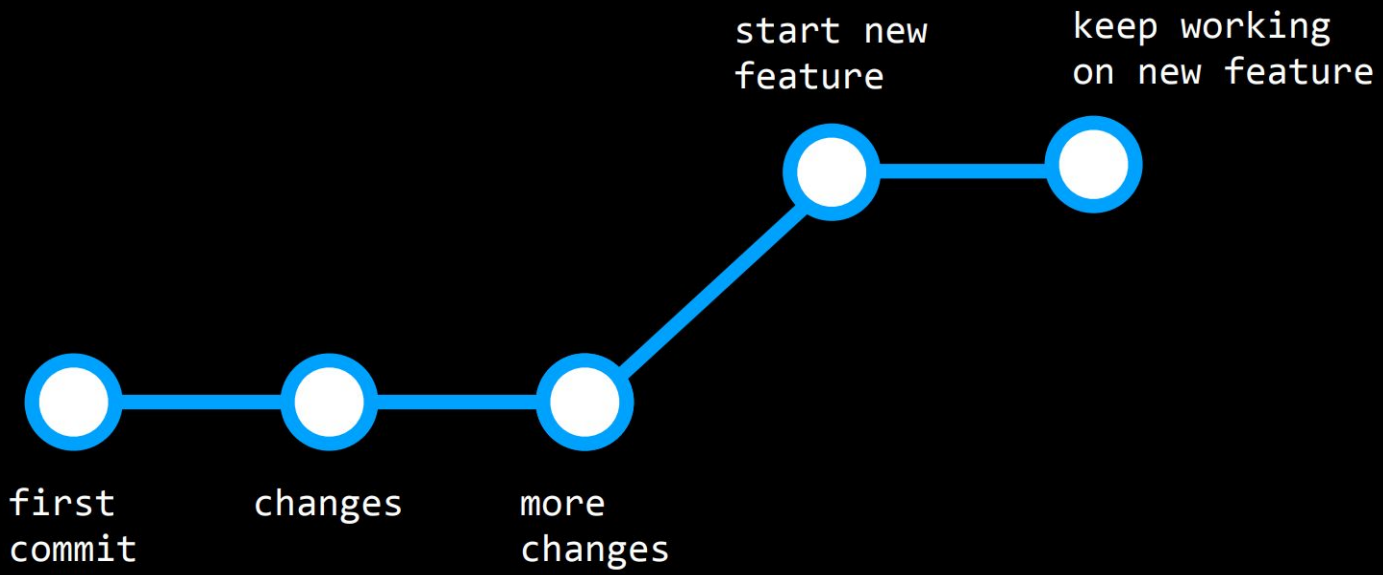


https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf

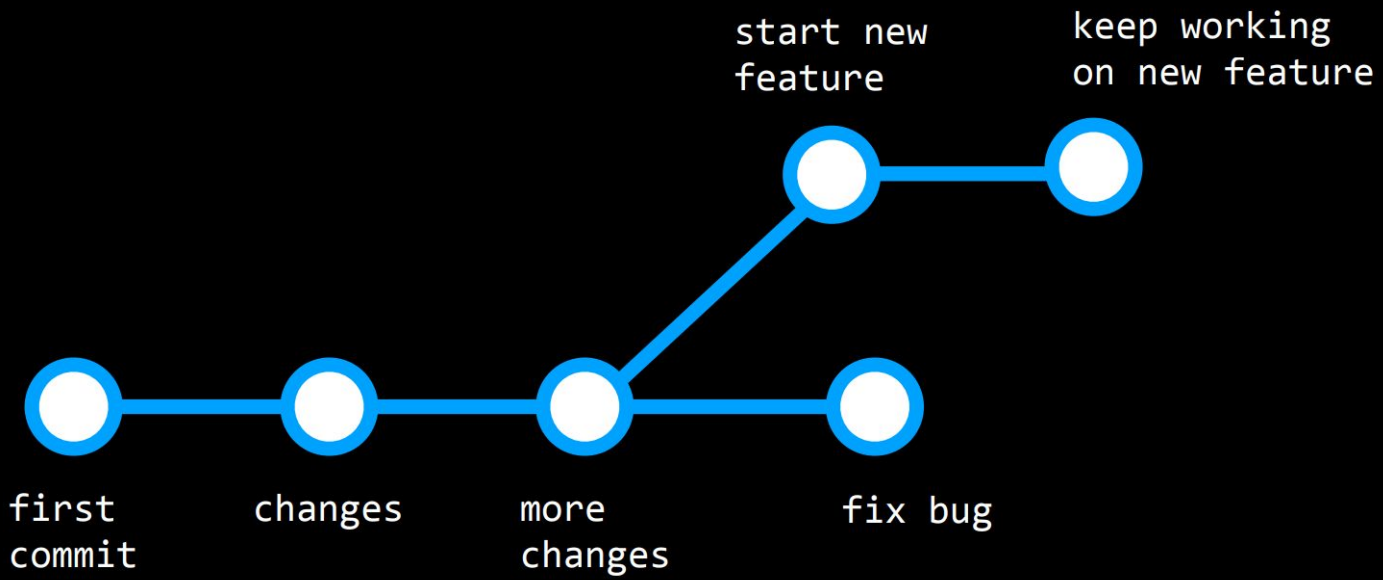
https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



