IN4MATX 285:
Interactive Technology Studio

Practice: Version Control

Today’s goals

By the end of today, you should be able to…

- Describe what version control is, and the basics of how it works
- Explain why developers usually use version control in their projects
- Use Git and GitHub to do basic version control functionality

Ever done something like this?

One solution: version control

Version control

- Enable collaboration between many developers
- Recover files or revert to a previous state
- Identify who made modifications

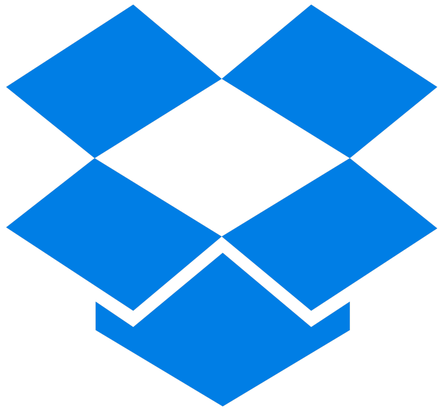
Version control

How it works, at 10,000 feet

- Everyone on the team has a copy of the code
- Every time a person makes a change, they “commit” that change and “push” it to the others
  - Other members of the team can then “pull” all changes
- The version control system keeps track of all differences
  - Keeps a reliable history of what was committed, when, and by whom

Version control

- Version control is essentially providing cloud syncing
  - Shared files across multiple people
  - A reliable backup in case something gets lost/broken
  - Version history
- **So why not just use Dropbox, Google Drive, iCloud, etc.?**



Version control

Why use it over cloud storage?

- Developers don’t always want the latest version
  - Maybe you’re working on a new feature that’s incomplete or buggy
- Version control is better for tracking changes
  - Code is just text, so it’s easier to monitor how one line changes over time
  - Easier to revert a change, or audit when/where a bug might have been introduced

So how do you use version control?

Version control

What should be committed?

- Code and text files
- Images or other resources needed to get your code to run
- Documentation for how your code works
- The names of libraries that you need
- Ideally, someone should be able to download your repository, hit some button to “install”, and then go!

Version control

What should not be committed?

- The actual libraries and frameworks
  - Anything which can be easily downloaded from somewhere else
  - More on this next time
- Secret information
  - Logins, user credentials
  - Assume that everything in version control is intended to be shared

