Git:
Version control system
a.k.a. allows users to collaboratively work on code
A few things to start:

**local**: code on your machine (i.e. your version of the codebase, your changes that you made)

**remote**: code not on your machine (i.e. the production codebase in Github)

**repository**: a project/set of files

**working directory**: the folder you’re working out of (i.e. /home/Desktop/myRepository)

**staging area**: a place for files getting ready to be pushed out (like a loading dock)
git pull
git pull:

- fetch and pull files from a remote repository

- updates files in local repository
git add
git add . or git add `<filename>`:

- add file and its changes to the staging area
- files that you change are *not* automatically staged
git commit
git commit -m "<new message>":

- collect the files in the staging area
- records the changes to the repository like a snapshot
Working directory

Git add operation

Staging area

Git commit operation

Git repository
git push
git push:

- upload local repository content to remote repository
- i.e. I’ve made changes and I want my partner to see them now!
git reset
git reset <filename>:

- Unstage file(s)

- i.e. I’ve done “git add” on file(s) that I don’t actually want to stage!
git reset HEAD~1:

- Remove the commit before

- `git reset HEAD~num` removes `num` commits before
git revert
git revert <commit ID>:

- Undo the commit from commit ID, as if it didn’t happen, but keep a record of it.
- This *creates a commit*

- i.e. I messed up, but don’t want to just delete the commit from the history. Instead, I want the log to show that I undid a commit.
git checkout <file>
git checkout `<filename>`:

- Change a file to the version on the remote repository
- i.e. I want the version of a file on the remote repository. You can think of this as an “undo” operation on your changes.
- Note: git checkout has different meanings depending on if you give it a filename, branch, commit, etc. We are only talking about `filename` for right now.
git status
git status:

- Shows state of the working directory and staging area
git log
git log:
- Shows your commit history
1. Repository
2. Clone operation
3. Working copy
   - Edit, Add, Move files
4. Modify working copy
5. Status and Diff operation
6. Review changes
7. Commit and Push operation
8. Commit changes
9. Amend and Push operation
10. Fix mistakes
11. Push operation
12. Push operation
Demo
Next week: Advanced Git
Merging, branching, cherry picking, rebasing, etc