What is Git?

Coding Boot Camp

Module 01

Git is a control system.

It is great at storing and managing files within a project and moving files around the internet.

Git is a version control system.

It is great at keeping track of file changes and attributing those changes to developers.
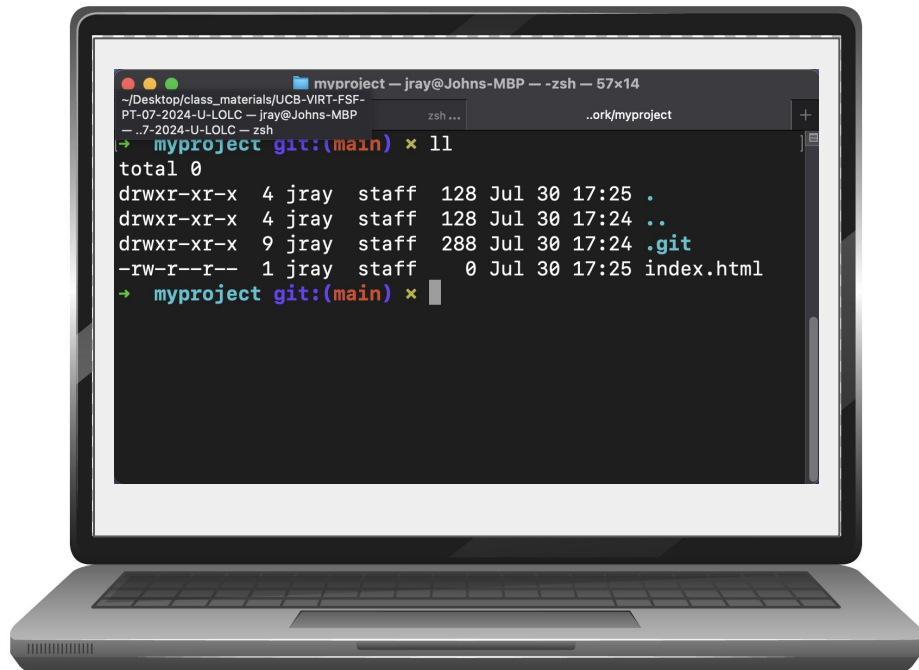
Git is a distributed version control system.

It is great at synchronizing file changes with project clones hosted remotely.

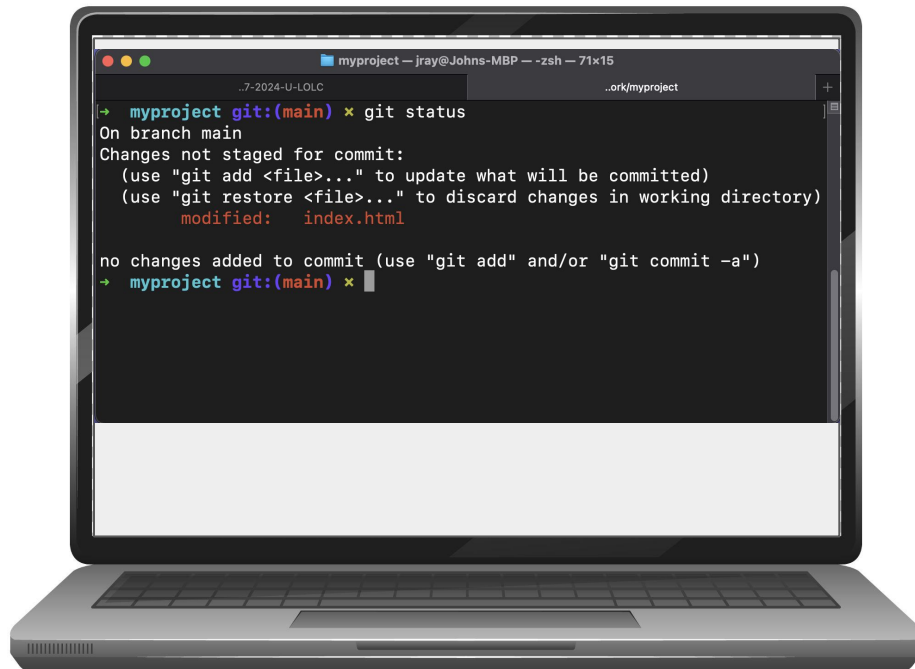# My local machine runs the git application