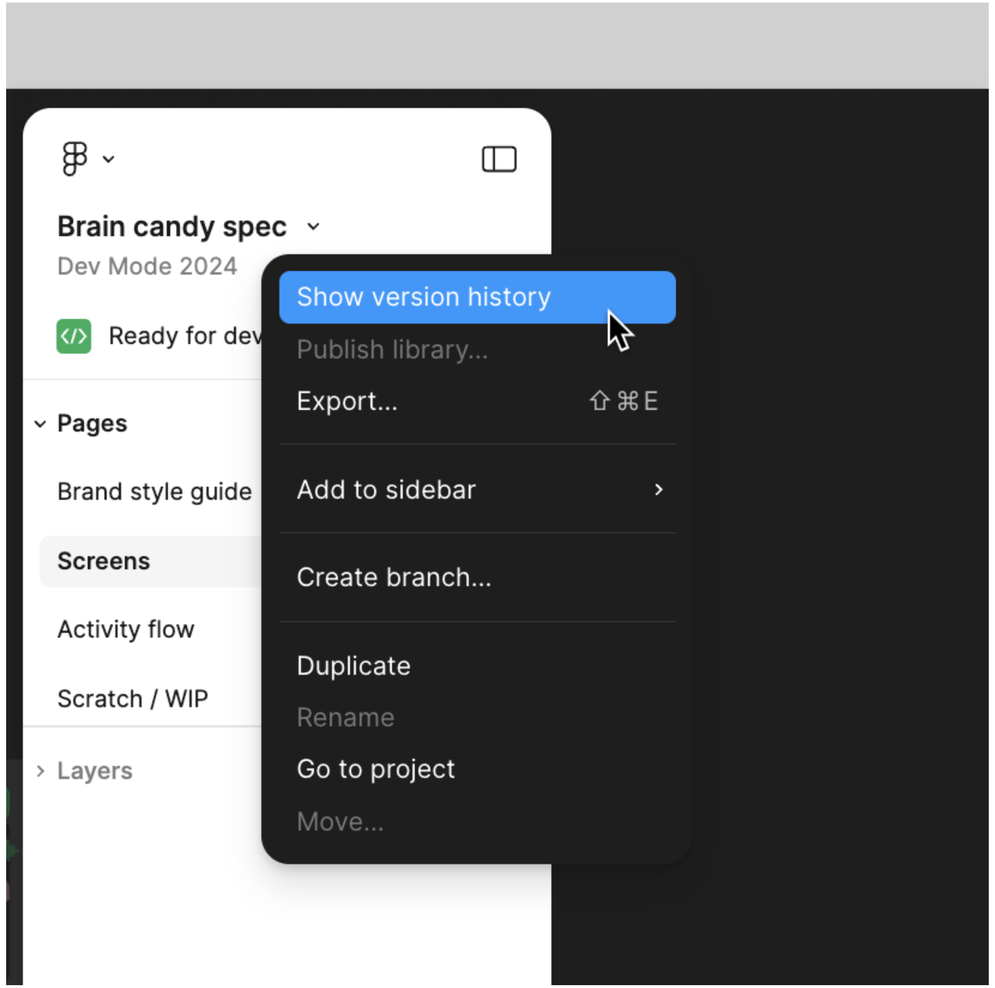
Version control

What should not be committed?

- Very large non-text files
  - Think videos, giant PDFs, slide decks
  - Version control works best with well-structured files (code, text) where it's easy to keep track of differences
  - Big files take a long time to download
- The program you're making itself
  - Version control is for the *code*, not the *output*; don't commit a whole app

Version control

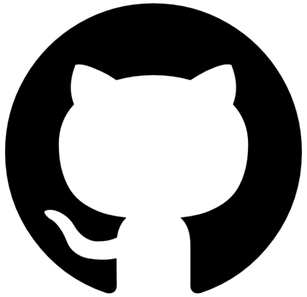
- Other creative tools often have their own system for version control
  - Figma, Dropbox, Google Drive, Adobe Suite
- Developers often let these tools manage their own versions, rather than adding them to their own version control systems



