git

Agenda:

Install Git

```
$ brew install git
```

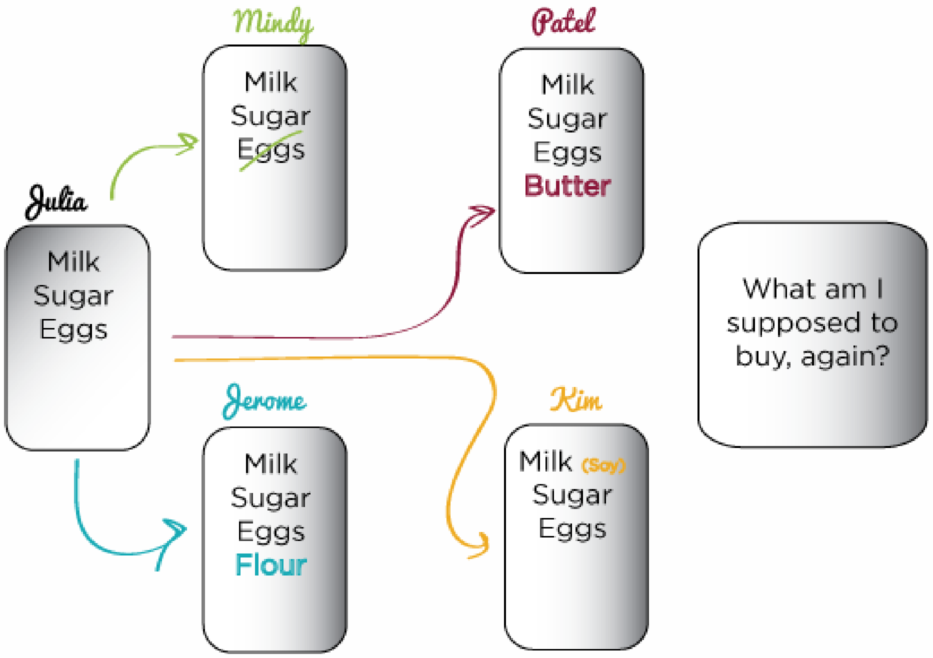
What is Version Control?

- Track History to see your changes over time and revert if mistakes were made
- Branch into different versions to work in parallel
- Collaborate with others to create anything from small websites to large programs

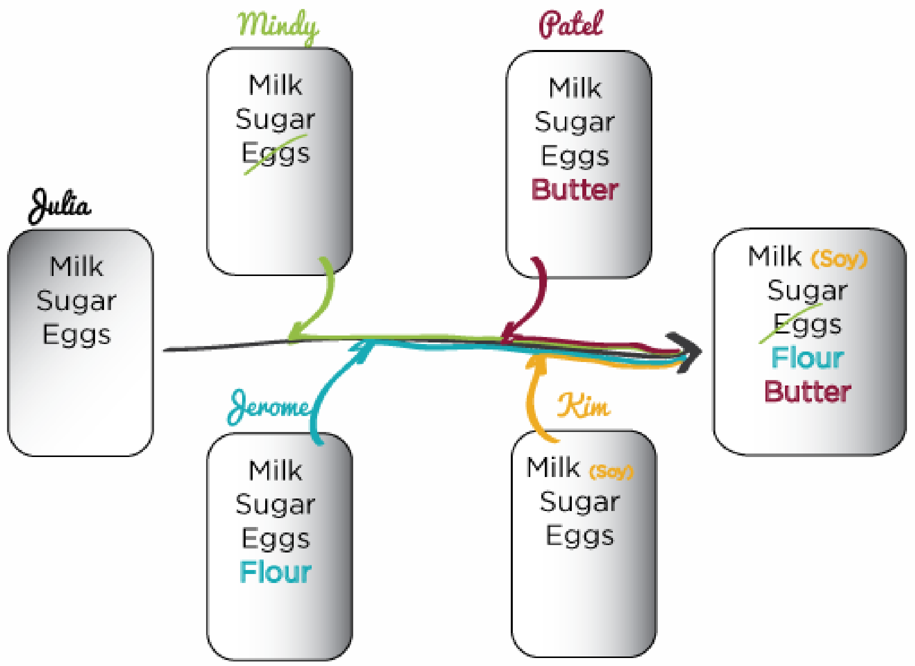
Have you ever lost a paper and had to start over?

Have you ever worked on something for a while and wished you could undo back to a certain point in time?

How did your last group project go?