`git` can control a folder on my local machine


```
🔴🟡🟢        📁 myproject — jray@Johns-MBP — -zsh — 57×14
~/Desktop/class_materials/UCB-VIRT-FSF-
PT-07-2024-U-LOLC — jray@Johns-MBP        zsh ...              ..ork/myproject        +
— ..7-2024-U-LOLC — zsh
→  myproject git:(main) × ll
total 0
drwxr-xr-x  4 jray  staff  128 Jul 30 17:25 .
drwxr-xr-x  4 jray  staff  128 Jul 30 17:24 ..
drwxr-xr-x  9 jray  staff  288 Jul 30 17:24 .git
-rw-r--r--  1 jray  staff    0 Jul 30 17:25 index.html
→  myproject git:(main) ×
```
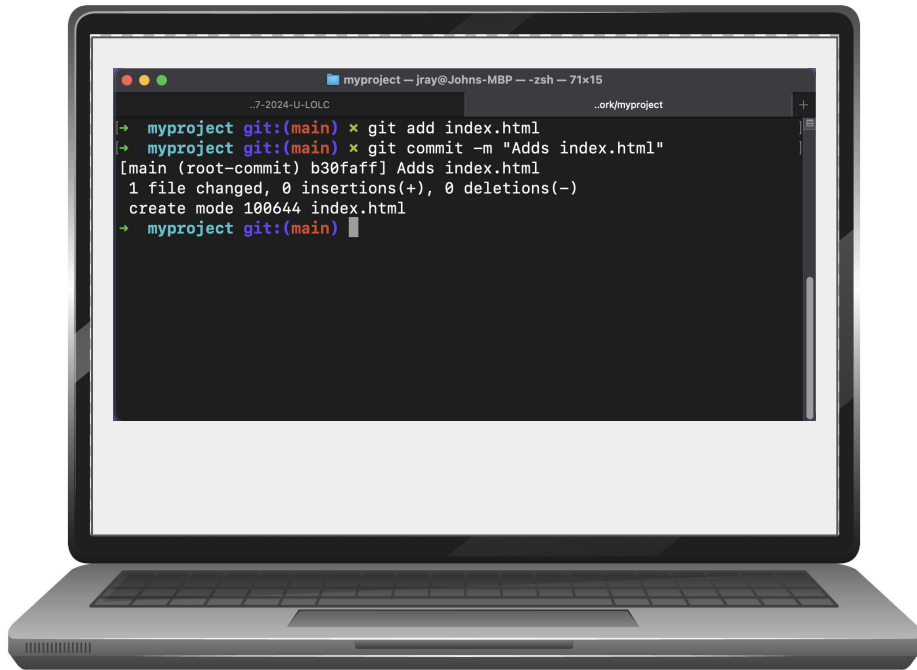
When I tell `git` to control a folder, `git` knows when a file has been added to the folder, removed from the folder, or changed in anyway.

`git` can keep track of changes over time


```
                 myproject — jray@Johns-MBP — -zsh — 71×15
   ...7-2024-U-LOLC                              ..ork/myproject          +
→  myproject git:(main) × git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
→  myproject git:(main) × █
```

Git will notice any change to any file. **Git is interested in changes within the folder**

`git` can keep track of changes over time


```
● ● ●            📁 myproject — jray@Johns-MBP — -zsh — 71×15
       ..7-2024-U-LOLC                              ..ork/myproject         +
→  myproject git:(main) × git add index.html
→  myproject git:(main) × git commit -m "Adds index.html"
[main (root-commit) b30faff] Adds index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
→  myproject git:(main)
```

Using `git` we can **add** and **commit** changes to our project. This allows us to go back to older version of our project.

A change can be a new file, removing a file, or making a change to a file's contents.

