

Intro to Git/Github

Kevin Chen, Albert Xing

Git

A version-control system!

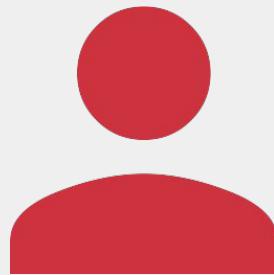
What is version control?

Version control is the practice of
tracking and managing changes
to code

Consider
the
following



Story time...



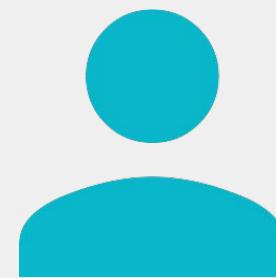
Ben



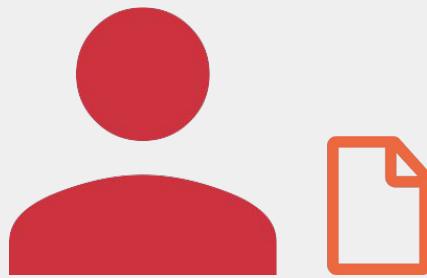
Ben



Ben



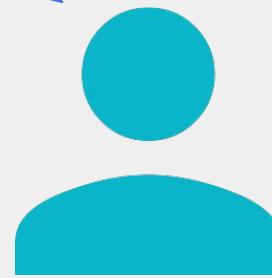
Alice



Ben



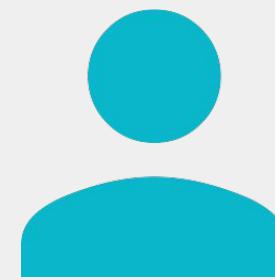
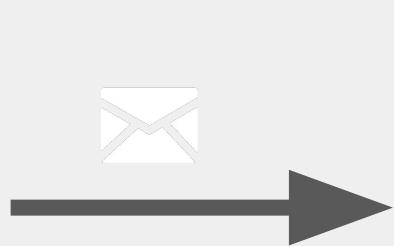
yo sick file. lemme get
in on this



Alice



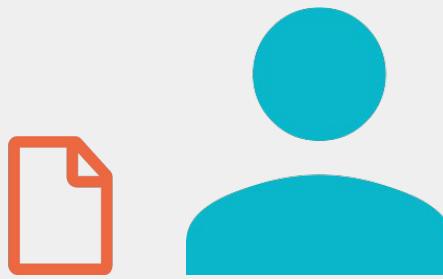
Ben



Alice



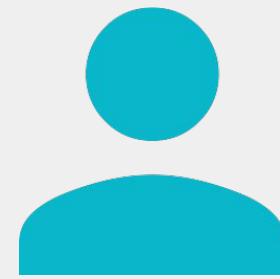
Ben



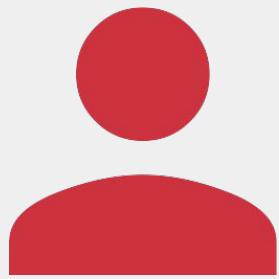
Alice



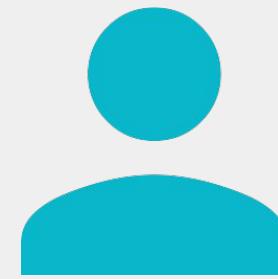
Ben



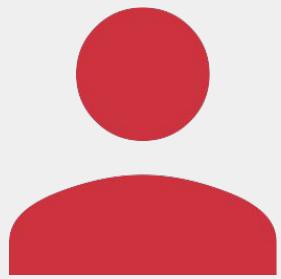
Alice



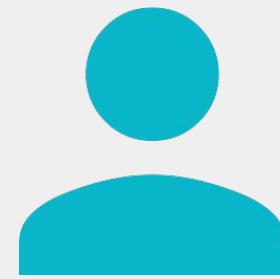
Ben



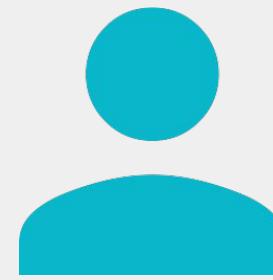
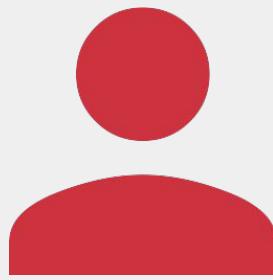
Alice

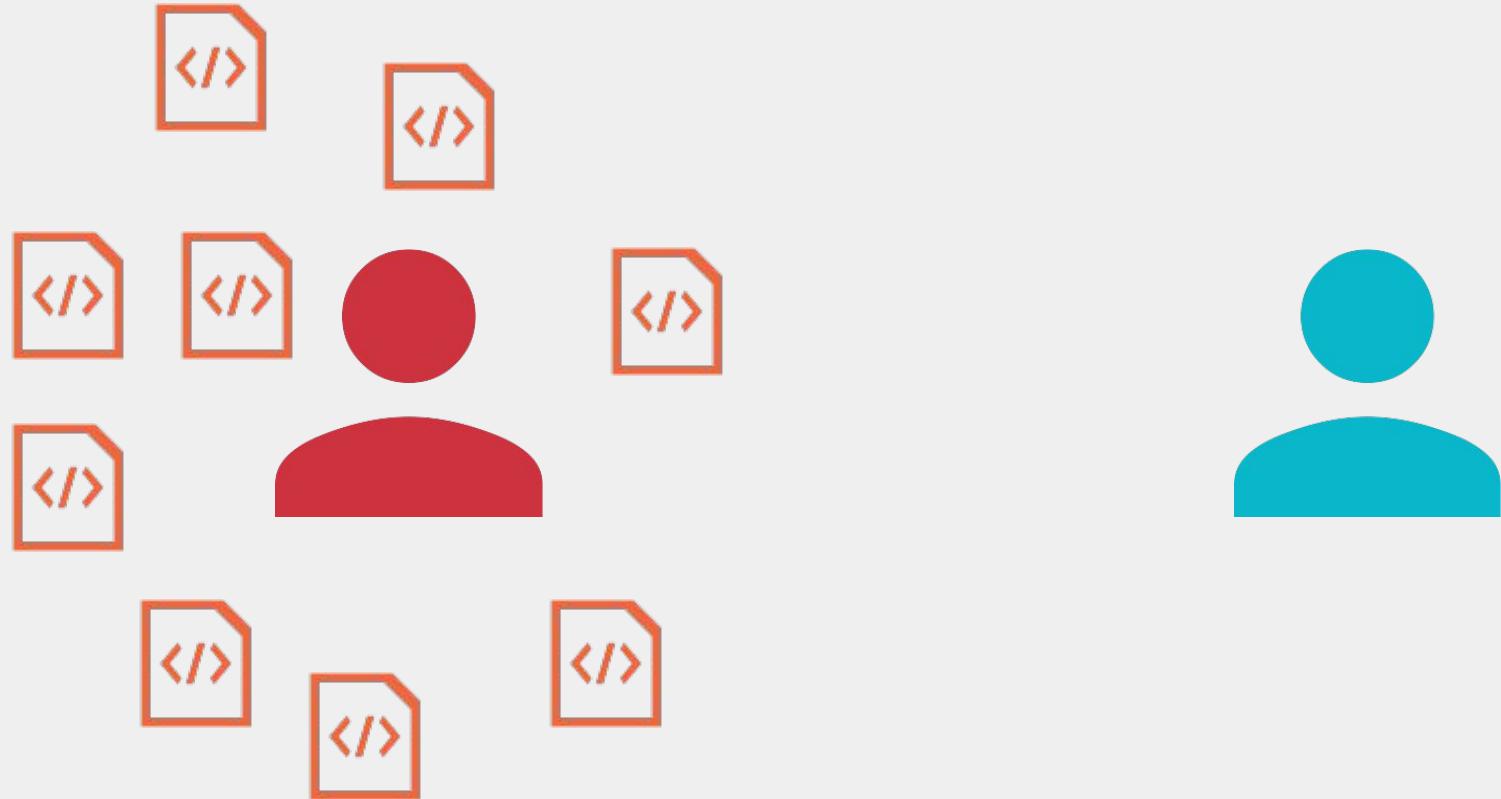


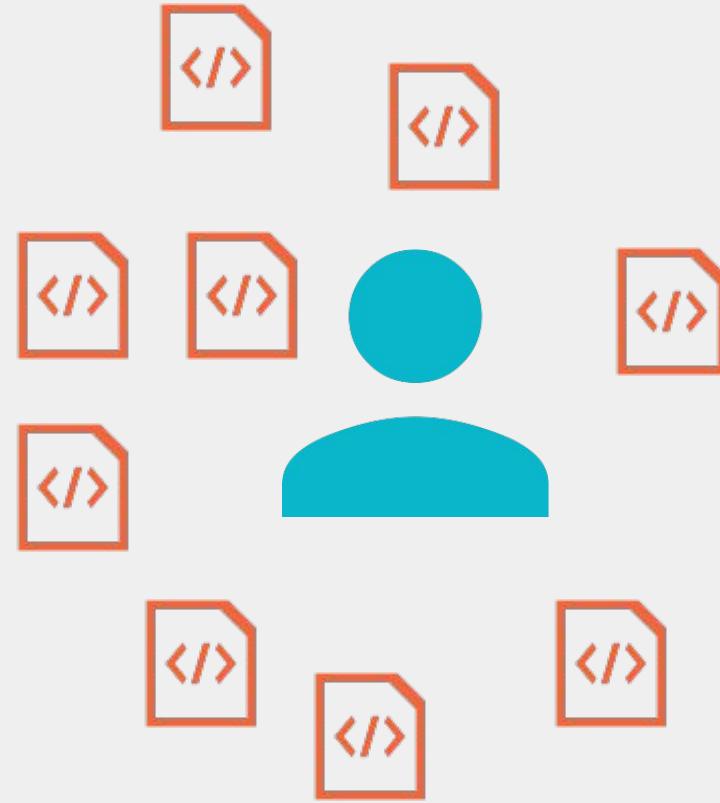
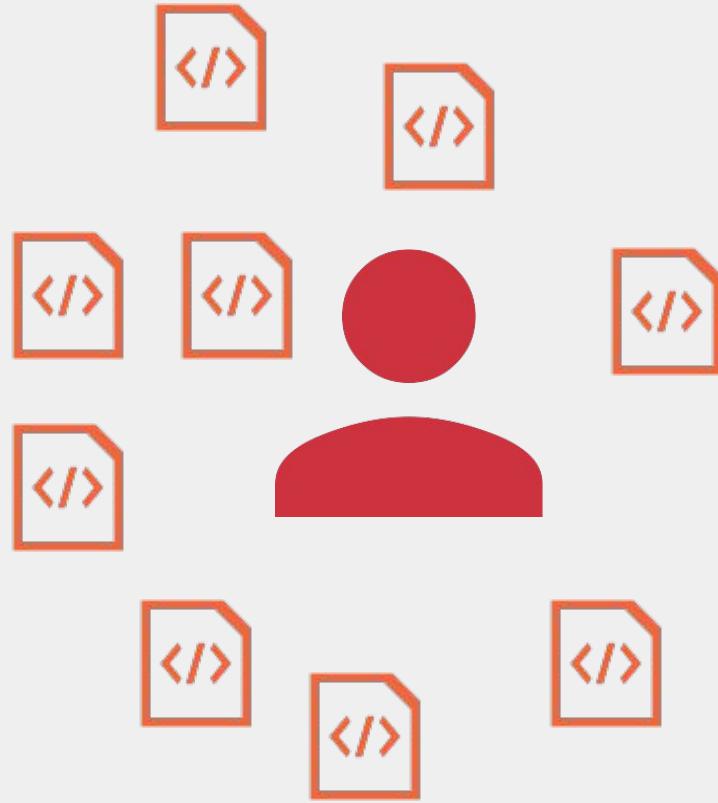
Ben