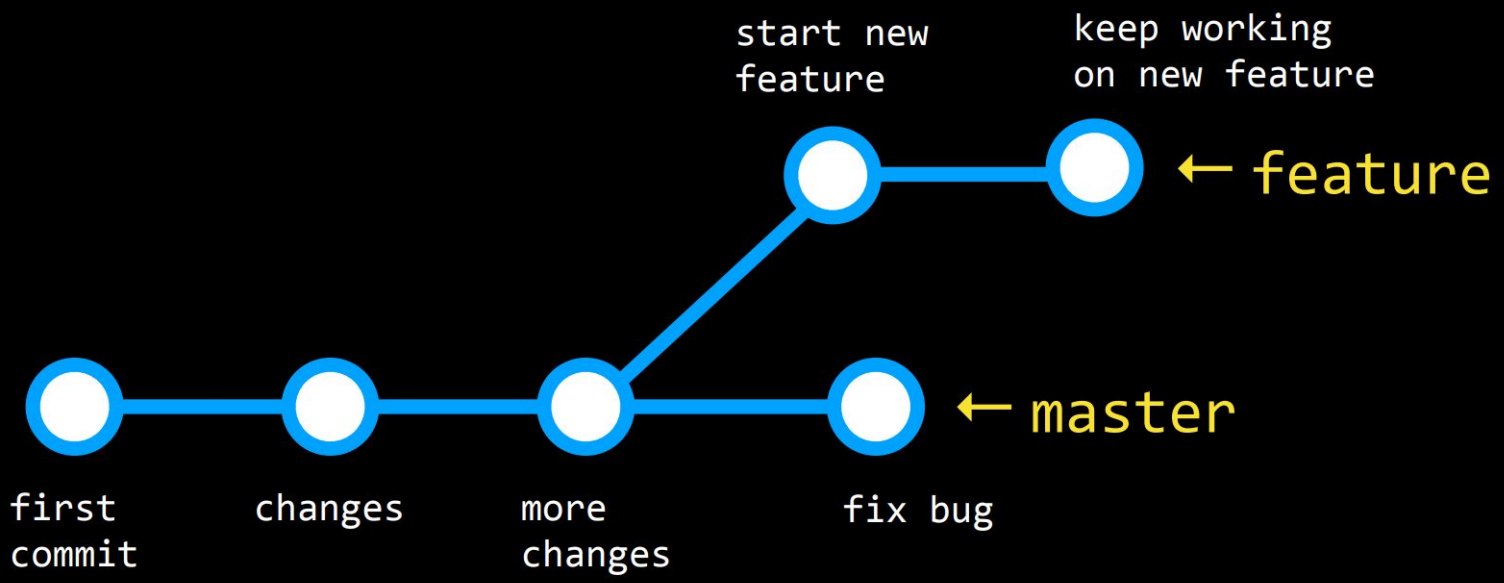
https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



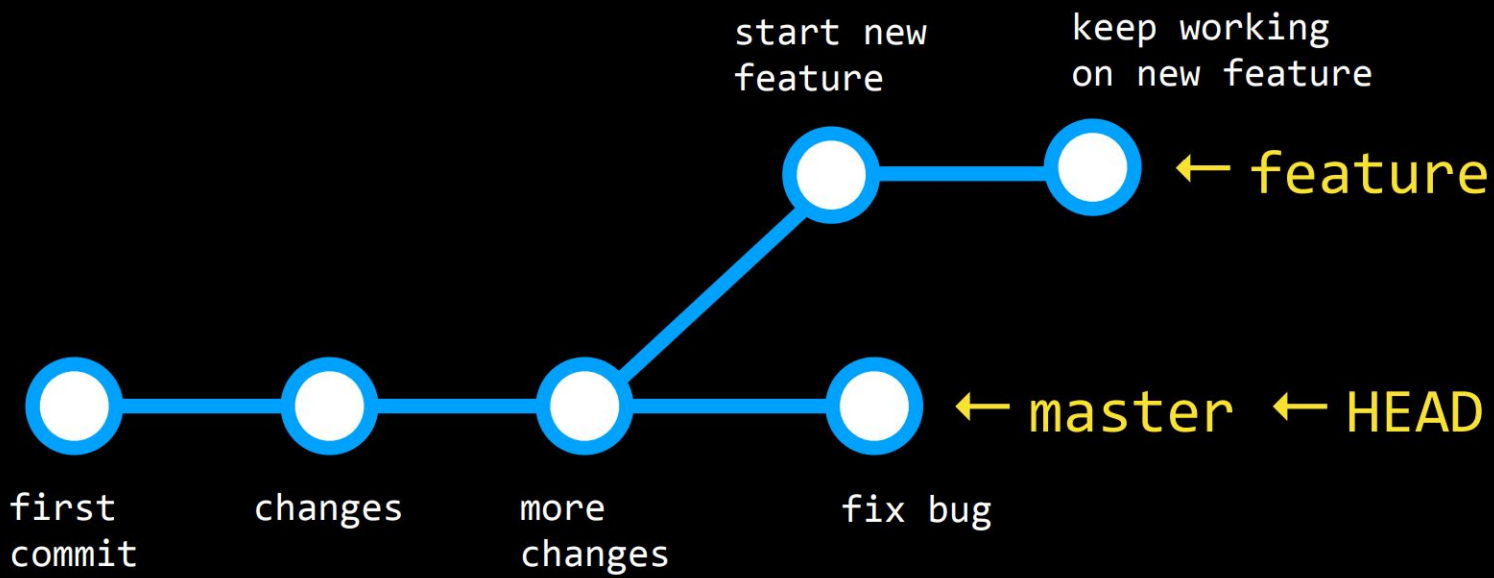
https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