`git` can keep track of changes over time

```
→ myproject git:(main) × git add —A
→ myproject git:(main) × git commit —m "updates the index.html"
[main 58da240] updates the index.html
 1 file changed, 1 insertion(+)
```
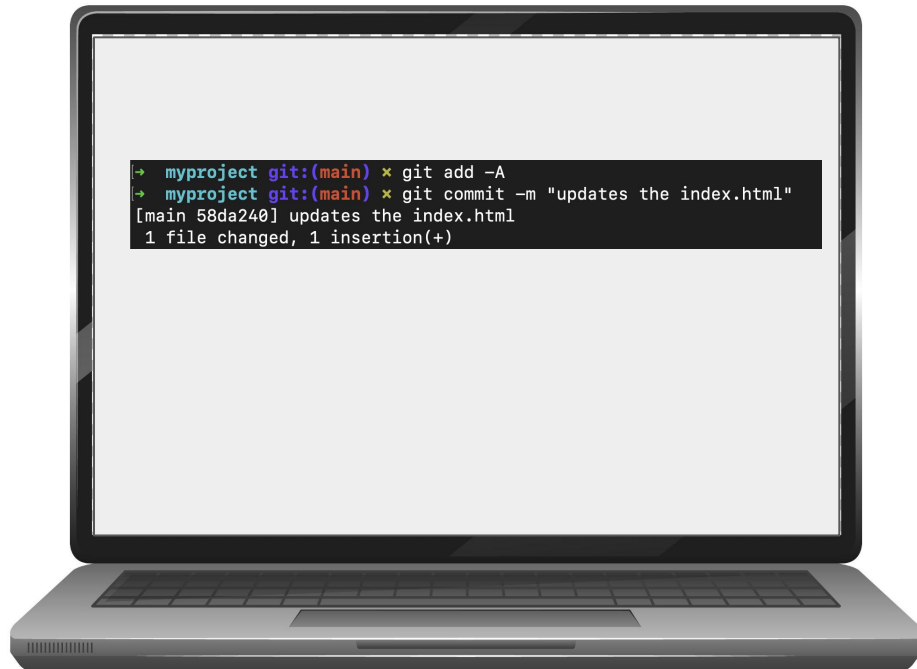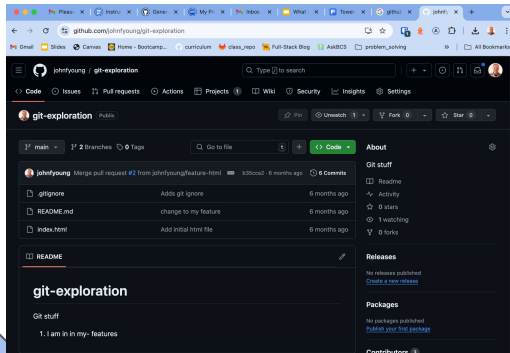
It's up to us which changes we want to keep. We instruct git to **add** changes to a **commit**.
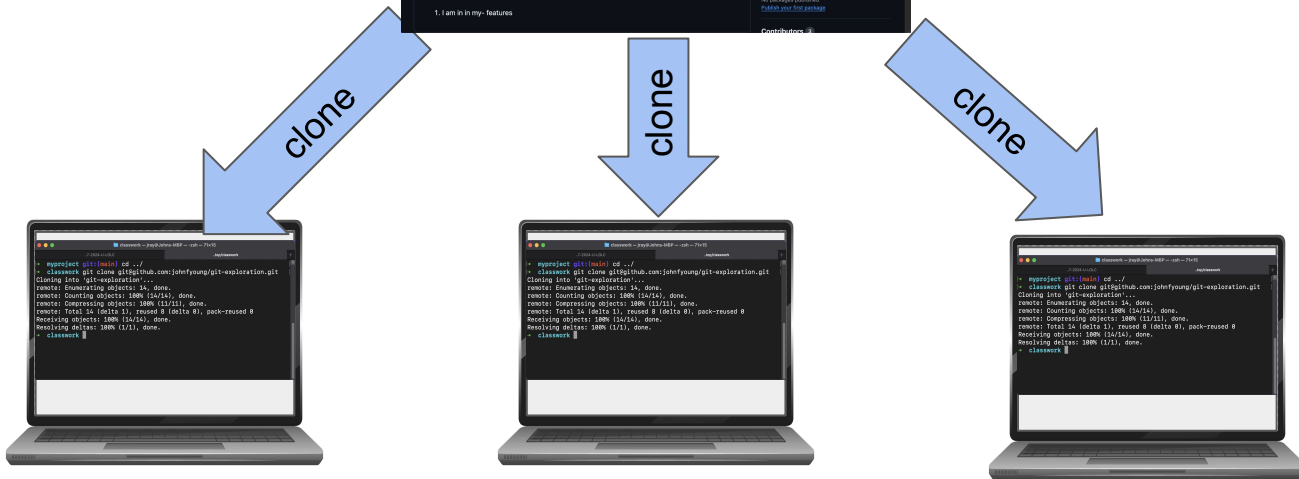
git can synchronize with remotely hosted clones
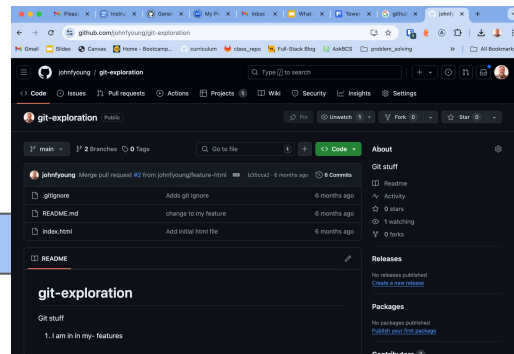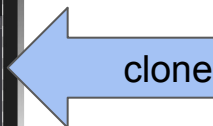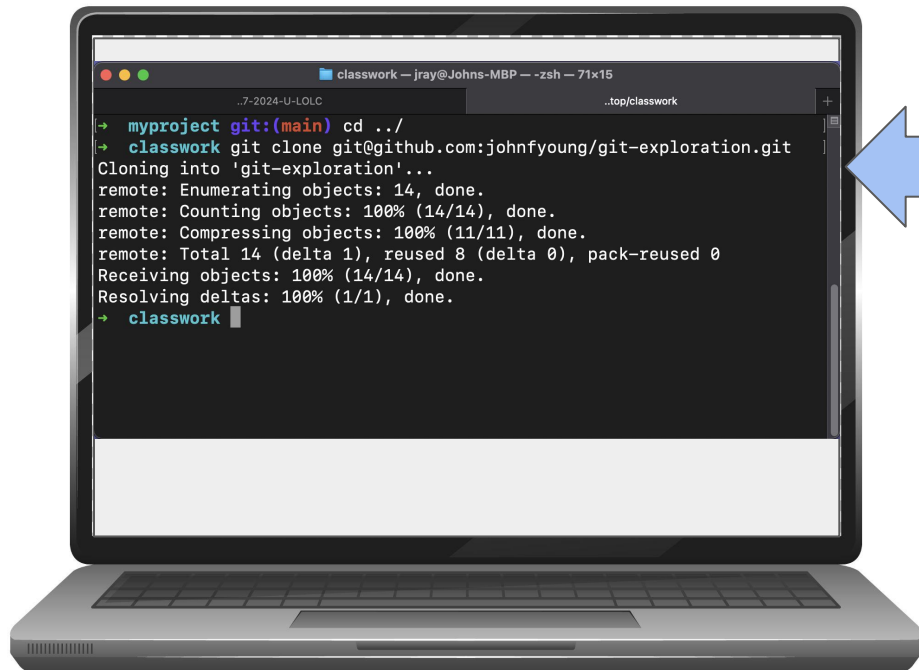
clone

