INTRO TO GIT

Digital Innovation Group
http://diging.asu.edu/

VERSION CONTROL

- Keep and manage different versions
  - Keep a history of changes
  - Revert changes if necessary
  - Keep track of who changed what when
- Possibly in the cloud

VERSION CONTROL SYSTEMS

- CVS
- Subversion
- Git

DISTRIBUTED VERSION CONTROL

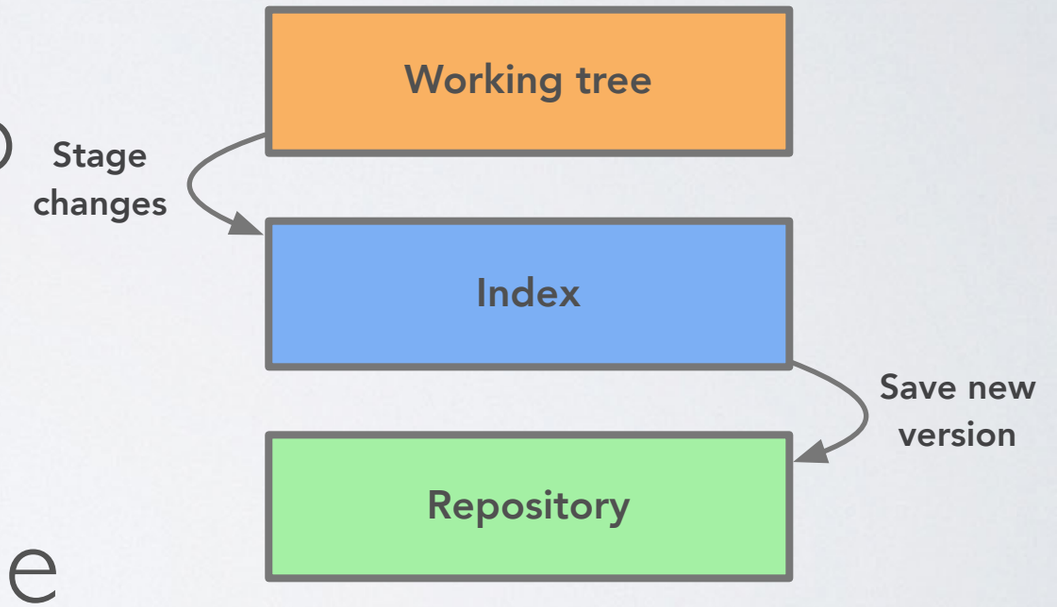
- Complete history on each computer
- Peer-to-peer not client-server
- Multiple copies means multiple backups

