Welcome!

Agenda

- Git
- GitHub

Git

History of Git

- Made in 2005 by Linus Torvalds
- Before that, he made the Linux Kernel
    - Here is a <u>Ted Talk</u>
    - Here is his <u>GitHub</u>
    - Here is the <u>source code for Git</u>

Warning

- Git and GitHub can be tough to get used to
- They take a long time to get comfortable with as well
- Most explanations get very technical very quickly:
  - So focus on the concepts (as always)

What does Git do?

- A version control system (or VCS)
  - It takes snapshots of our projects
  - Gives us a project-wide undo button!
- A collaboration tool
  - It merges differences in our code for us
- A local development tool
  - Supports non-linear development

What does Git do?

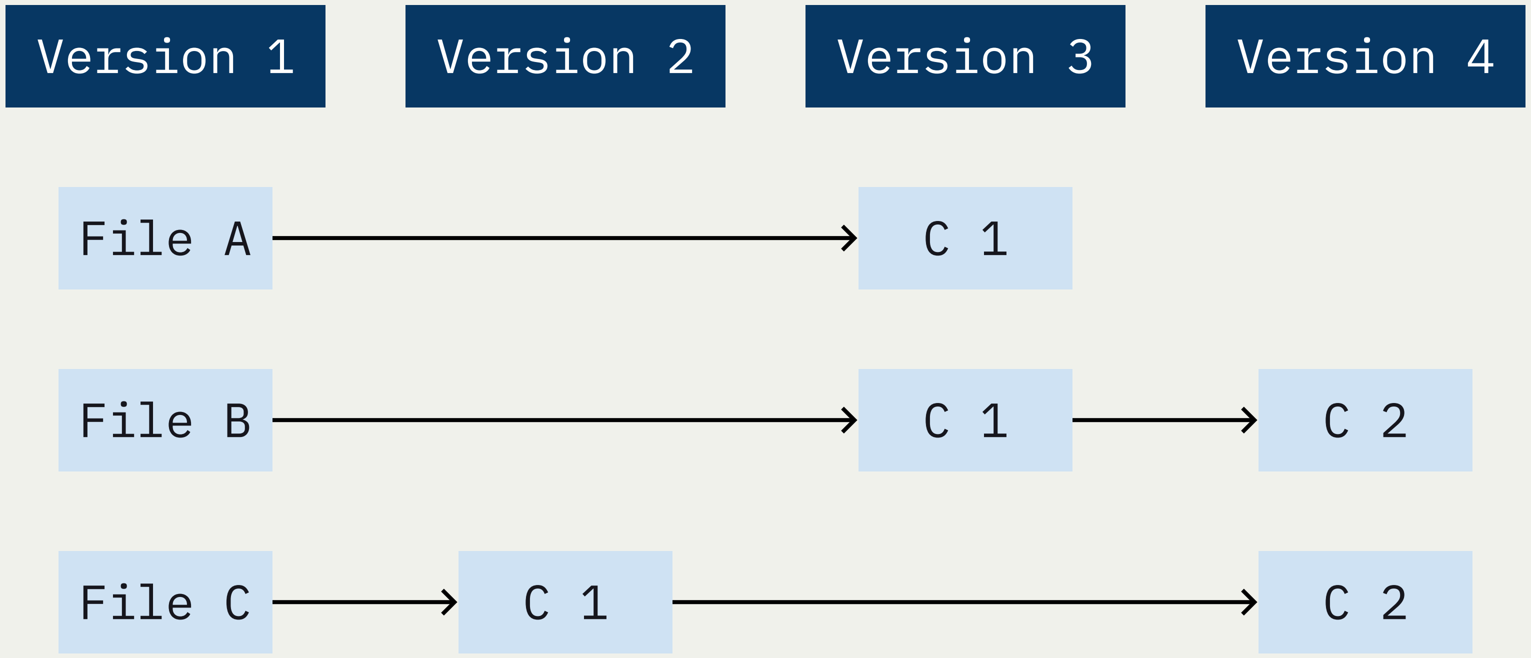
It's a tool for modern-day teamwork

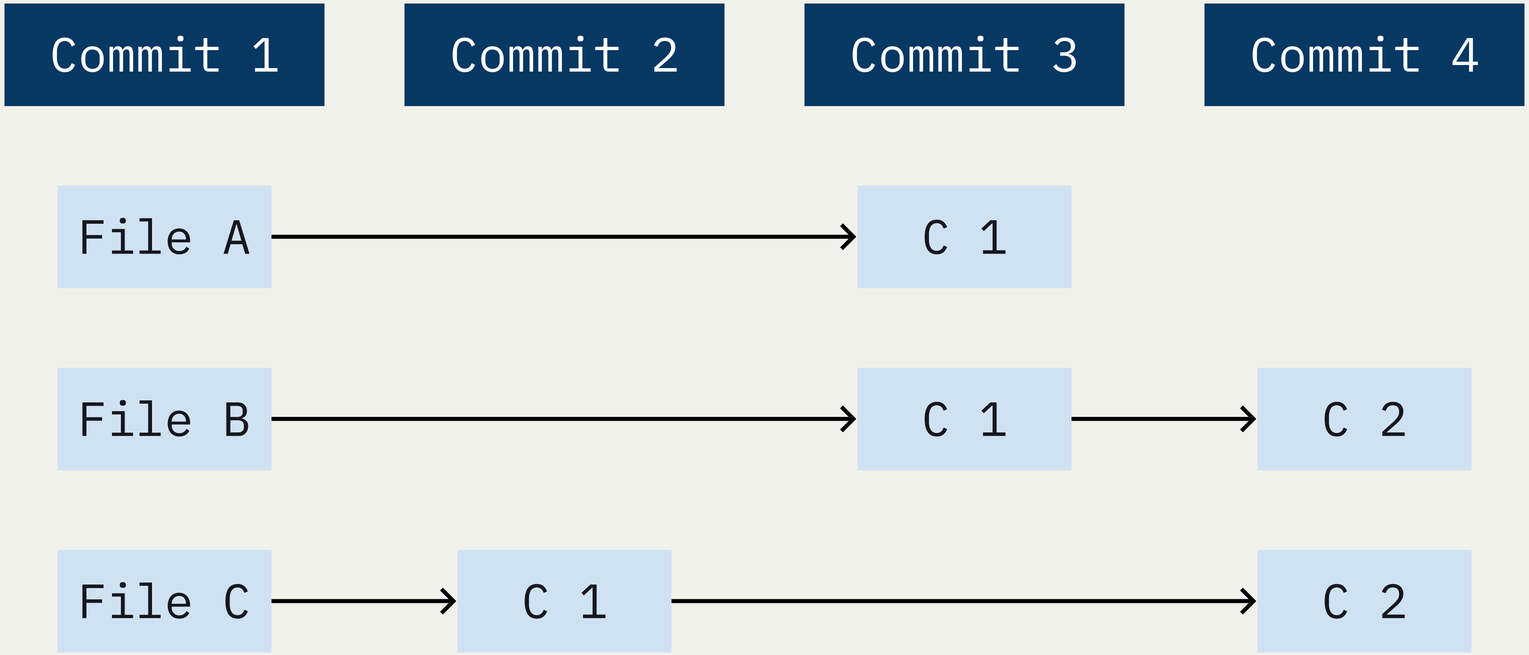
For people who are working asynchronously, on a shared body of work

It saves us from moving floppy disks around, or saving lots of copies of the

one file.

The more people there are on a project, the more likely you are to use it (though I use it all the time, for anything)


Version 1
Version 2
Version 3
Version 4
File A
C 1
File B
C 1
C 2
File C
C 1
C 2


Commit 1
Commit 2
Commit 3
Commit 4
File A
C 1
File B
C 1
C 2
File C
C 1
C 2

Why use Git?