Alice



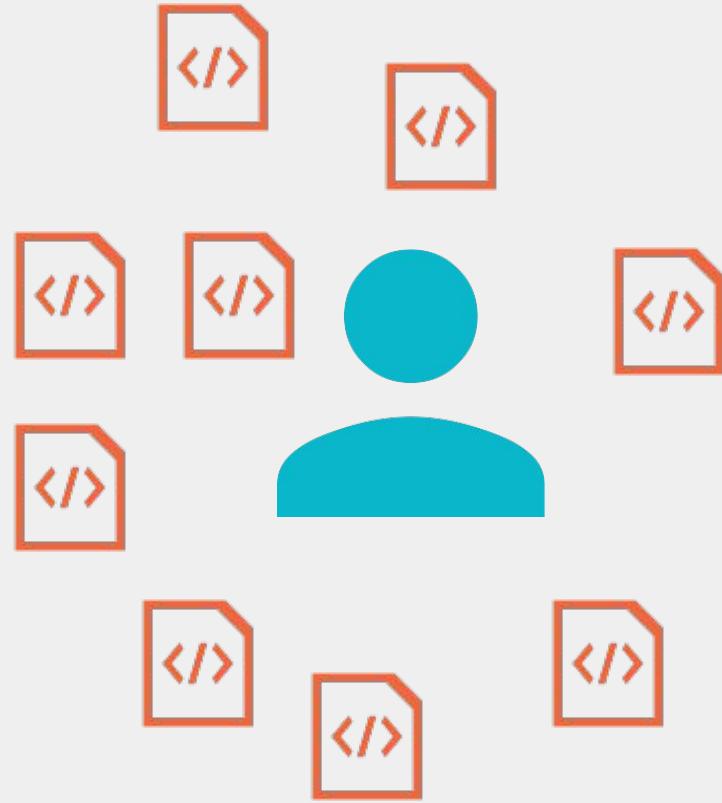
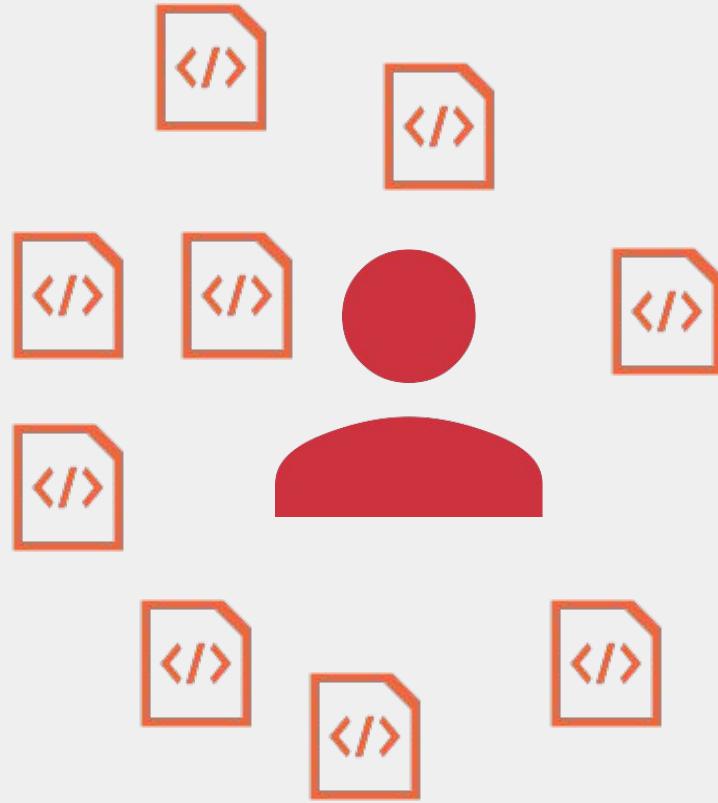


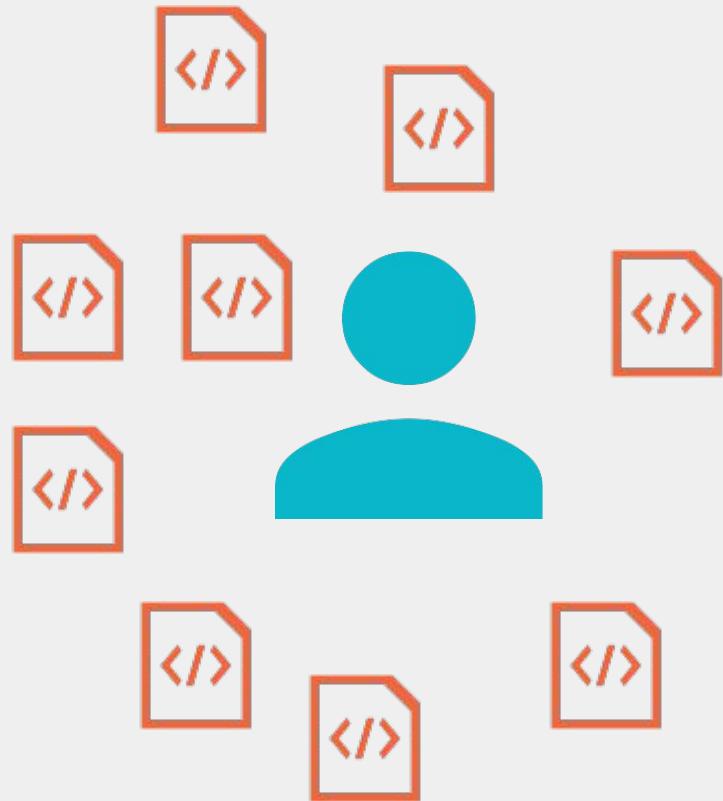
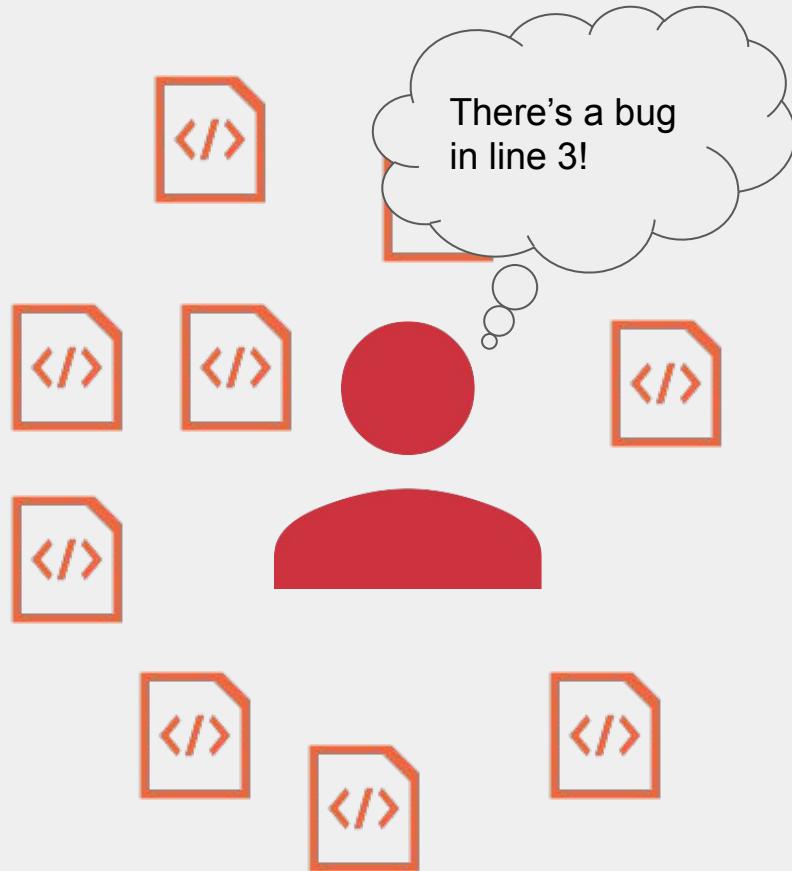


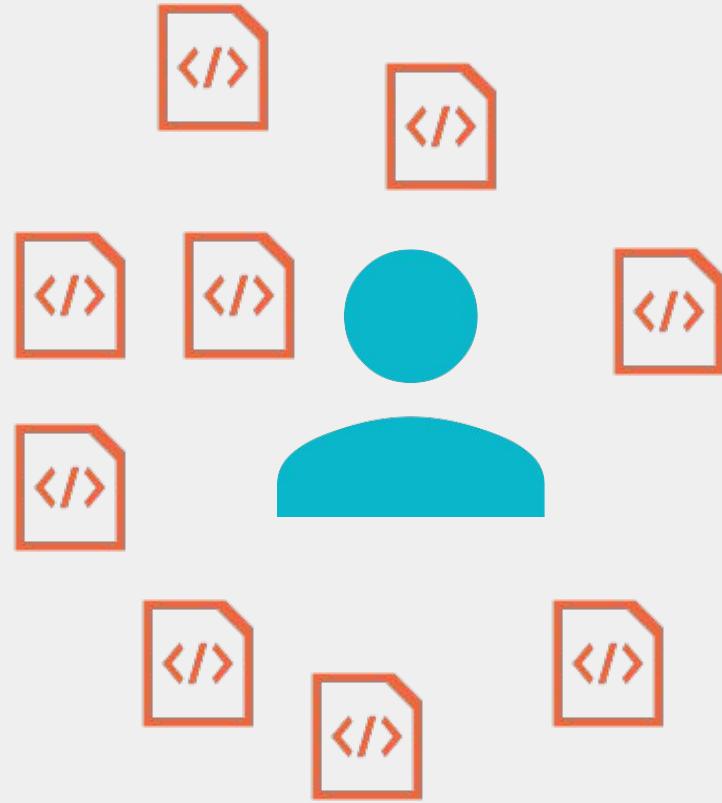
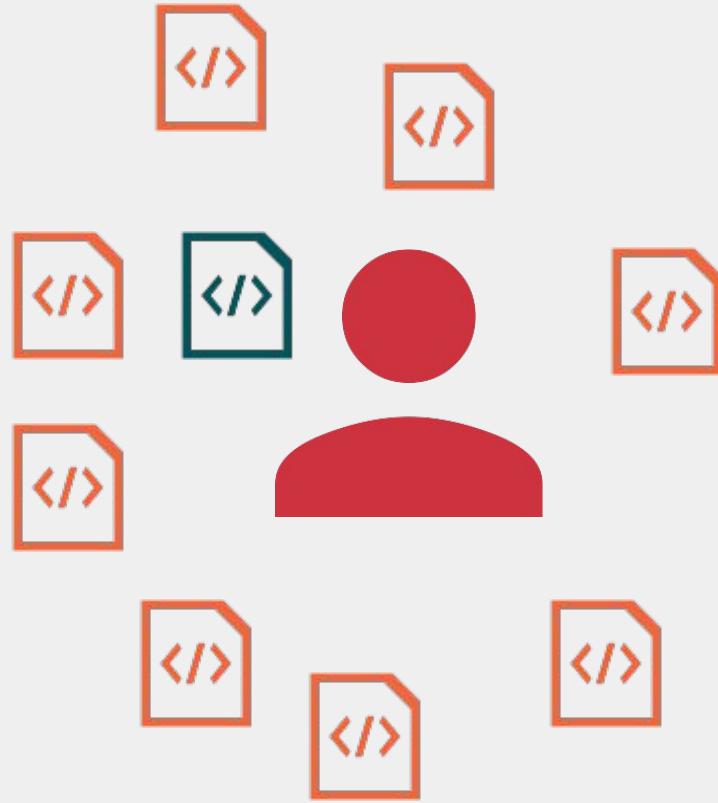
What can possibly go wrong?