Instead of starting a project locally, we can **clone** one from a remote git host, like Github. Github also runs the git application and hosts git projects to the internet
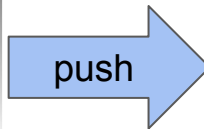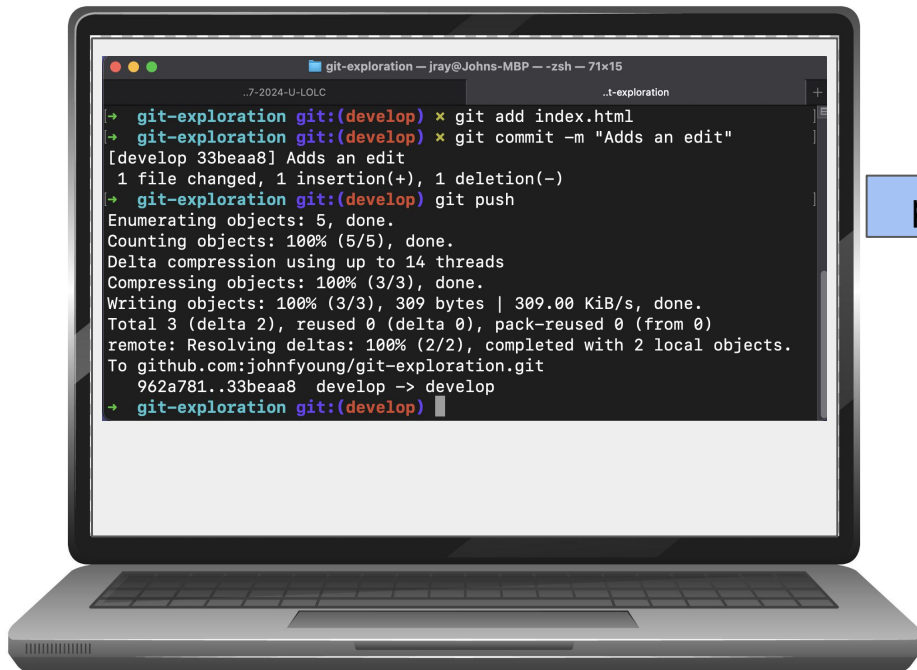
git can synchronize with remotely hosted clones