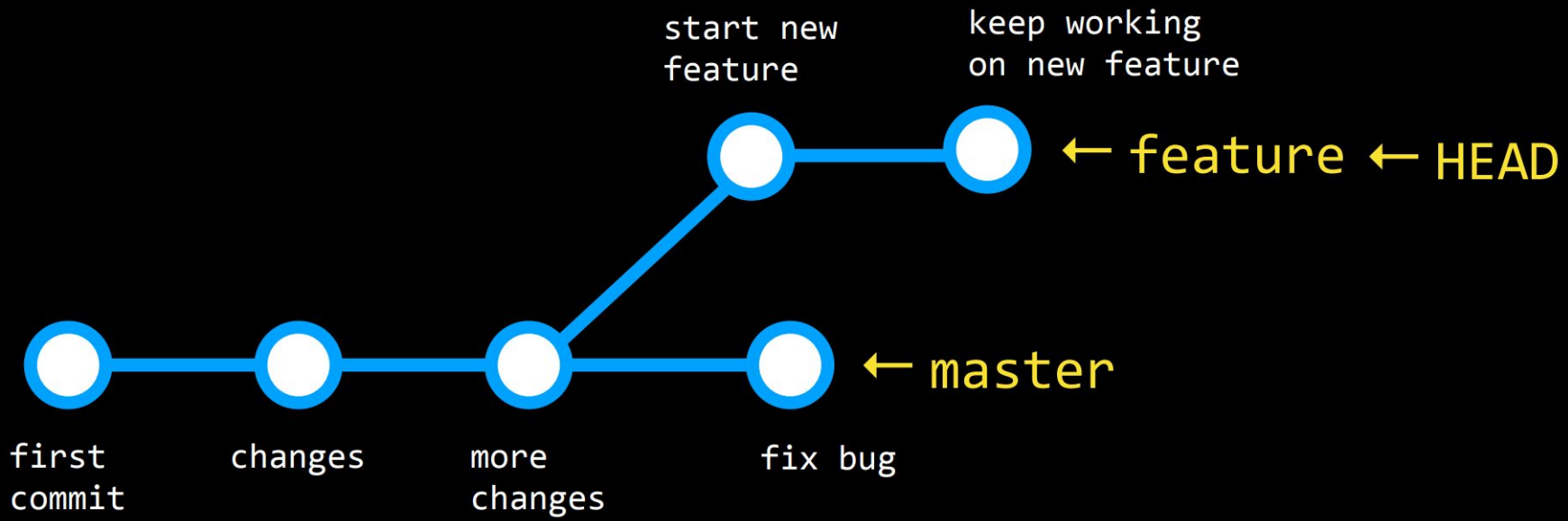
https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