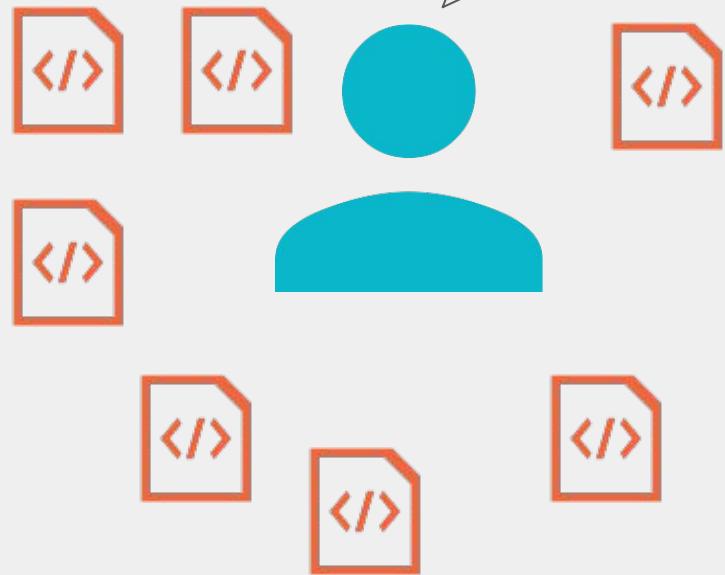
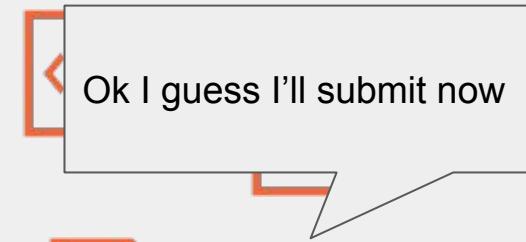
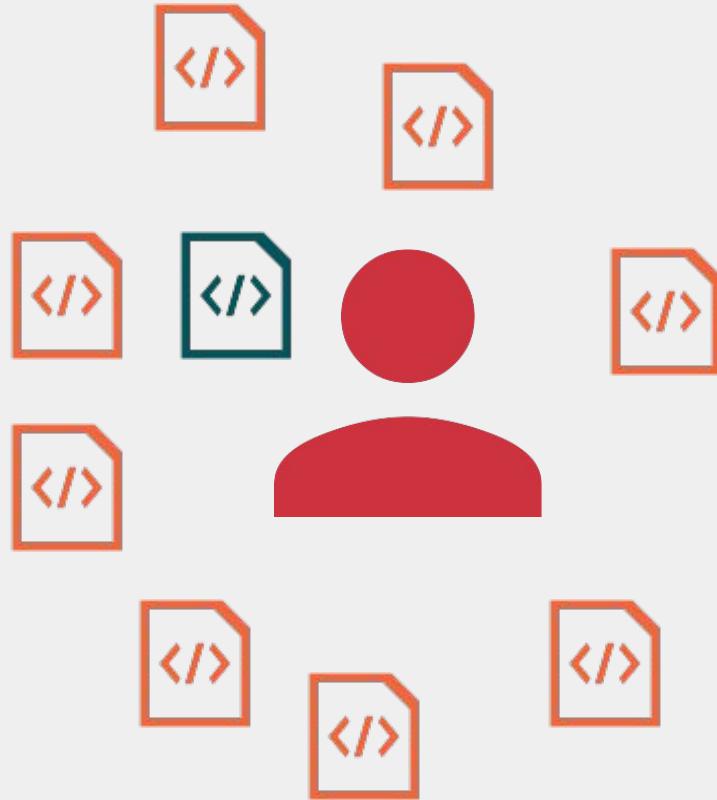
Problems with coding in projects

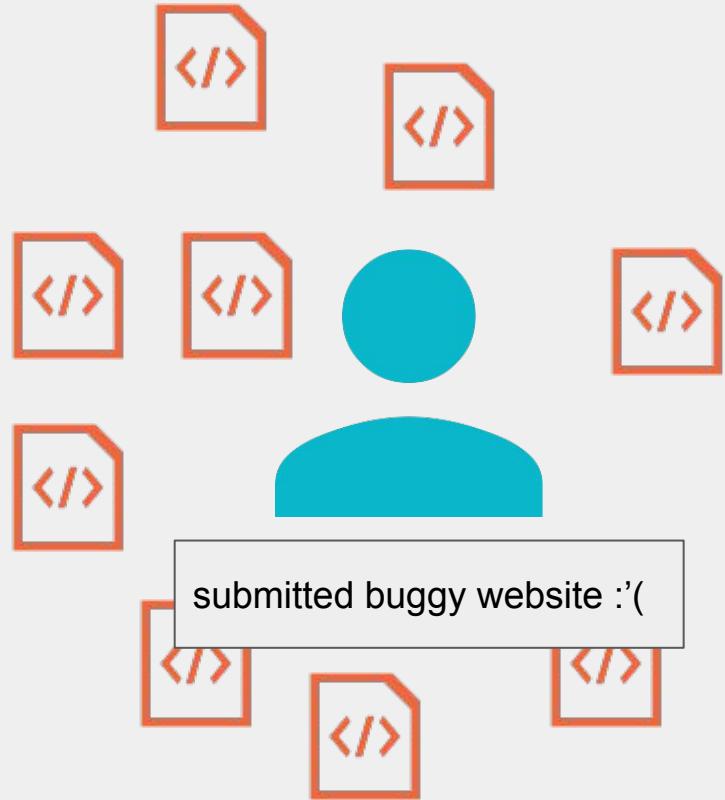
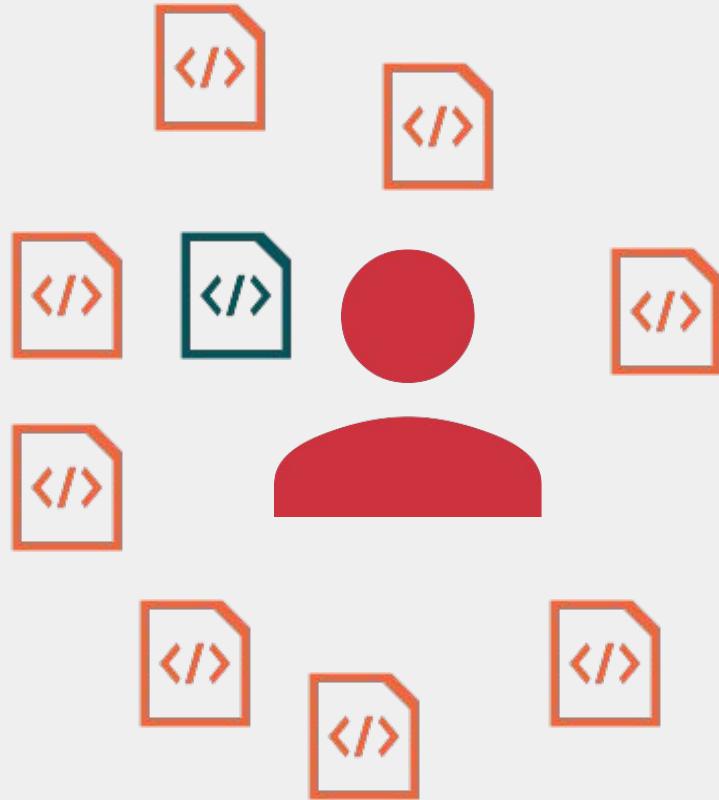
1. Synchronization of files











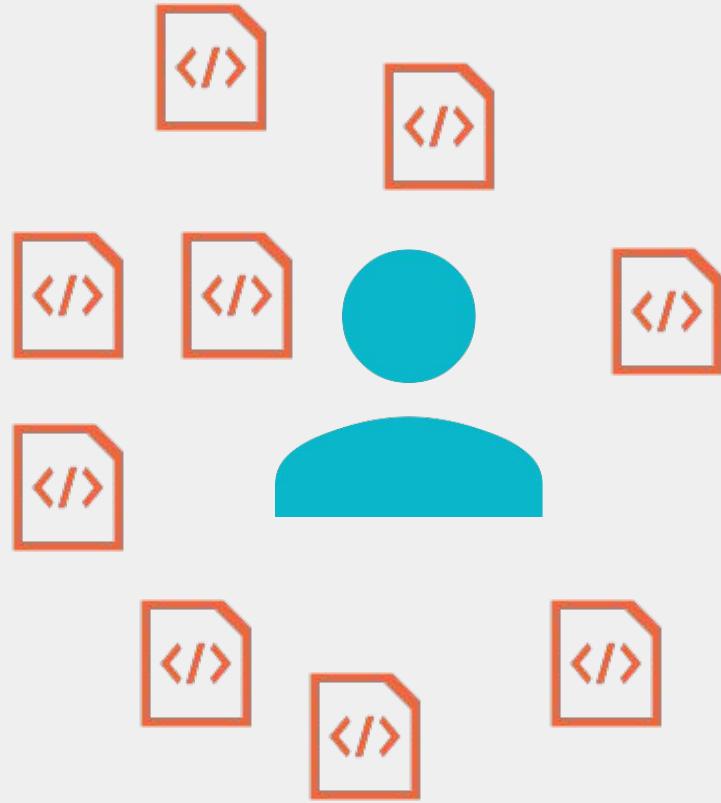
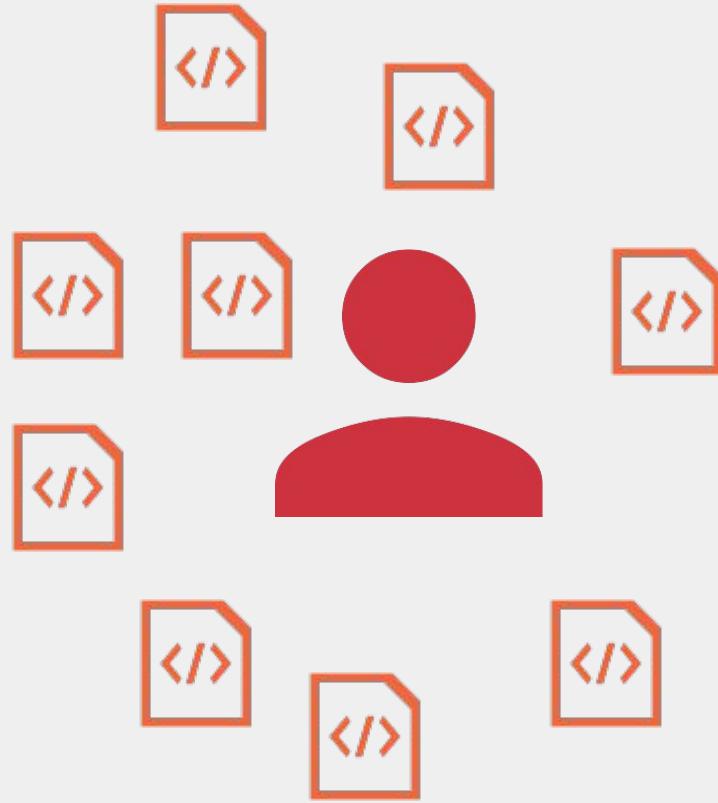