Julia
Milk
Sugar
Eggs
Mindy
Milk
Sugar
~~Eggs~~
Patel
Milk
Sugar
Eggs
Butter
Jerome
Milk
Sugar
Eggs
Flour
Kim
Milk (Soy)
Sugar
Eggs
What am I supposed to buy, again?


Mindy
Milk
Sugar
~~Eggs~~
Patel
Milk
Sugar
Eggs
Butter
Julia
Milk
Sugar
Eggs
Milk (Soy)
Sugar
~~Eggs~~
Flour
Butter
Jerome
Milk
Sugar
Eggs
Flour
Kim
Milk (Soy)
Sugar
Eggs

git

Not

GitHub

Short History

1. Linux operating system created in 1991 by Linus Torvalds
2. Changes were passed around as patch files over email
3. Switched to a proprietary DVCS in 2002 called BitKeeper
4. Had to start paying for it 2005, so they ditched it and built their own

How does it work?

Generic Workflow

1. Create a Repository for your project
2. Add files for git to track them
3. Commit your files when you're done working on them
4. Make Branches for experimental work
5. Checkout commits or branches to look at different versions of your code

Team Workflow

1. Share your Repository
2. Push your commits so others can get your code
3. Pull any new commits created by your team so you can update your code with new changes
4. Resolve Merge Conflicts that might happen when changes happen at the same time

First Things First

Git should be installed by now, so now we configure it.

```
$ git --version
git version 2.23.0

$ git config --global user.name "Your Name"
$ git config --global user.email your@email.com
```

Config

- A place to store your settings
- System, User, and Local settings
- /etc/gitconfig < ~/.gitconfig < .git/config

```
$ git config -l
$ git config core.editor
```

Init

Create a Repository

```
$ cd ~
$ mkdir my-repo
$ cd my-repo
$ git init
```

You only need to do this once.

Status

See the three states of your files

```
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add"
```

You'll be typing this all the time.

Make Changes

```
$ echo "Hello World" > fileA.txt
$ git status
...
Untracked files:
  (use "git add <file>..." to include in what will be c

        fileA.txt
```

Add

Move changes to Staging Area

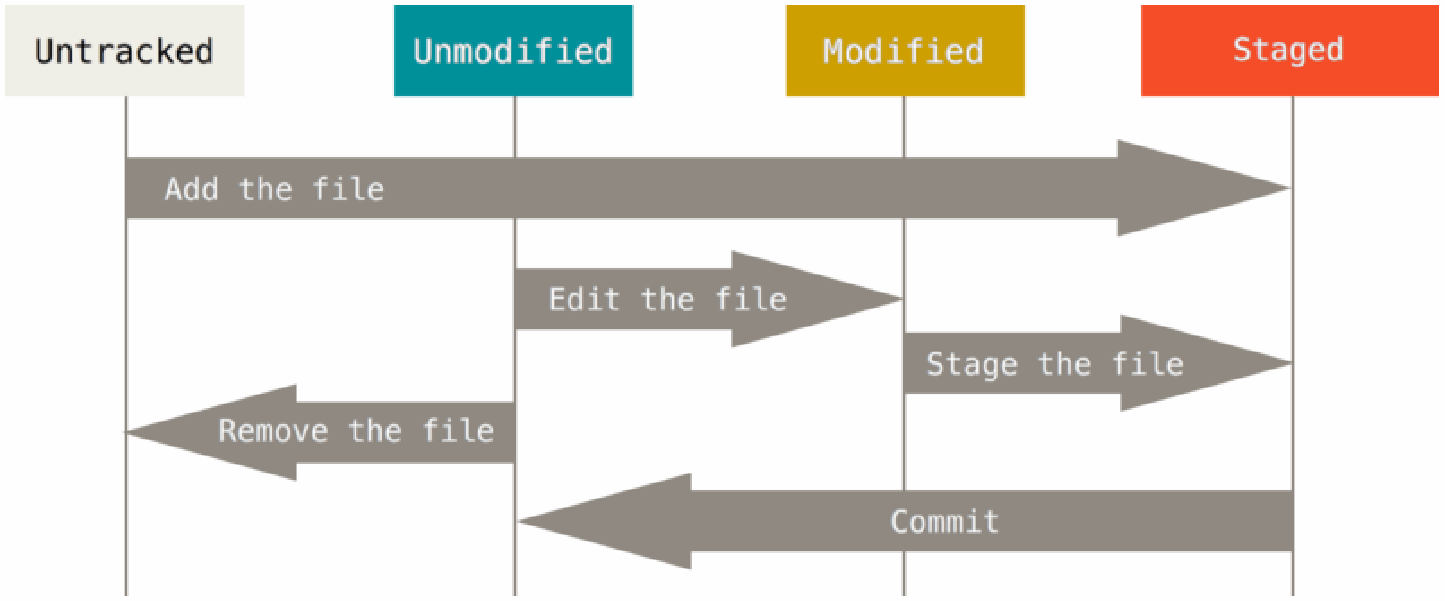
```
$ git add fileA.txt
$ git status
...
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

          new file:   fileA.txt
```