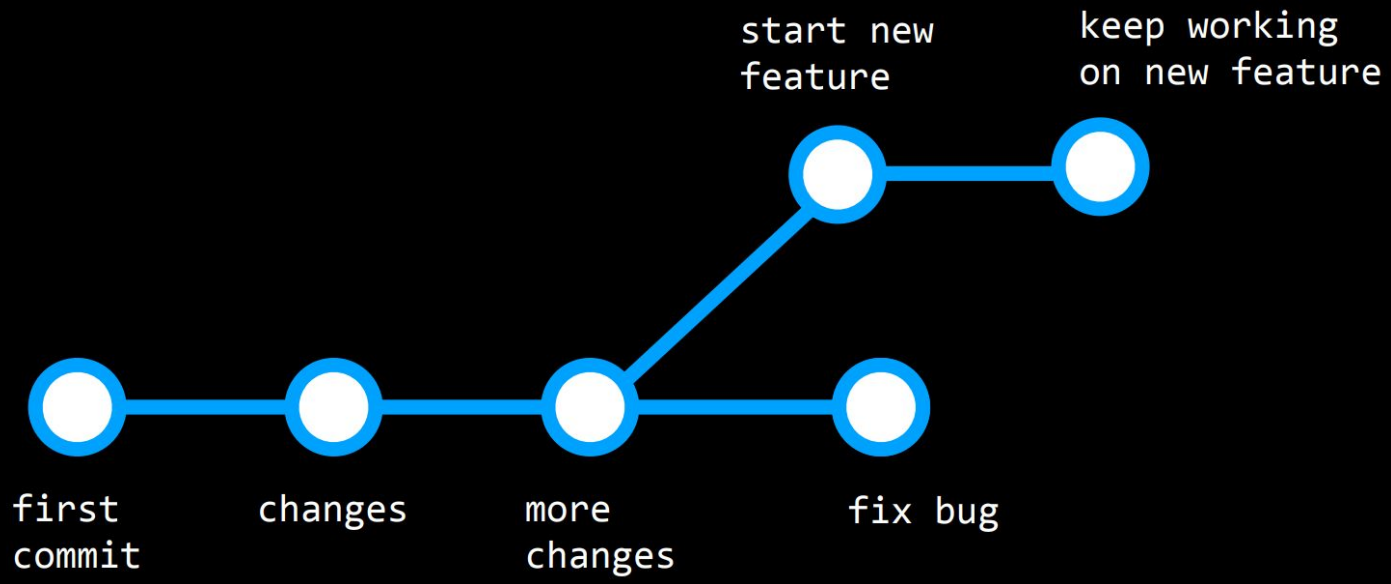
https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



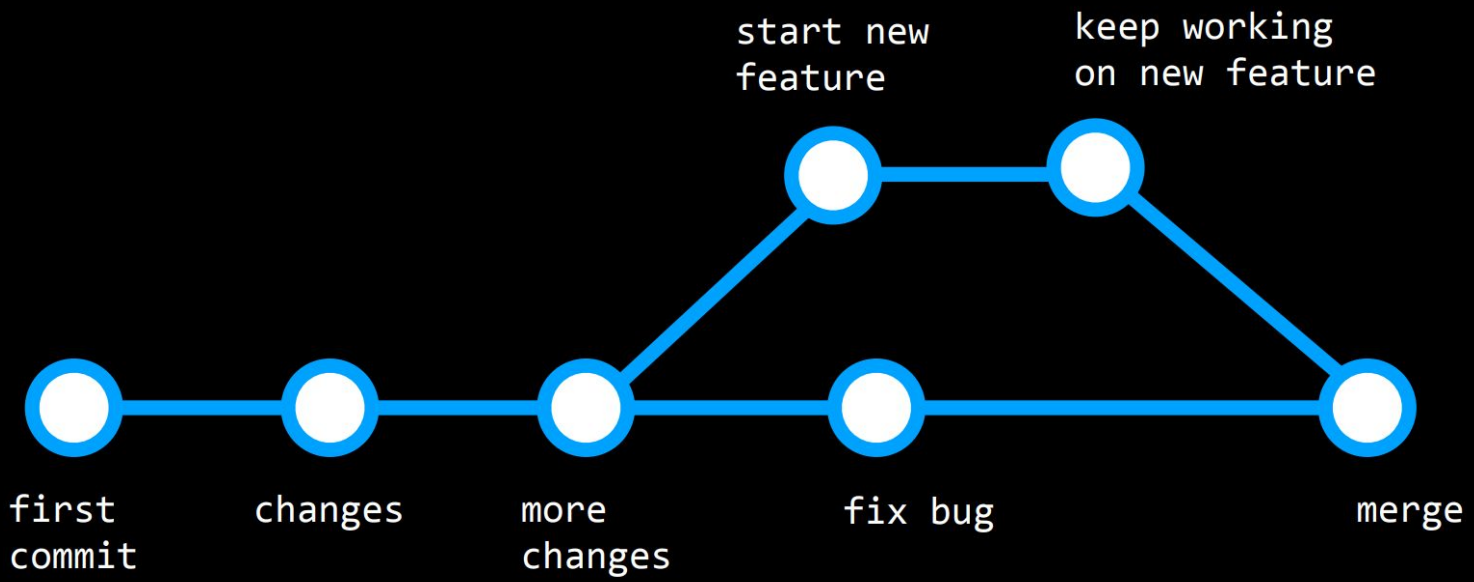
https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf



https://cs50.harvard.edu/web/2018/spring/resources/1/lecture1.pdf

`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?