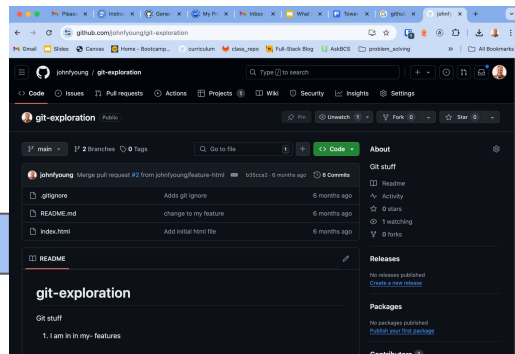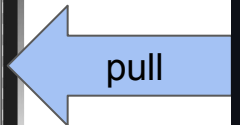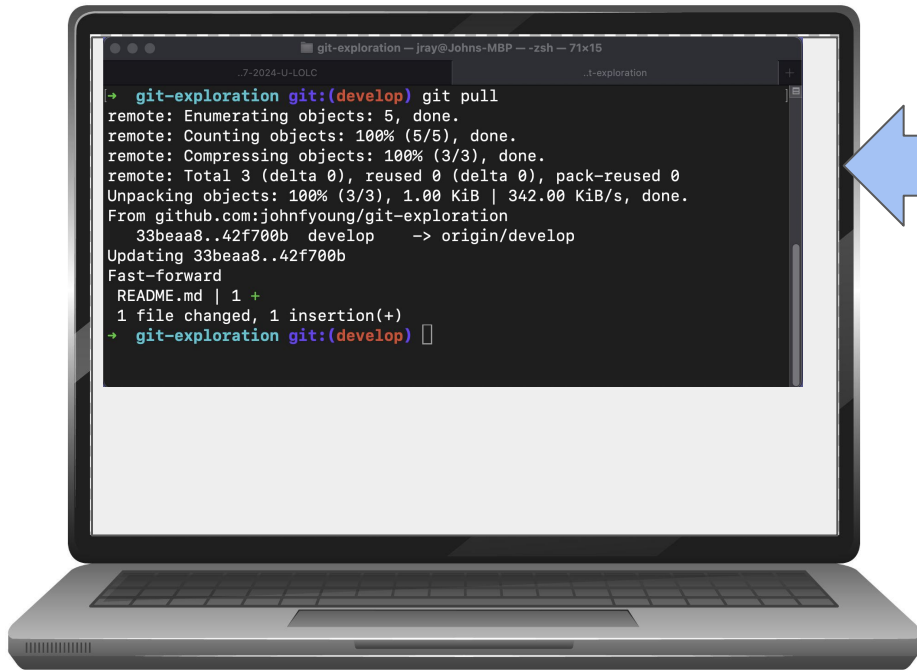My teammates can also clone this repo to their local machines

git can synchronize with remotely hosted clones
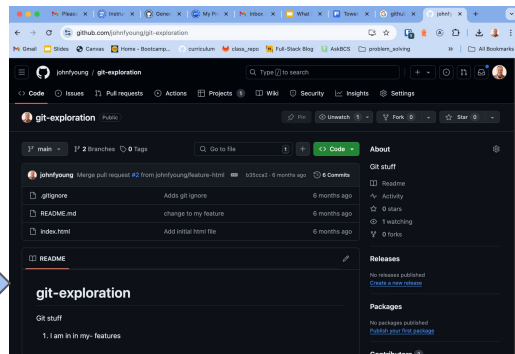
push

I can make changes to my local project clone and **push** the changes to to the remote clone.

git can synchronize with remotely hosted clones

pull

I can also **pull** changes added to the remote clone by my teammates.