GIT'S DIFFERENT PARTS

- Working tree
- Index
- Repository

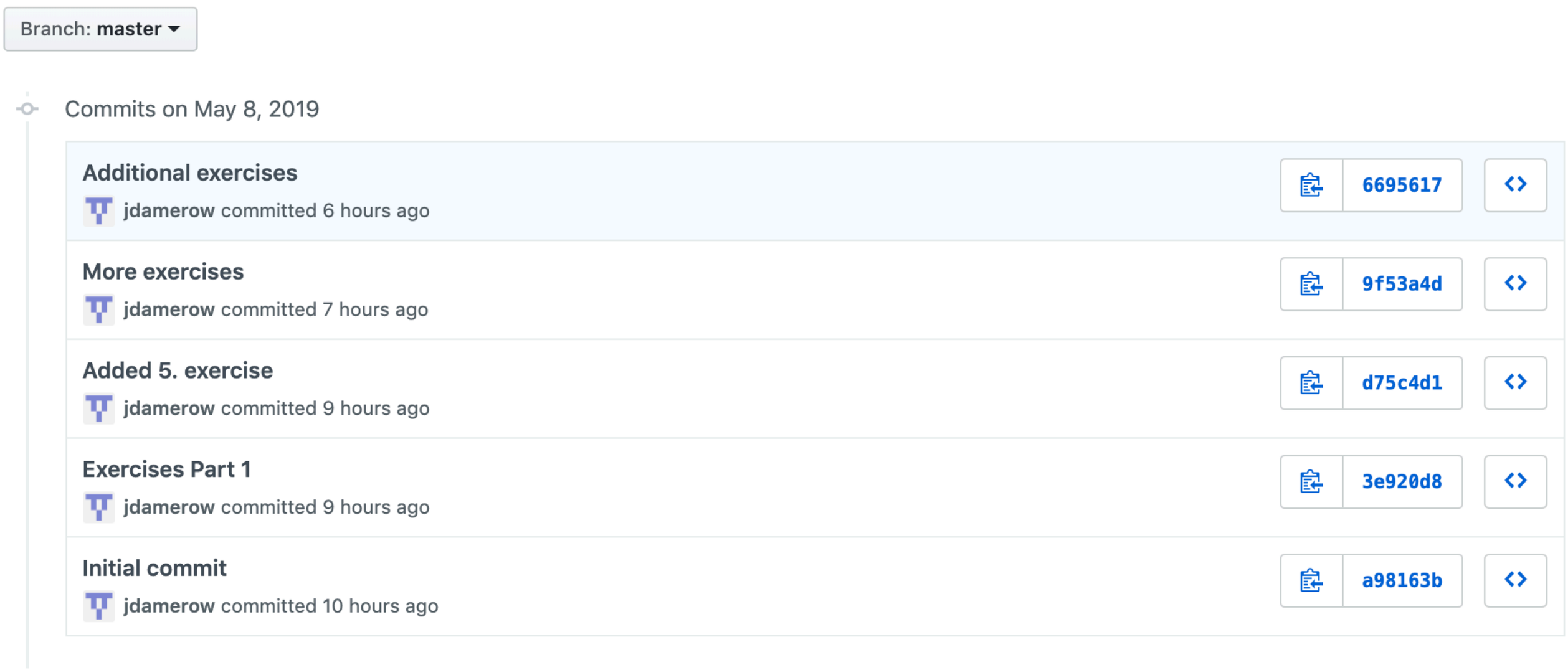
GIT WORKFLOW

- Modify a file (or multiple ones)
- Stage changes to be added to the repository
- Commit staged changes to the repository