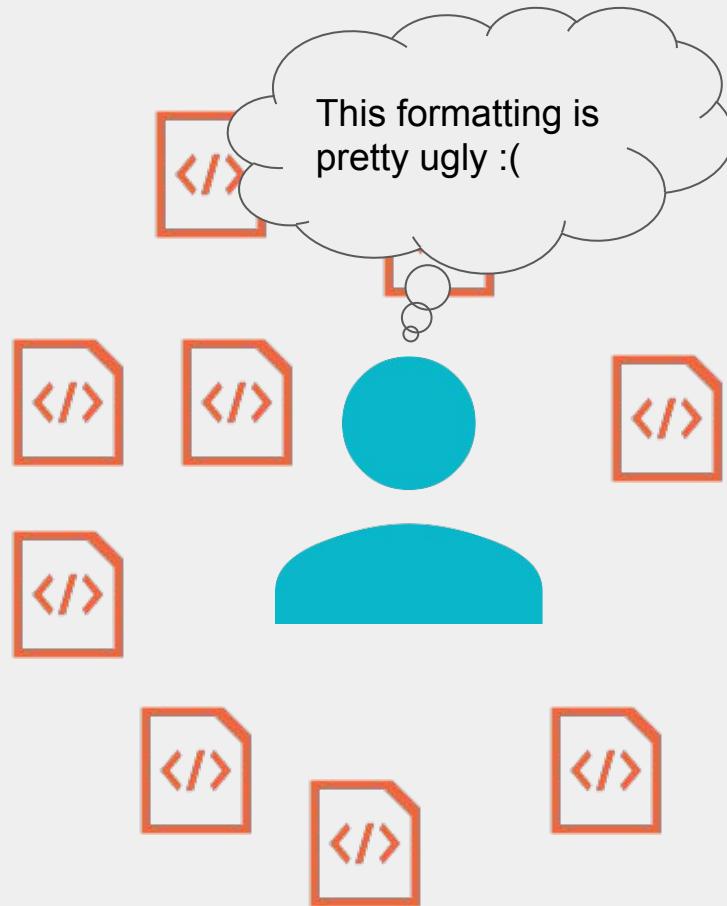
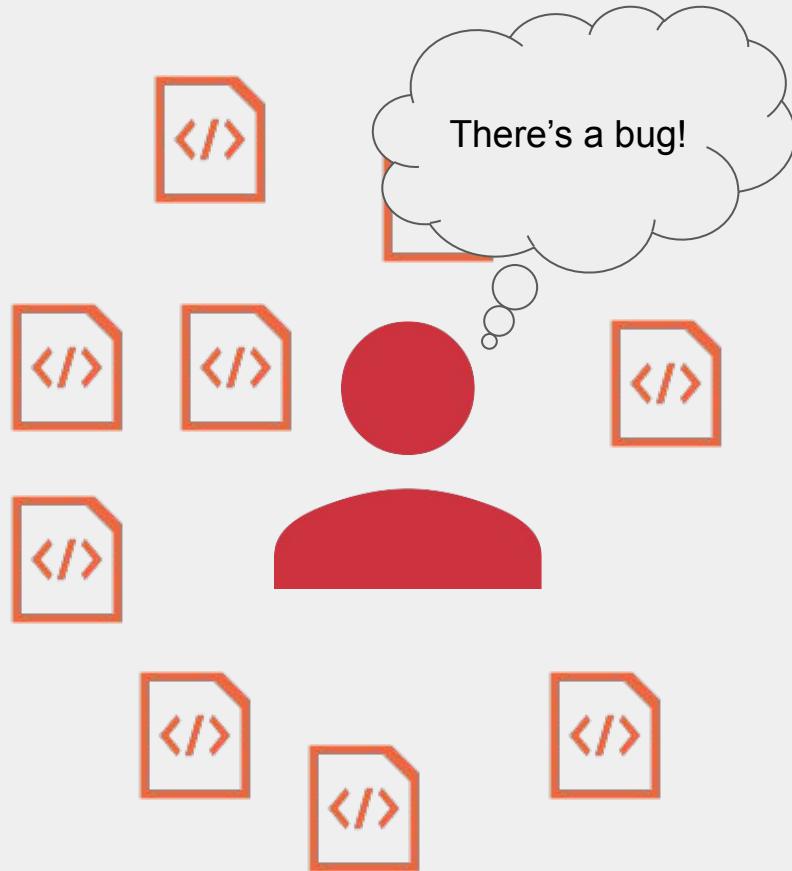
Problems with coding in projects

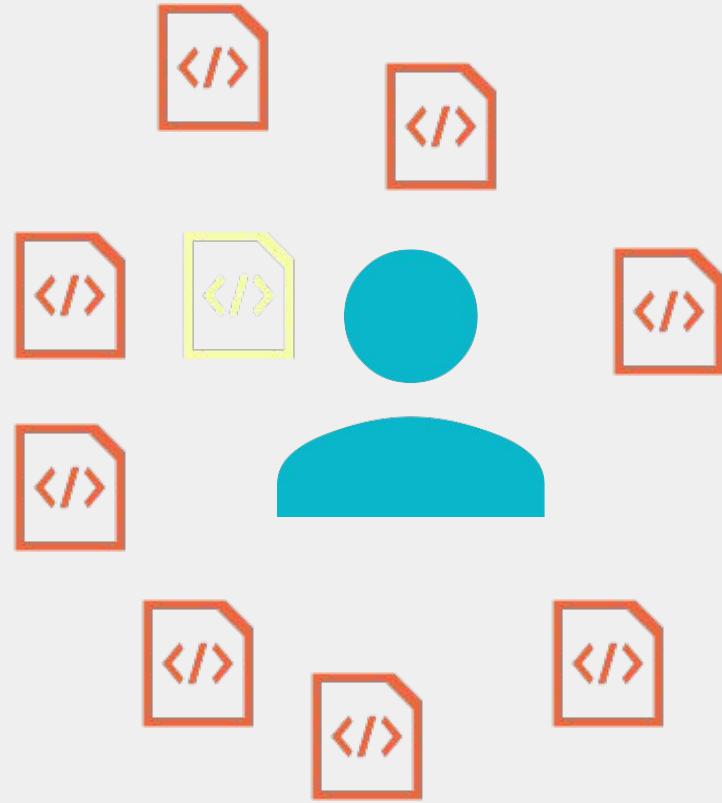
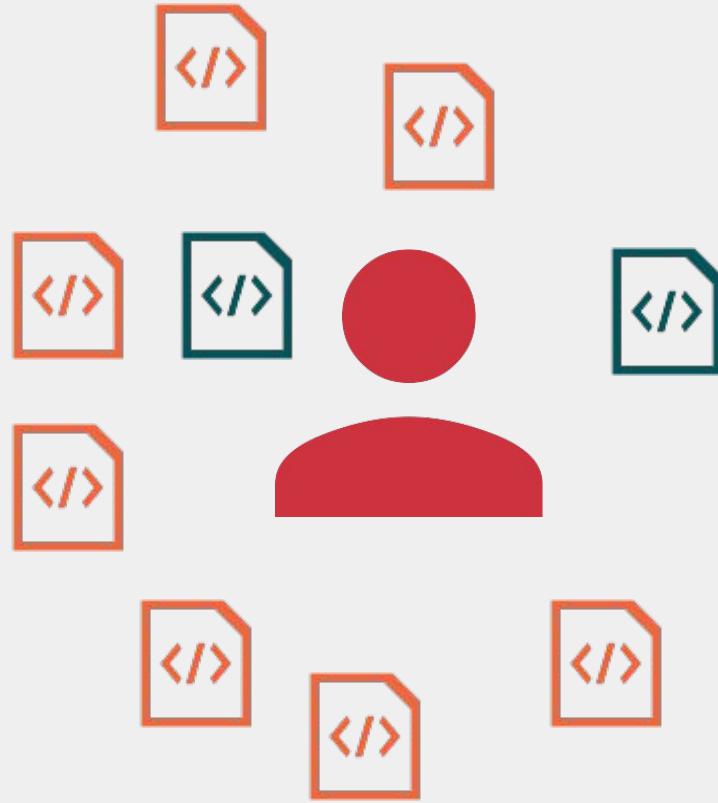
1. Synchronization of files (no one knows which file is the most up-to-date)

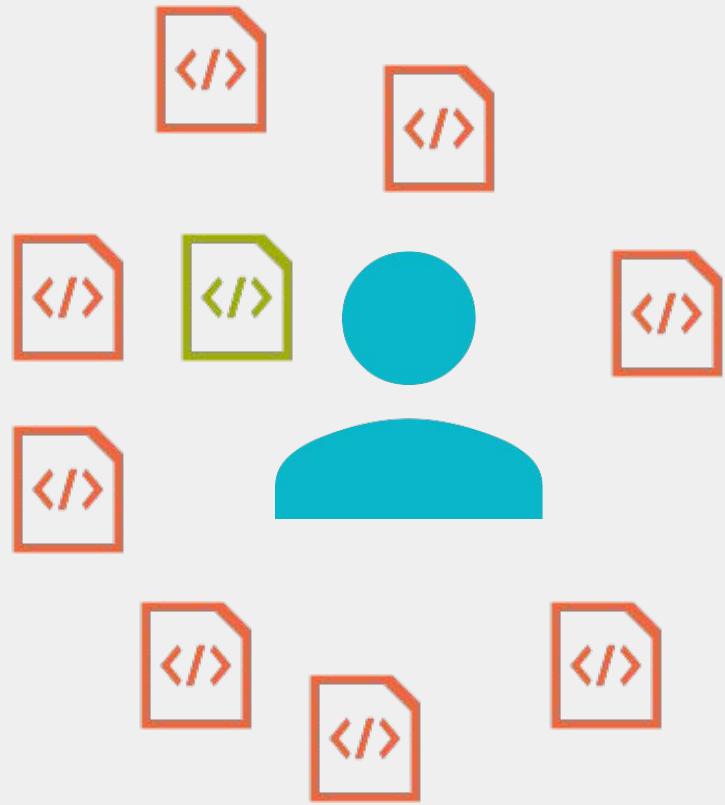
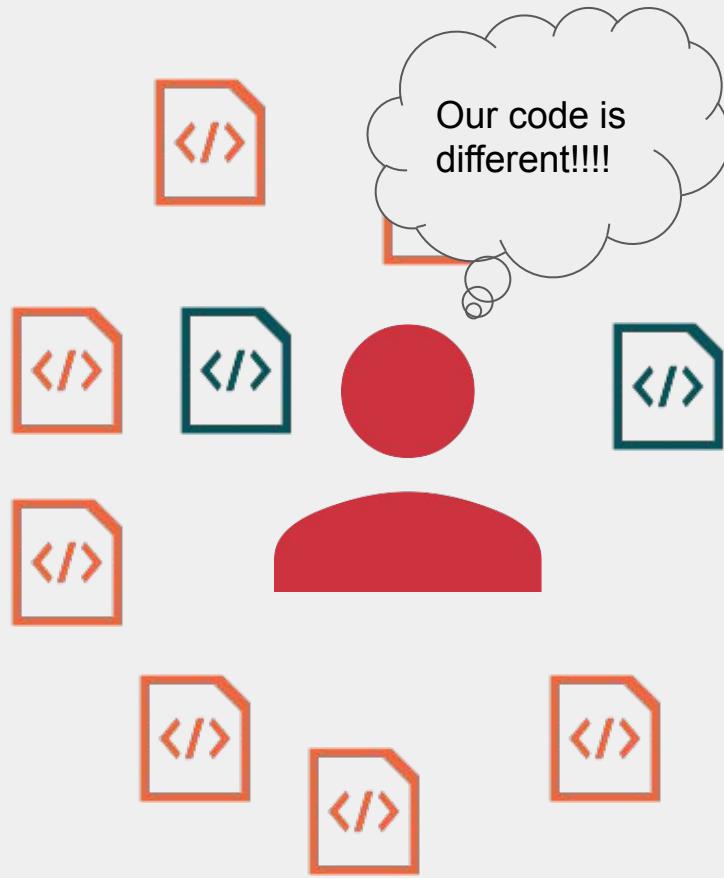
Problems with coding in projects

1. Synchronization of files (no one knows which file is the most up-to-date)
2. Collaboration of files

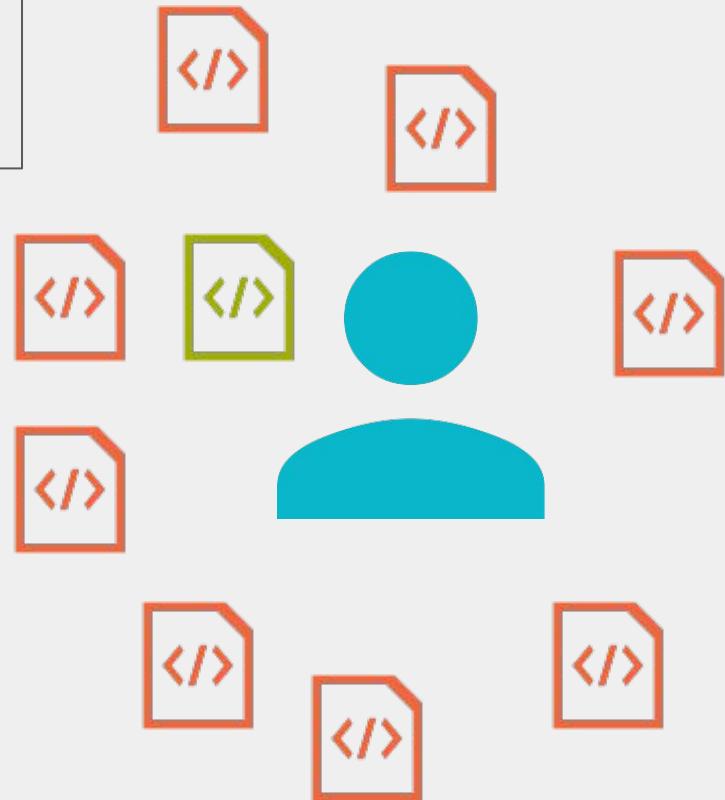
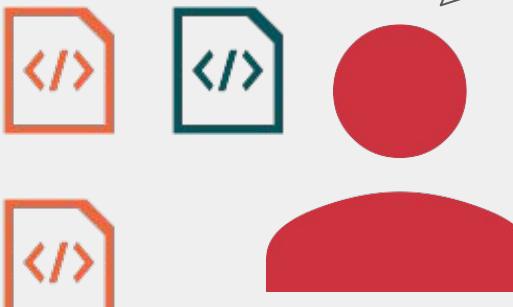