Commit

Save a snapshot of the repository

```
$ git commit
$ git status
On branch master
nothing to commit, working tree clean
```

File Lifecycle



Diff

See differences between working directory and the commited repository

```
$ echo "goodbye" >> fileA.txt
$ git diff
$ git status
```

Save Modifications

```
$ git add fileA.txt
$ git status
$ git commit -m "Changed file A"
$ git status
```

Log

See the repo history

```
$ git log
```

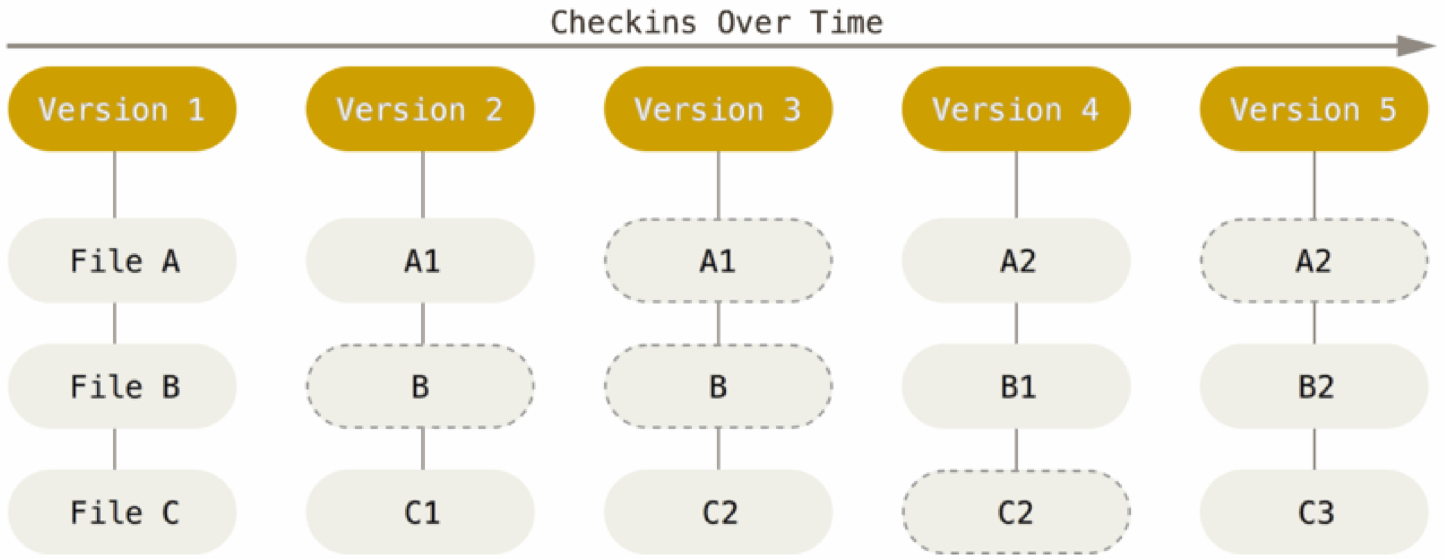
Clone

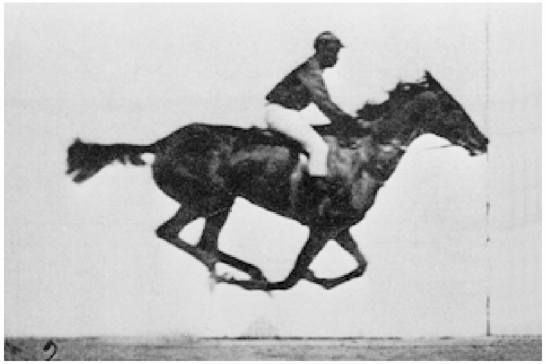
Copy a remote repository

```
$ cd ~
$ git clone https://github.com/ts-cset/cset-105.git
$ cd cset-105
```

Practice

Codecademy - Learn Git


Checkins Over Time
Version 1
Version 2
Version 3
Version 4
Version 5
File A
File B
File C
A1
B
C1
A1
B
C2
A2
B1
C2
A2
B2
C3