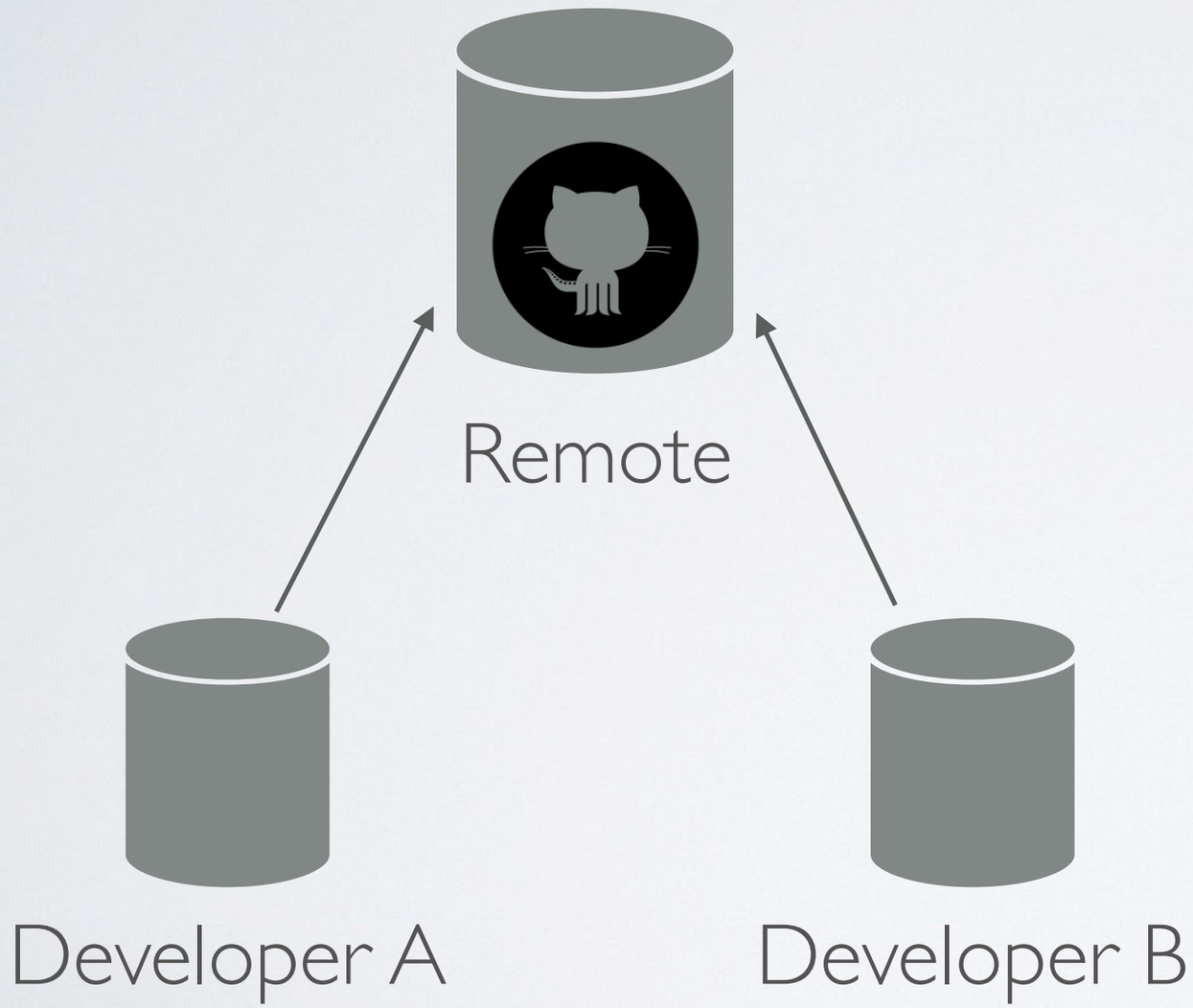


COMMITS

- "Saved versions" of your files

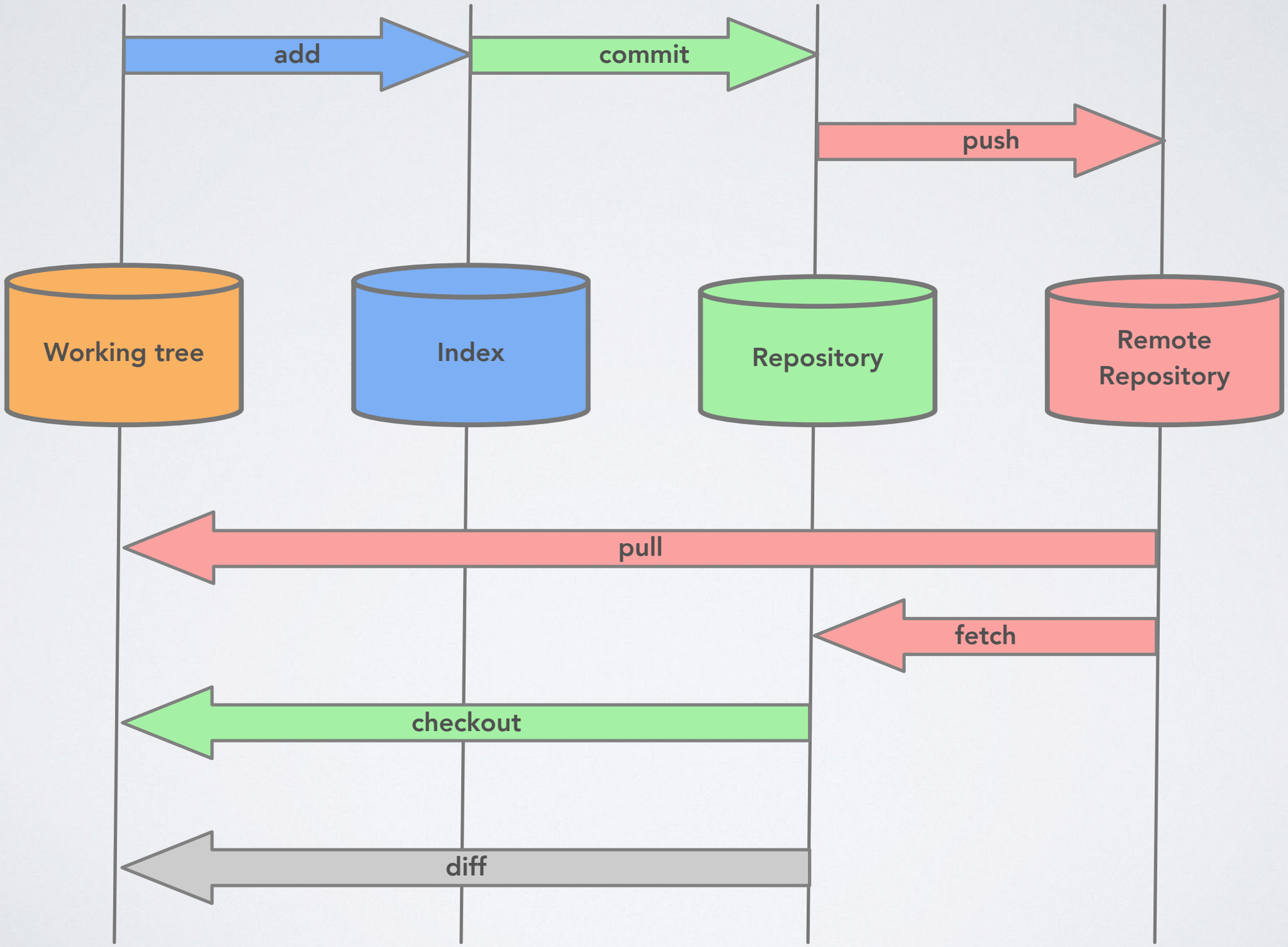


SHARE CODE



- Code is pushed from local to remote
- And pulled from remote to local

GIT STEPS


add
commit
push
Working tree
Index
Repository
Remote Repository
pull
fetch
checkout
diff

HOW TO USE GIT

```
git [command] [args]
```

e.g.:

```
git add myfile.txt

git commit -m “my new file.”

git push
```

LET'S GIT!

https://www.katacoda.com/courses/git

Scenario 1

GIT EXERCISE

- Create a new folder
- Init a new repository
  `git init`
- Add a file with some content
- Add your file to the staging area
  `git add <filename>`
- Commit your changes
  `git commit –m “message”`
- Create a repository on GitHub
- Add the new repository as remote to your local repository
  `git remote add origin https://github.com/user/repo.git`
- Push your changes
  `git push –u origin master (you only need “–u origin master” the first time you push to a repository)`
- Make changes to the file through the GitHub webpage.
- Pull your changes to your local copy
  `git pull`
- Add a second file with some content
- Stage, commit, and push your changes
  `git add, git commit, git push`