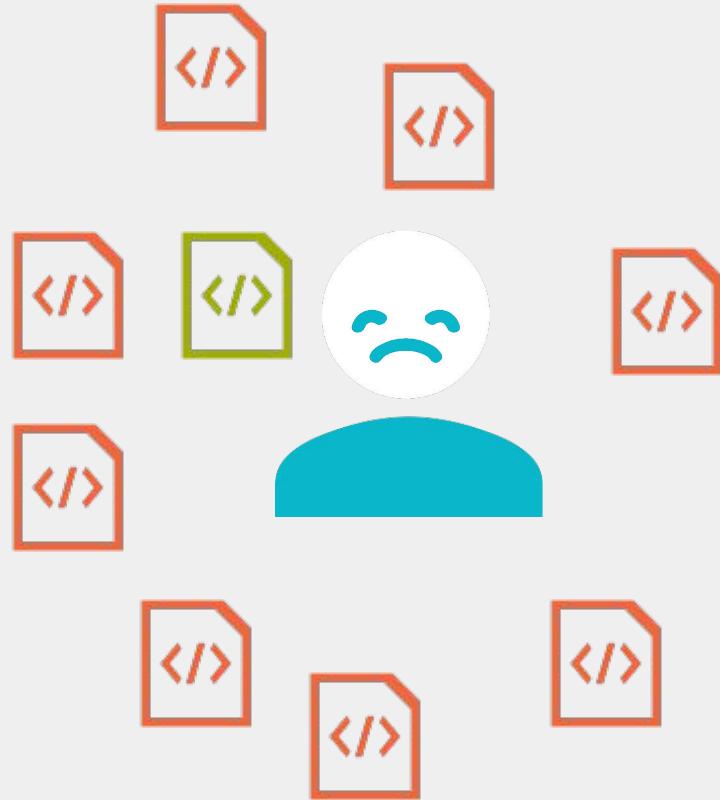
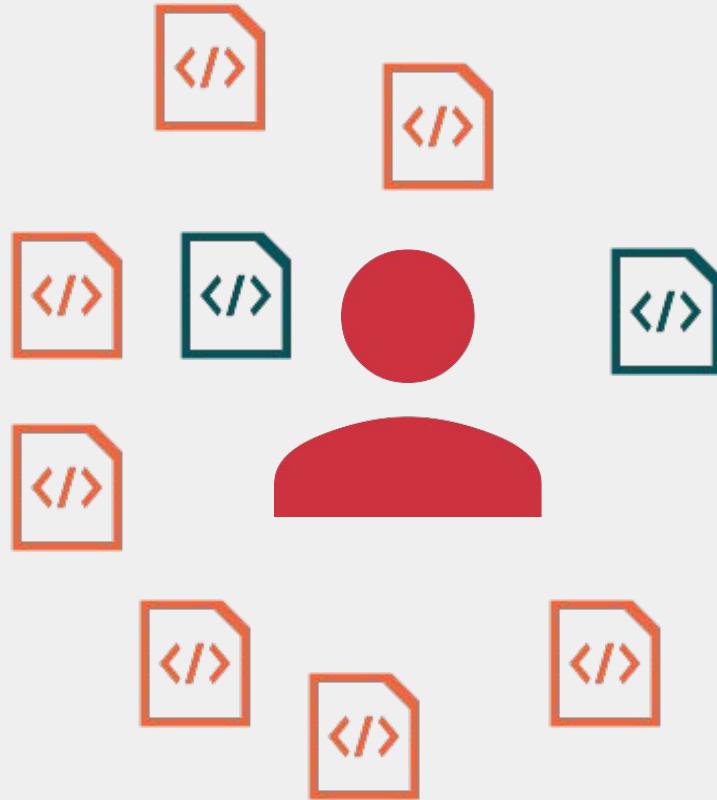






Hey I know you made changes in like
50 places in that file but can you copy
the changes over one by one



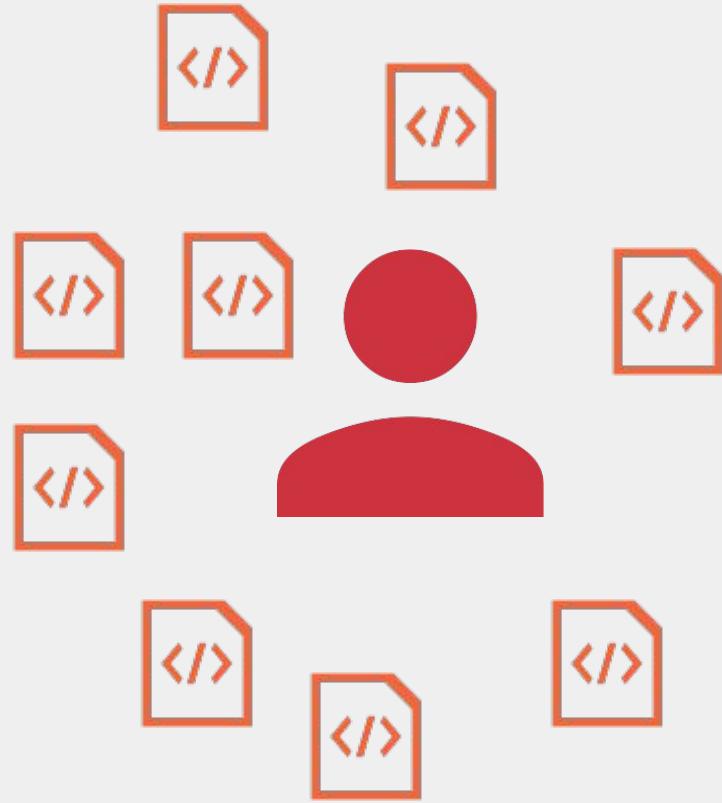


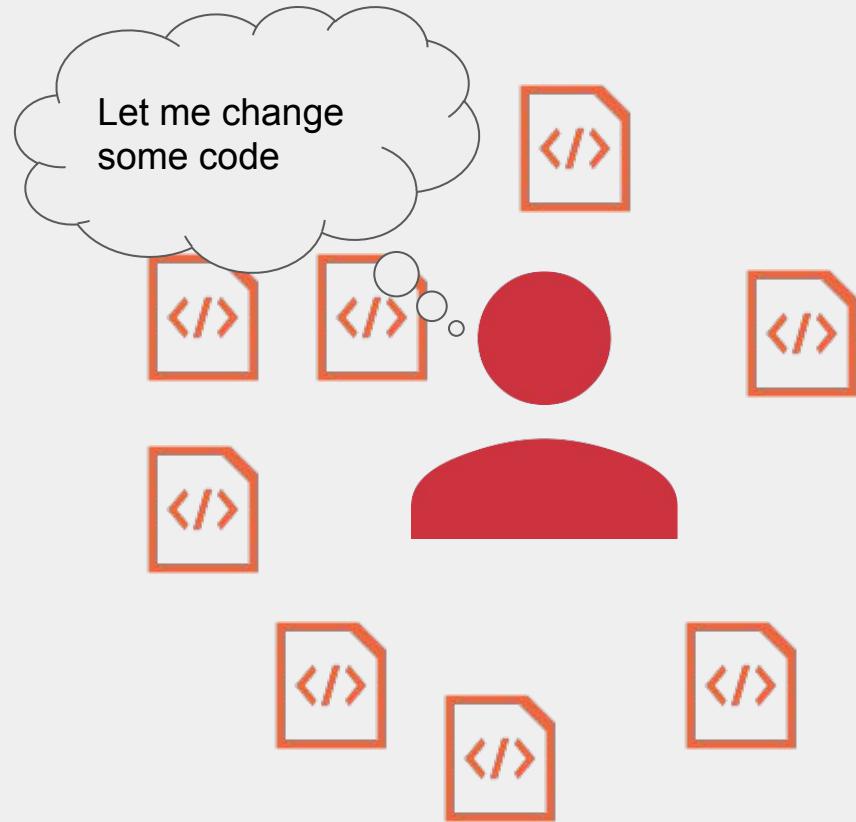
Problems with coding in projects

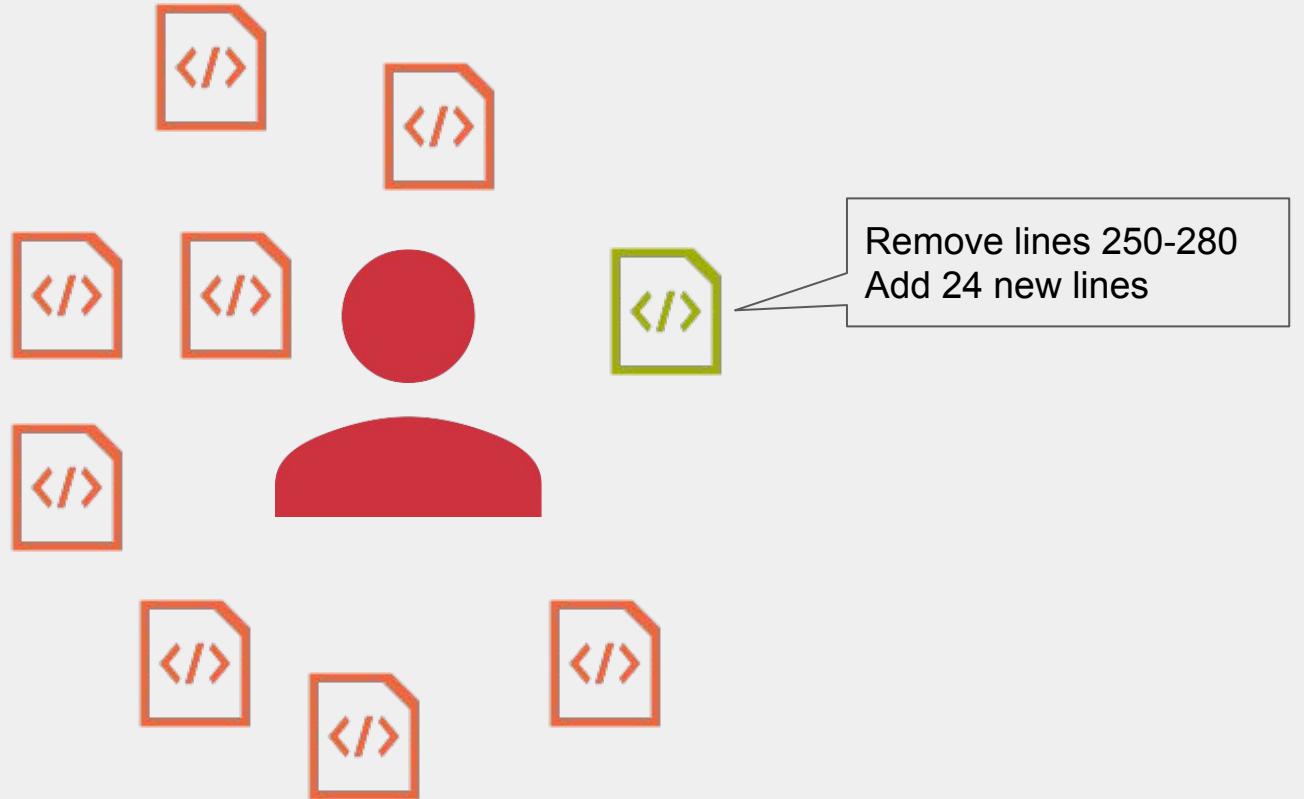
1. Synchronization of files (no one knows which file is the most up-to-date)
2. Collaboration of files (if there are two versions of the same file, both of them fix a different problem, it's hard to know how to join the files)