- You make a change and want to take it back?
    - *Git can undo it*
- You want to find where everything went wrong?
    - *Git will show you*
- You want to try out a new feature that will potentially cause problems?
    - *Git can protect you*
- You want to work with a bunch of people?
    - *Git will make that easier*

Terminology

- **Repository**
    - A project
- **Add**
    - Tell Git to pay attention to a file(s)
- **Commit**
    - Tell Git to take a snapshot of a file(s)
- **Origin**
    - A place where your code is stored

Terminology

- **Push**
  - Put all the code up online (normally using GitHub)
- **Branch**
  - A version of your project
- **Clone**
  - When you copy a project (normally from GitHub) to your computer

Terminology

- **Fork**
  - Your copy of someone else's GitHub repository
- **Merge Conflict**
  - What happens when two pieces of code can't be automatically merged
  - You need to decide what you want
- **Pull Request**
  - When you request to have a project include your code

How do we use Git?

- The Command Line
- Applications
  - GitHub Desktop
  - SourceTree
  - GitKraken
  - Plus more...

Sign up for a GitHub account!

Let's set up our Git

```
git config --global user.email "YOUR GITHUB EMAIL"

git config --global user.name "YOUR GITHUB NAME"

git config --global color.ui "auto"

git config --global core.editor "code --wait"
```

Run these commands one at a time in your terminal, making sure to add your GitHub email and GitHub name in the quotes for the first two

Some useful commands

```
git init
```

Creates a new local **repository**

Some useful commands

```
git status
```

Shows what is currently happening with your repository

Some useful commands

```
git add README.md

git add .
```

Makes Git watch certain files. The `.` means to watch everything in the current directory

Some useful commands

```
git commit -m "A commit message"
```

Takes a snapshot of all added files

Some useful commands

```
git log
```

Shows the previous commits

Resources

- Atlassian: Learn Git
- Official GitHub Git Tutorial
- CodeSchool
- Code Academy
- Git & GitHub for Poets
- Git For Humans

GitHub

What is GitHub?

- It is a website that hosts Git repositories
- Helps with collaboration
- It is a Graphical User Interface (GUI)
- Helps us perform common tasks
- The Dropbox or Google Drive for code

Why do we use it?

- To share our code with other computers
- For collaboration (Pull Requests, Forks etc.)
- It acts as a portfolio
- To visualise what is going on
- As a project management tool (Projects)
- An error reporting system (Issues)
- Documentation (Wiki)
- Free Hosting (GitHub Pages)
- It is the Industry Standard

How do we use it?

Once you have a local Git repository...

- Create a repository on GitHub
- We need to tell Git where the code should be stored
    - `git remote add origin URL`
- We need to push (or upload) all of the code
    - `git push origin master`

How do we use it?

Once you have a GitHub repository...

- We need to pull (or download) all of the code
  - `git pull origin master`
- We can also clone a repository
  - `git clone URL`