One version control system: Git and GitHub

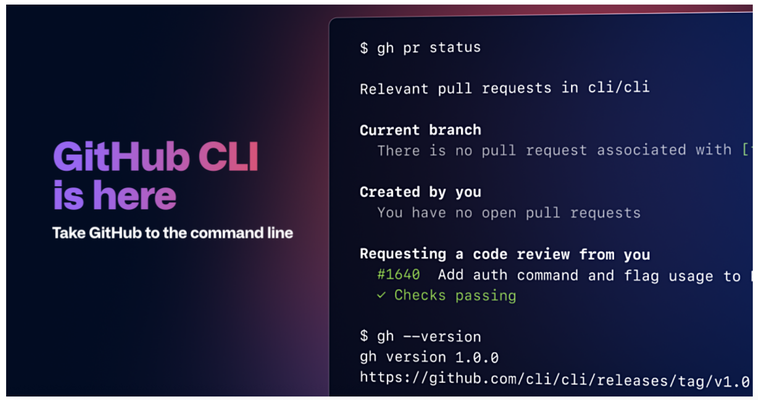
Git and Github

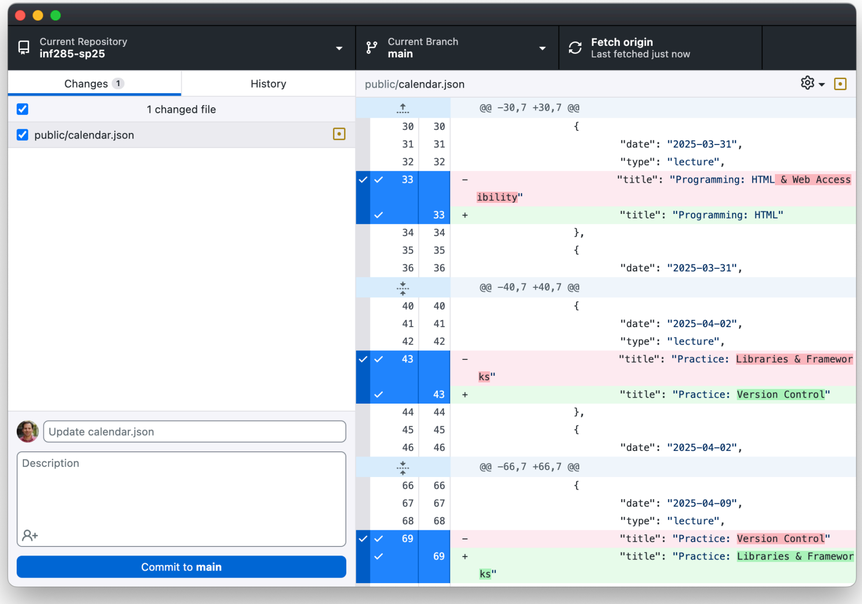
- Git: the protocol specifying how files under version control are managed
  - Free, open-source
- Github: a website/service that hosts Github repositories
  - Owned by Microsoft since 2018
  - Repositories can be public or private
  - The largest host of public code



Git and Github

- Many developers interact with them via command line (terminal)
  - But, unless you're experienced, it's a giant pain to do so
- Github provides Desktop tools which (somewhat) simplify the process