Problems with coding in projects

1. Synchronization of files (no one knows which file is the most up-to-date)
2. Collaboration of files (if there are two versions of the same file, both of them fix a different problem, it's hard to know how to join the files)
3. Version history

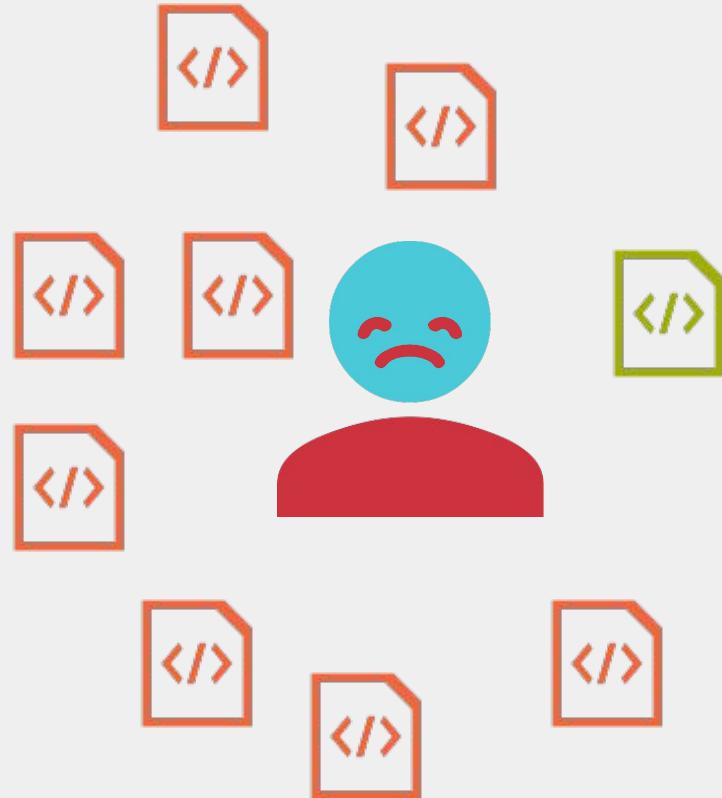






An hour later...





Problems with coding in projects

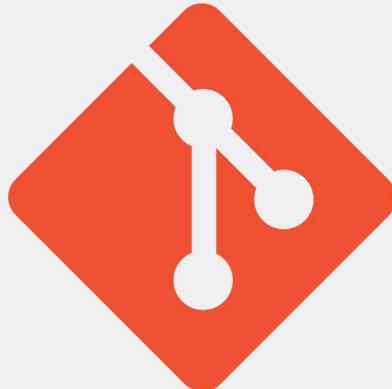
1. Synchronization of files (no one knows which file is the most up-to-date)
2. Collaboration of files (if there are two version of the same file, both of them fix a different problem, it's hard to know how to join the files)
3. Version history (if we break something, we can't go back to a working copy)

How do we solve these problems?

What about Google Docs?

- Solves our synchronization and merging issues!
- Keeps a version history automatically!
- BUT, real-time collaboration is usually *not* what you want...
 - What if you're developing different features?
 - We want to be able to work locally, then sync up (or merge) changes when finished

How do we solve these issues?
Introducing...



git

What does Git do?

Git tracks *changes*.
(How? ~math~)

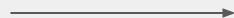
mydoc.txt

I'm writing a story.

It's really good!

And about cats!

version1



I'm writing a story.

It's really good!

And about cats!

Like corgis :D

version2

mydoc.txt

I'm writing a story.

It's really good!

And about cats!

Like corgis :D

version2



I'm writing a story.

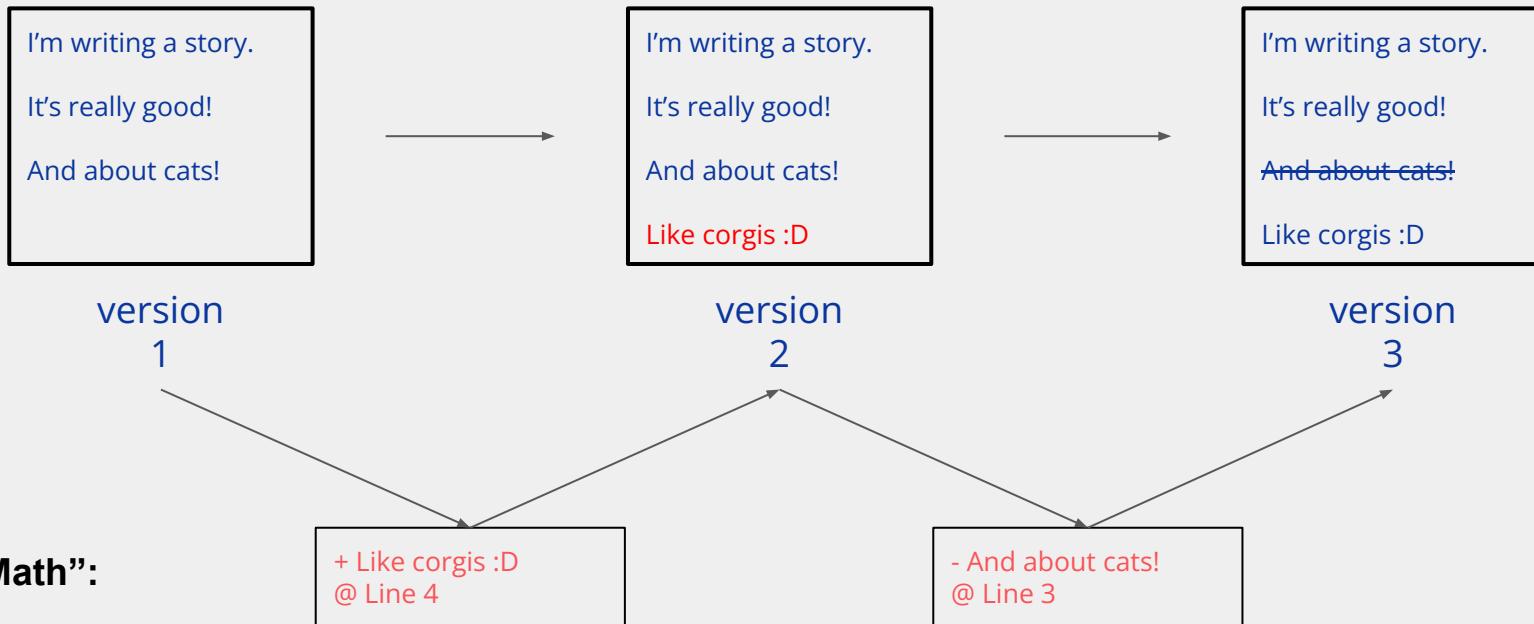
It's really good!

~~And about cats!~~

Like corgis :D

version3

mydoc.txt



In Git terminology...

- **Repository** - a set of files to track changes on
 - Aka your website
- **Commit** - a set of changes for a repository, transitioning from one version to the next
 - Aka the “math”
- **Log** - the set of commits for each version of the repository
 - Aka the commit history

Problems with coding in projects

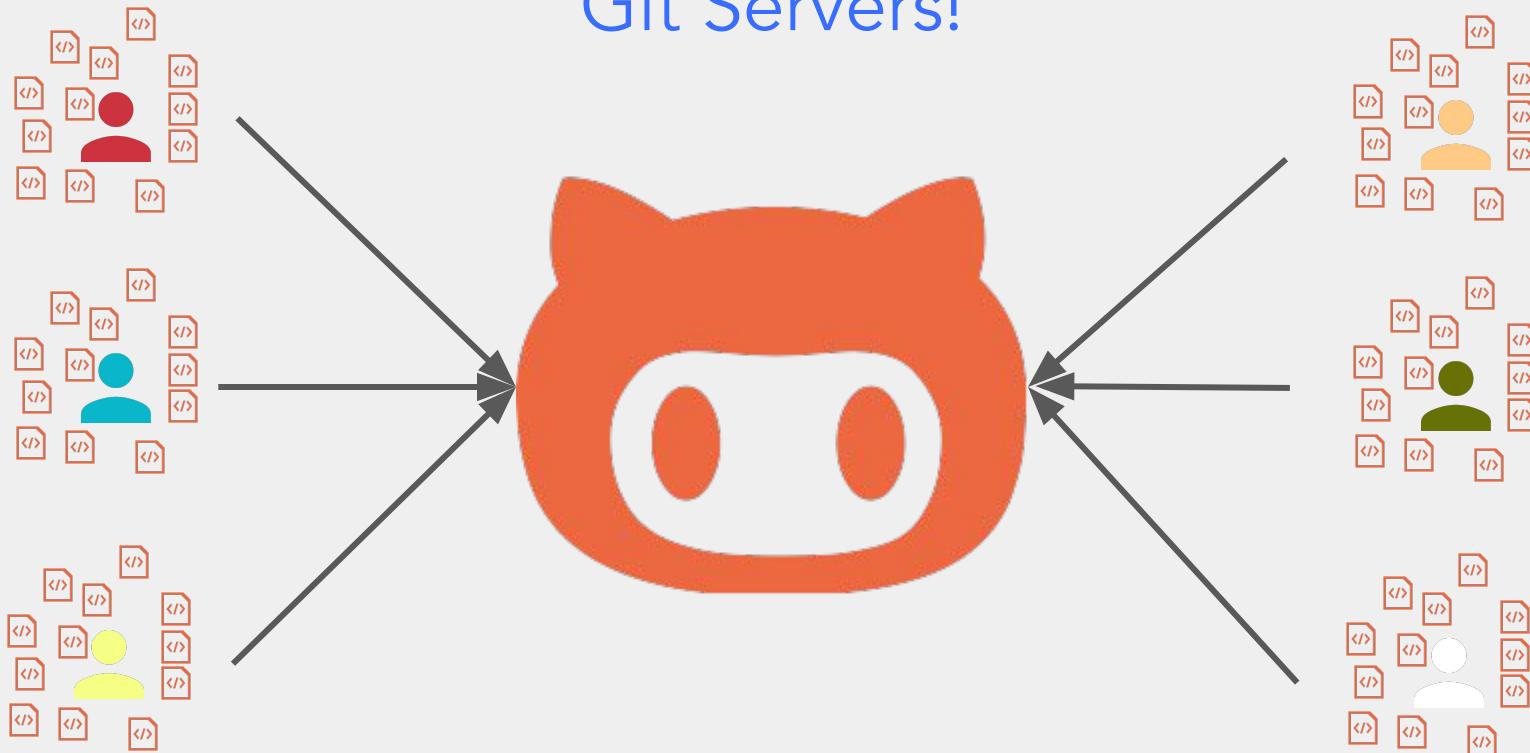
1. Synchronization of files
2. Collaboration of files
3. Version history

Problems with coding in projects

1. Synchronization of files
2. Collaboration of files
3. Version history
 - a. Keep a log of the changes to a file!
 - b. Allows us to easily revert to any previous version.

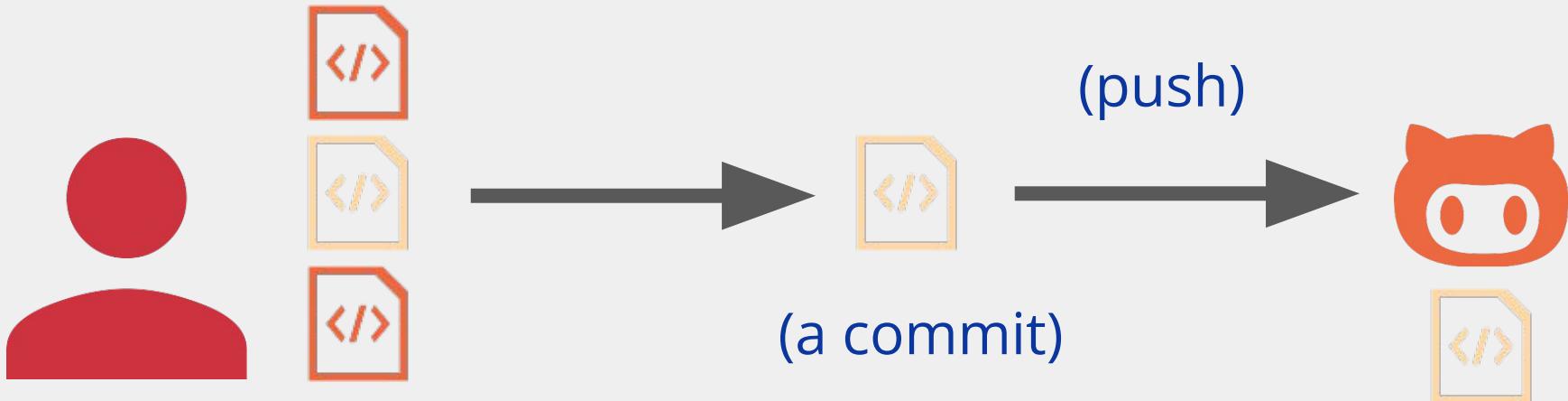
But what about synchronization and collaboration?