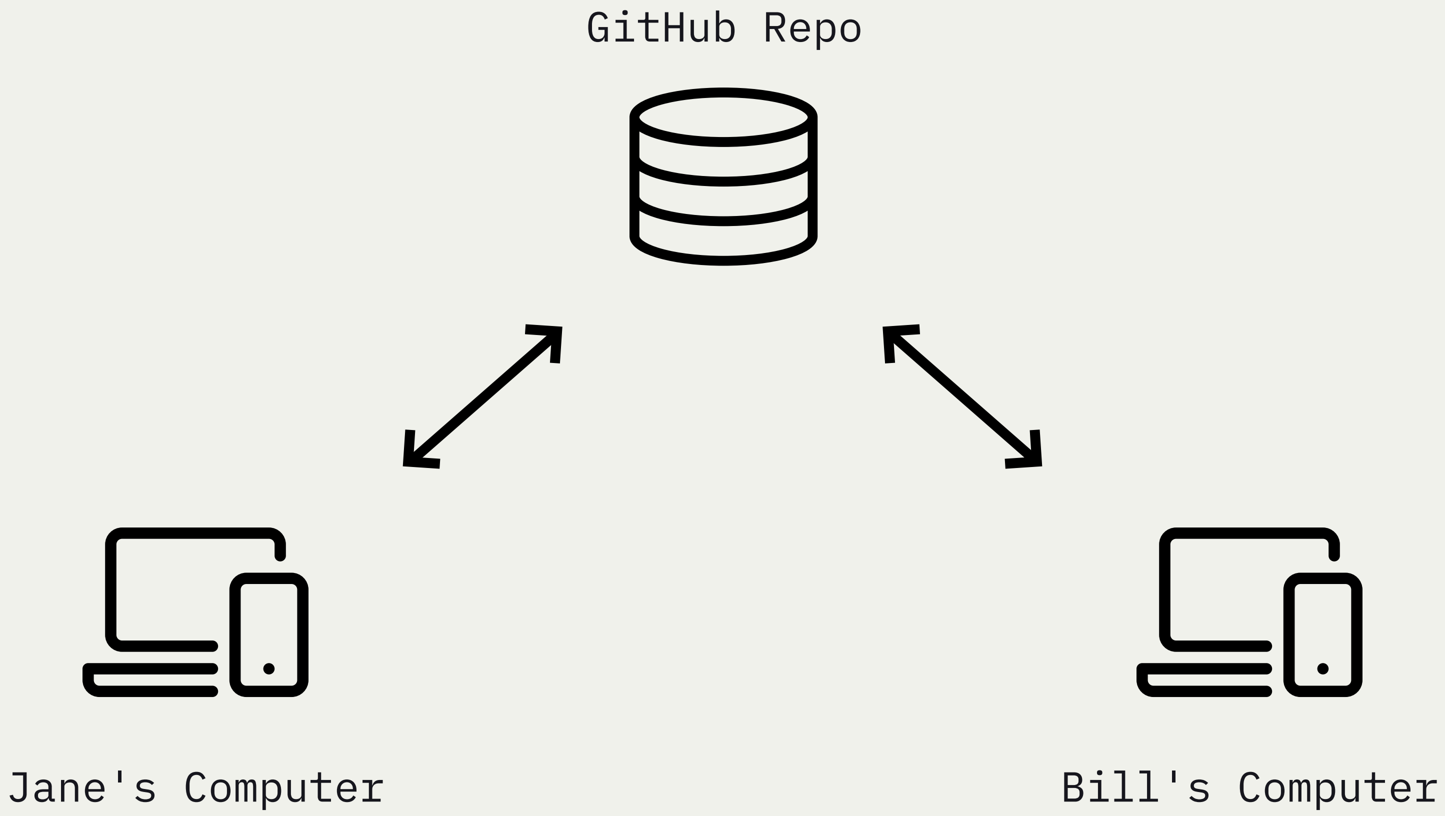
A Typical Upload Workflow

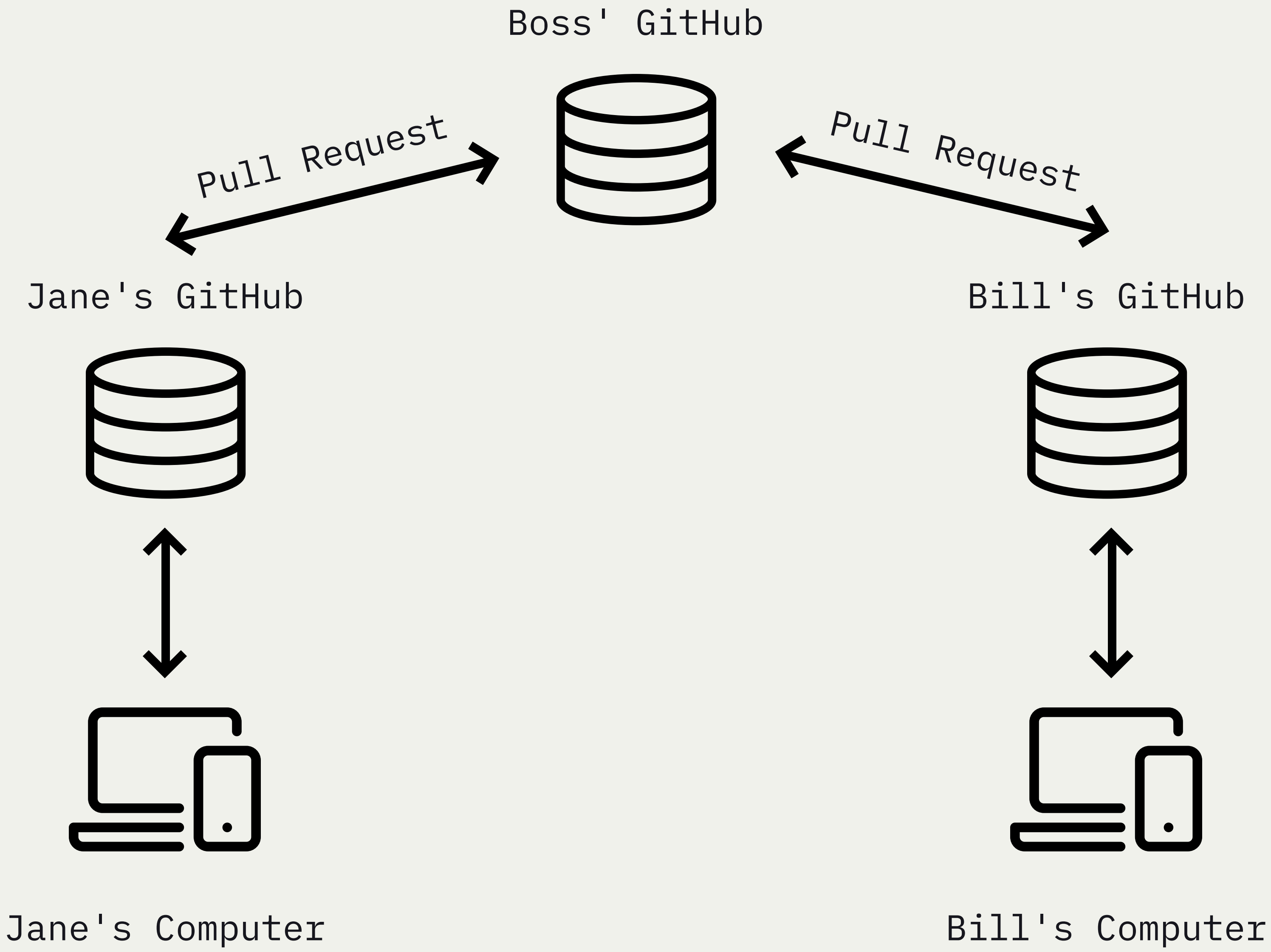
```
git add .

git commit -m "Made changes"

git push origin main
```

Collaboration Approaches


GitHub Repo
Jane's Computer
Bill's Computer


Boss' GitHub
Pull Request
Pull Request
Jane's GitHub
Bill's GitHub
Jane's Computer
Bill's Computer

Review

Homework

- Add homework to a GitHub repository!
- Do more with APIs!
- Track the International Space Station's Position
  - Using this API
  - Show the current location of it in your HTML
  - Bonus: Link it up to the Google Maps API!
- Or anything other API you want
- *Note: Don't use ones requiring OAuth*
  - We will be going through this later

What's next?

- More APIs
- More AJAX

Thank you!