Git Philosophy

- Streams of Data
- Everything is Local
- Data Integrity

SHA-1 Hash

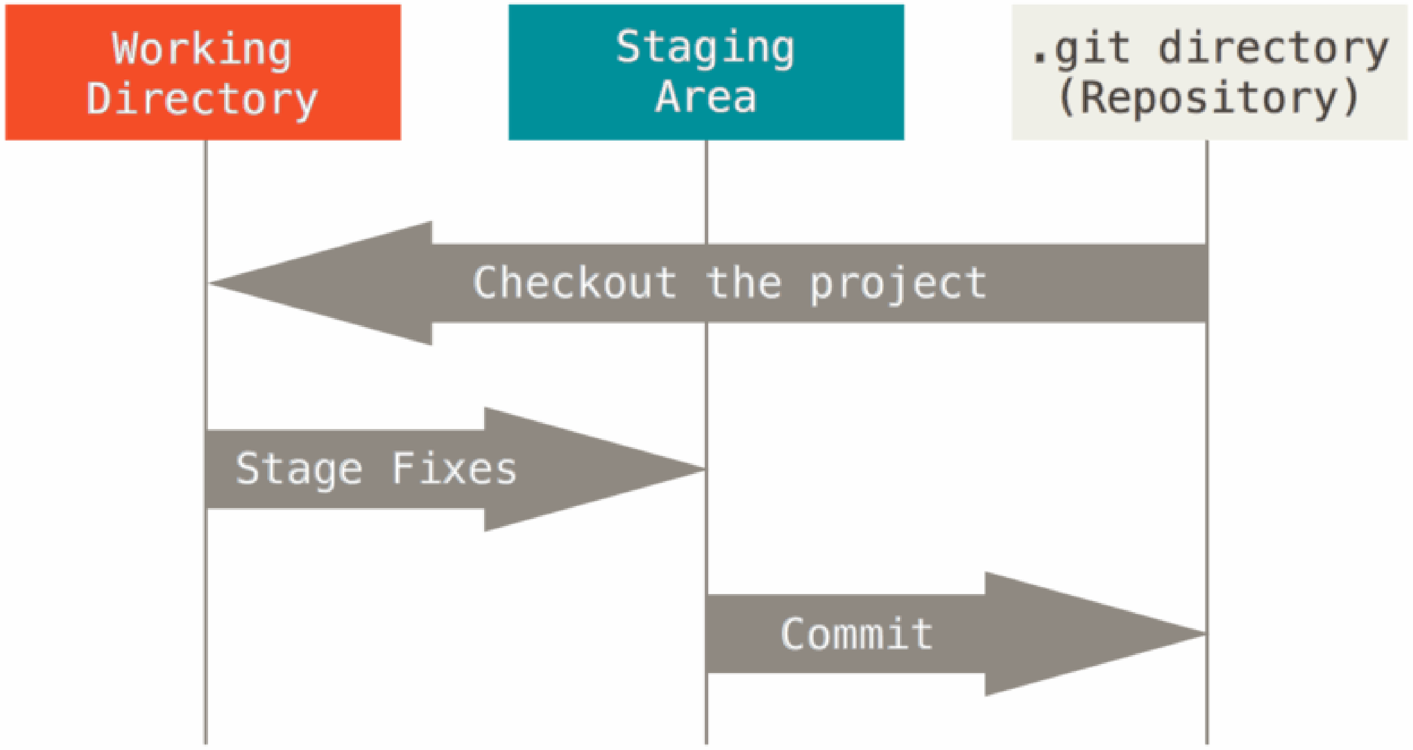
```
24b9da6552252987aa493b52f8696cd6d3b00373
```

The Three States

- Commited: the file is stored safely in git's database
- Modified: the file has changed from the last commit
- Staged: the modified file has been marked to be included in the next commit snapshot

The Three Sections

- Git Directory: holds the project's metadata and history
- Working Directory: a single snapshot of one version of the project, pulled from the database for you to use/modify
- Staging Area: a single file in the git directory that stores info about what is going into your next commit


Working Directory
Staging Area
.git directory (Repository)
Checkout the project
Stage Fixes
Commit

More Practice

- Official Book - Pro Git
- Atlassian Tutorial - Getting Started
- Github Guide
- Github Practice - Git-it

Next Week

- Pushing and Pulling remotes
- Branching and Merging
- Forking and Pull Requests
- Undoing Things
- How to use Github