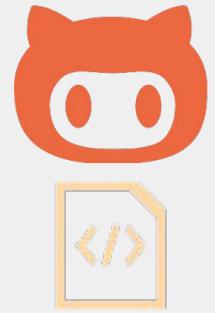
Git Servers!



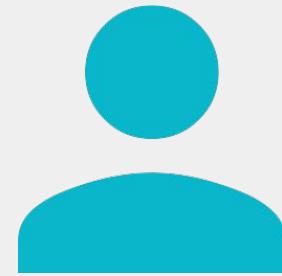
How does this work?

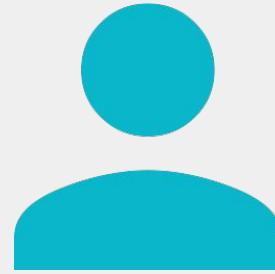
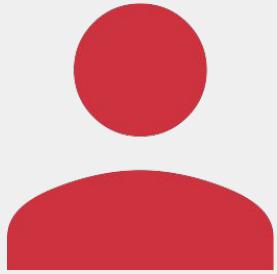






(pull)





Git terms so far...

commit (n.) - a packaged set of changes

push (v.) - to send commits to the git server

pull (v.) - to get most updated files from the git server

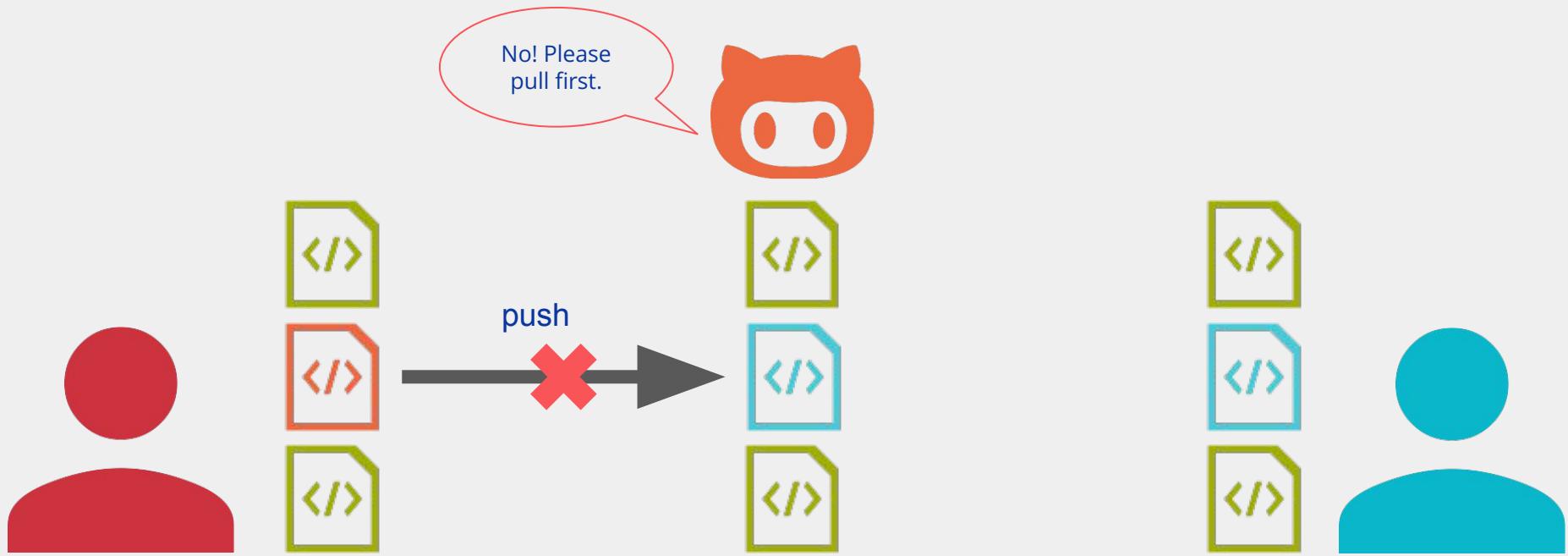
But what about merges?

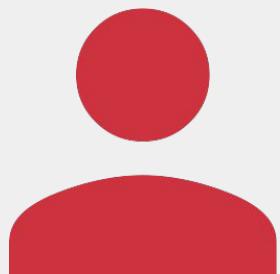


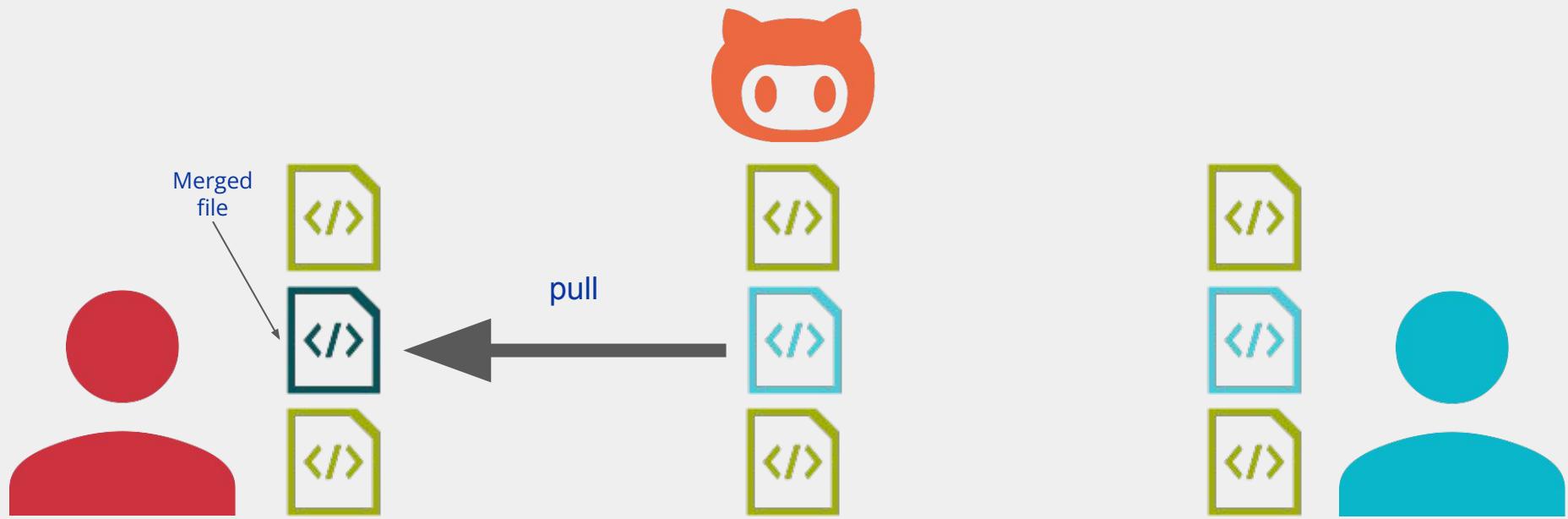


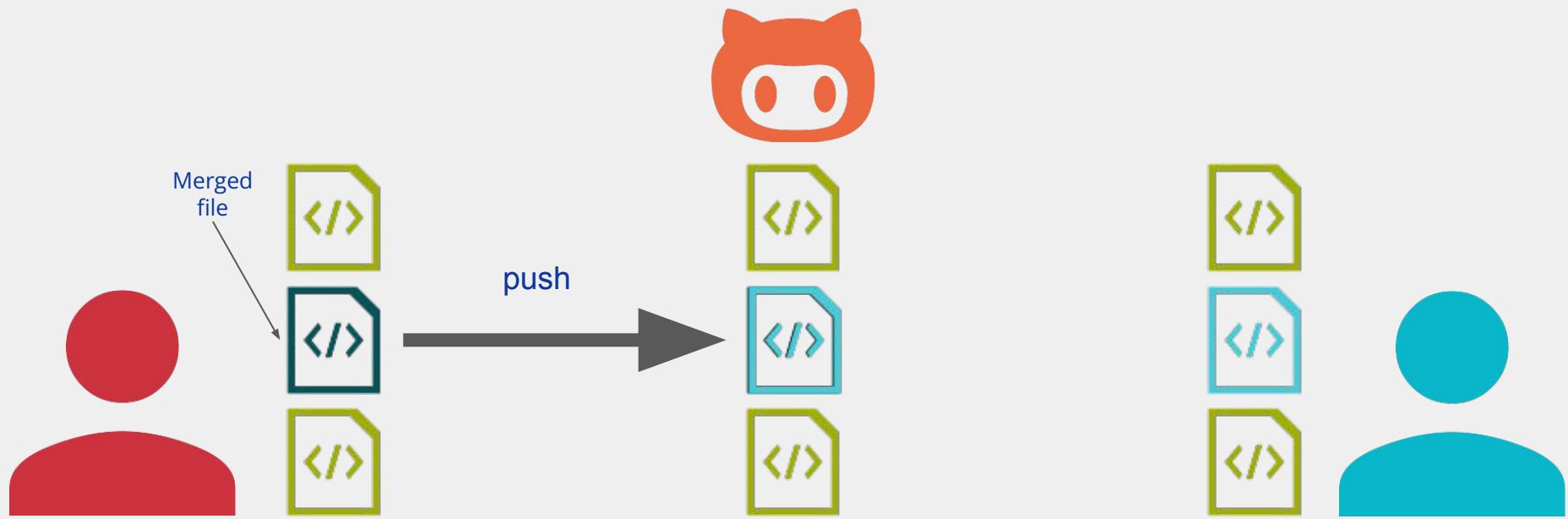
push











Problems with coding in projects

1. Synchronization of files
2. Collaboration of files
3. Version history
 - a. Keep a log of the changes to a file!
 - b. Allows us to easily revert to any previous version.