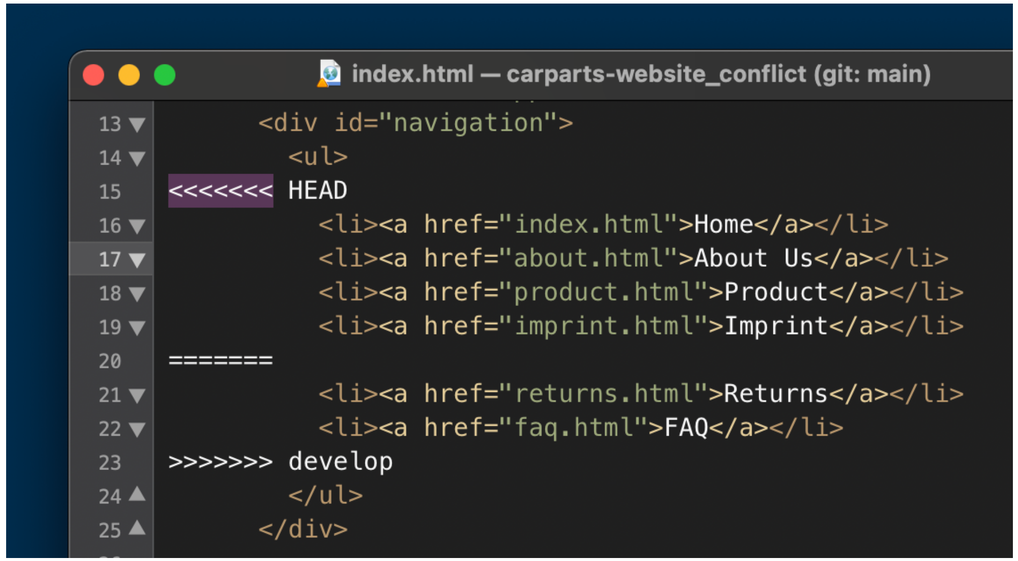


https://github.com/apps/desktop

Some version control concepts

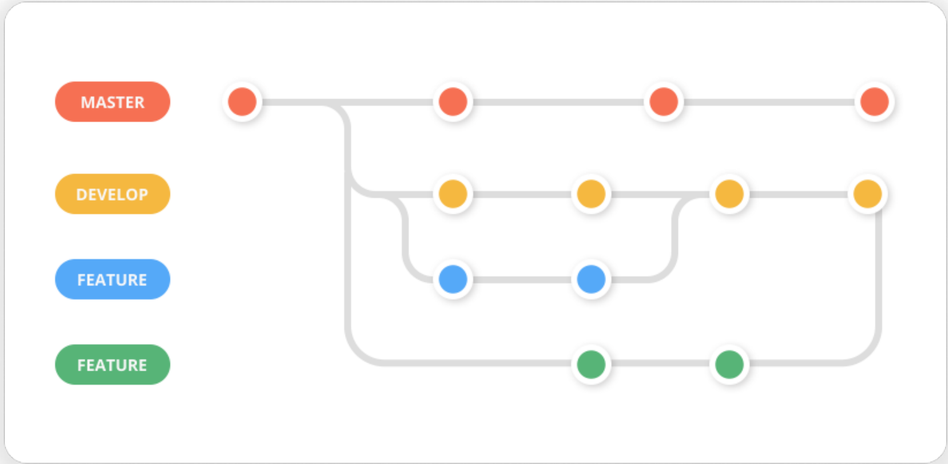
Conflicts

- Uh oh, two people edited the same file at the same time. **Now what?**
- Version control systems can sometimes resolve this automatically
  - If you edited different parts of a file, for example
- But if not, a developer must *resolve* those conflicts
  - Choose which version to use, or merge the two together



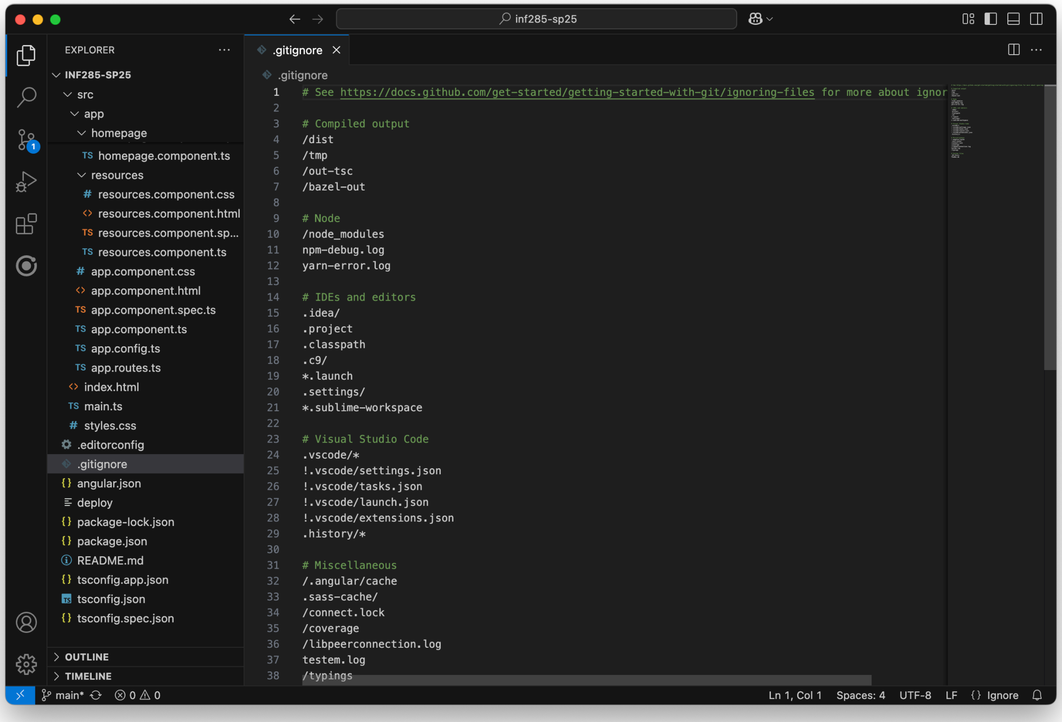
Branches

- You're working on something big, but it'll take a while. You want to still be able to use version control. **What do you do?**
  - You make a *branch* off of the main repository, and *merge* it back in later
- "Main" or "master": the main branch, or sometimes the one which customers/users can see
  - Sometimes there's a separate branch for development



.gitignore

- You need libraries to run your code, and your code keeps generating large files as output. How do you avoid committing these?
  - You list these files in your *.gitignore*
- This file is sometimes written for you
  - There's a whole complicated syntax that we won't get into here



One Github repository:
The IN4MATX 285 website

Today’s goals

By the end of today, you should be able to…

- Describe what version control is, and the basics of how it works
- Explain why developers usually use version control in their projects
- Use Git and GitHub to do basic version control functionality

IN4MATX 285:
Interactive Technology Studio

Practice: Version Control