Problems with coding in projects

1. Synchronization of files

- a. Push commits to a Git server (a single, unified source of truth)
- b. Pull collaborator's changes before modifying on your computer

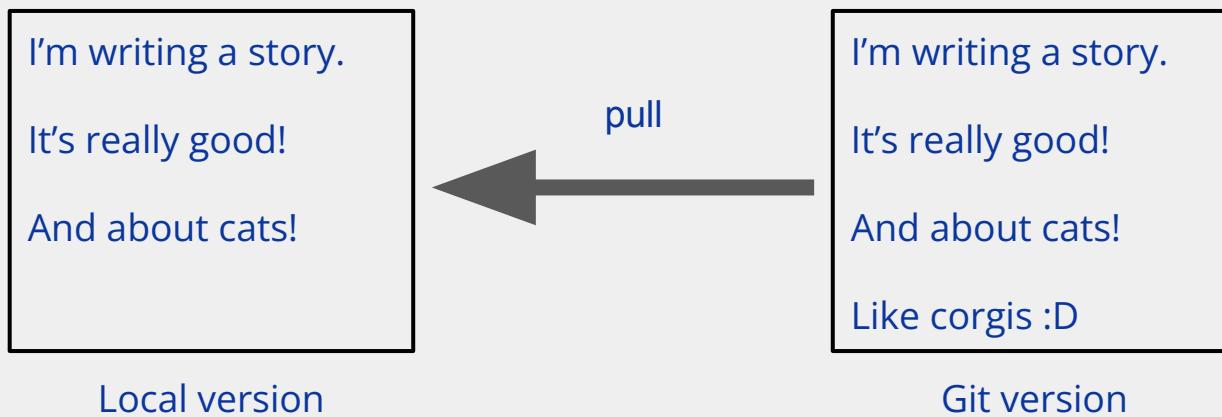
2. Collaboration of files

3. Version history

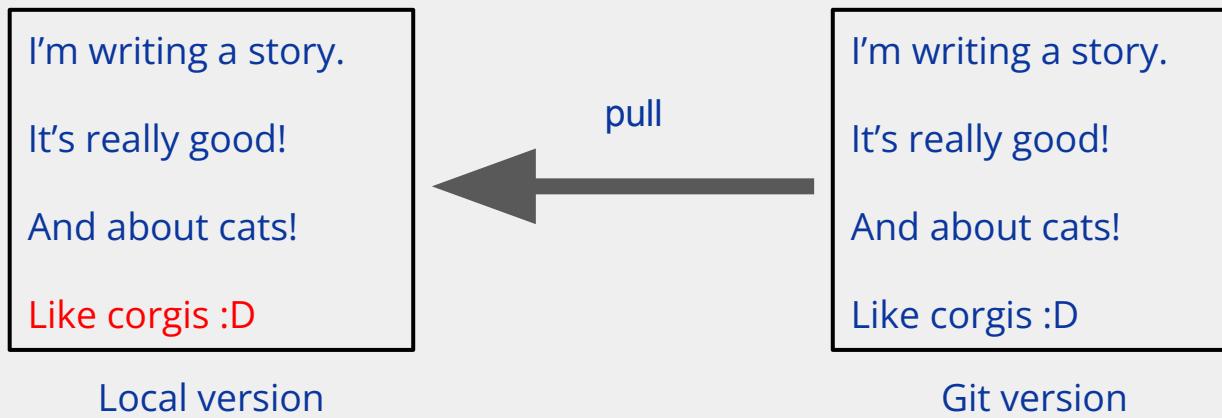
- a. Keep a log of the changes to a file!
- b. Allows us to easily revert to any previous version.

What actually happens during a merge,
though?

Scenario 1 – mydoc.txt

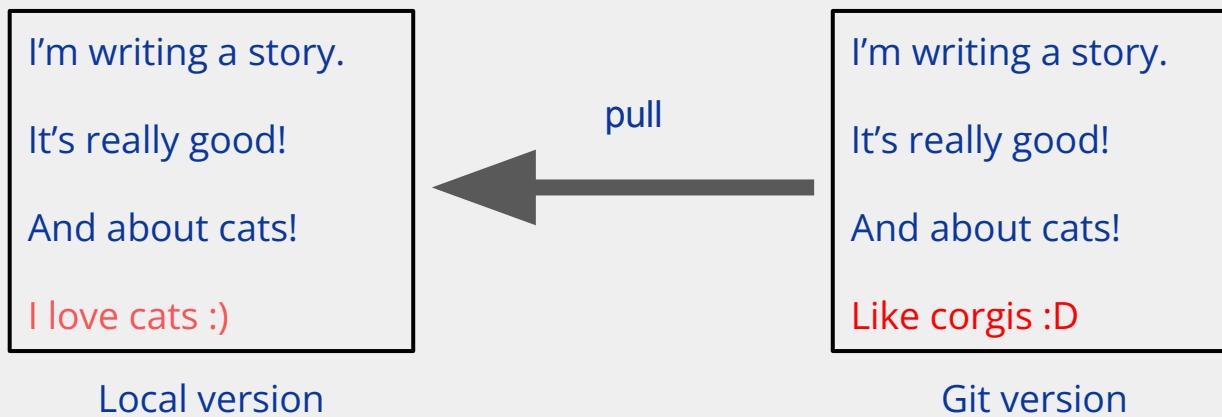


Scenario 1 – mydoc.txt



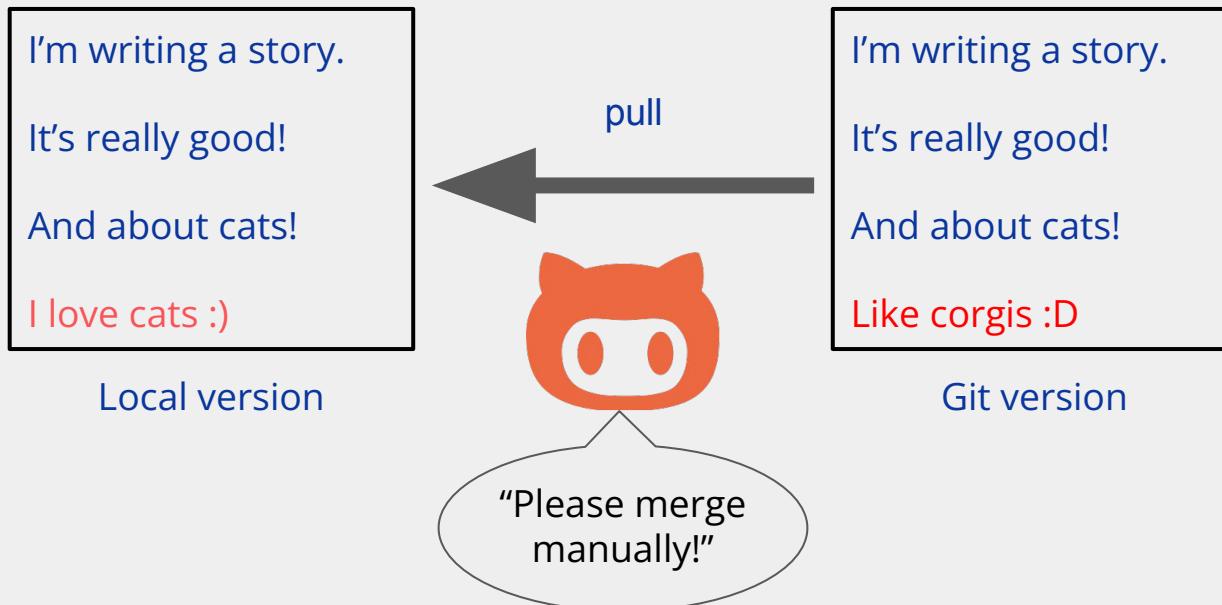
No conflicts – easy merge!

Scenario 2 – mydoc.txt

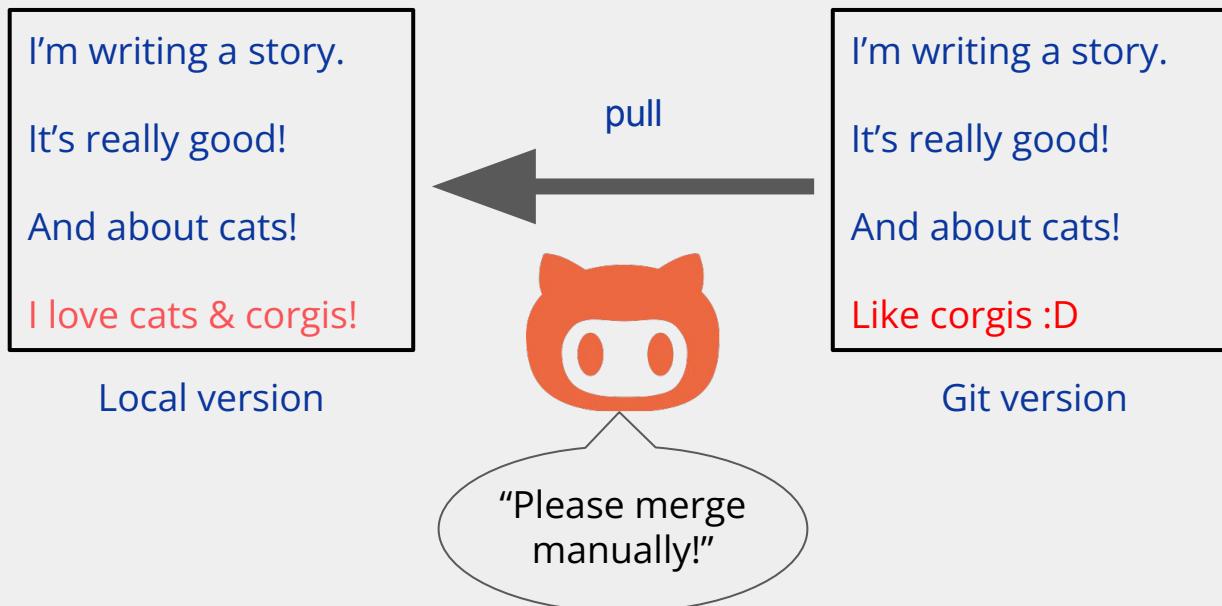


Merge conflict!

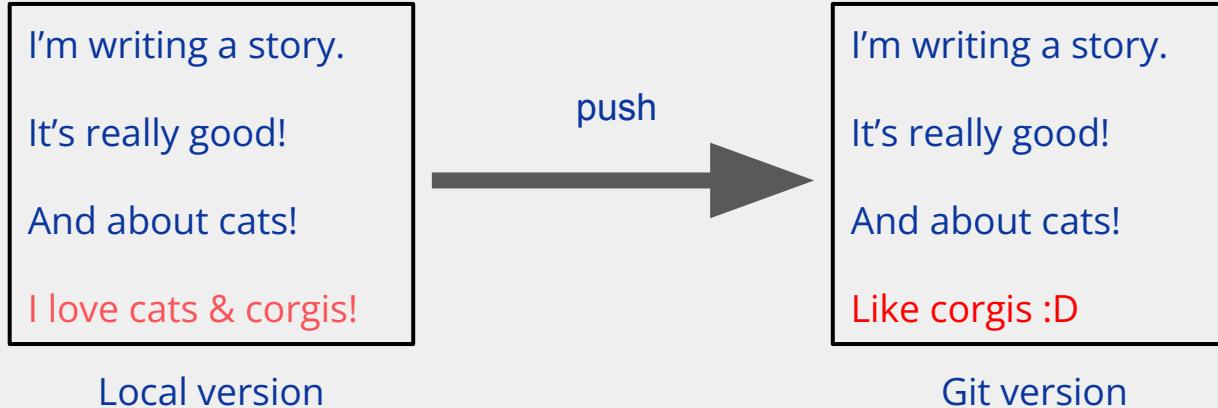
Scenario 2 – mydoc.txt



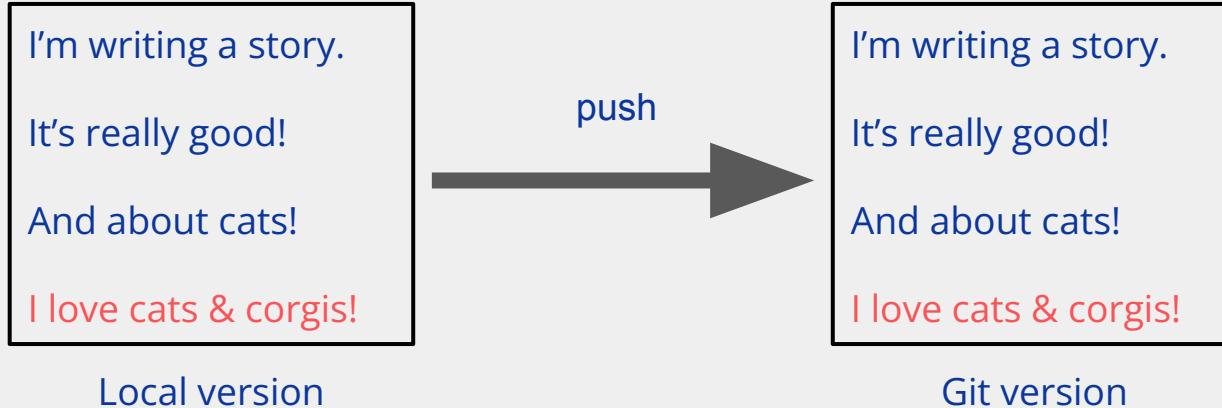
Scenario 2 – mydoc.txt



Scenario 2 – mydoc.txt



Scenario 2 – mydoc.txt



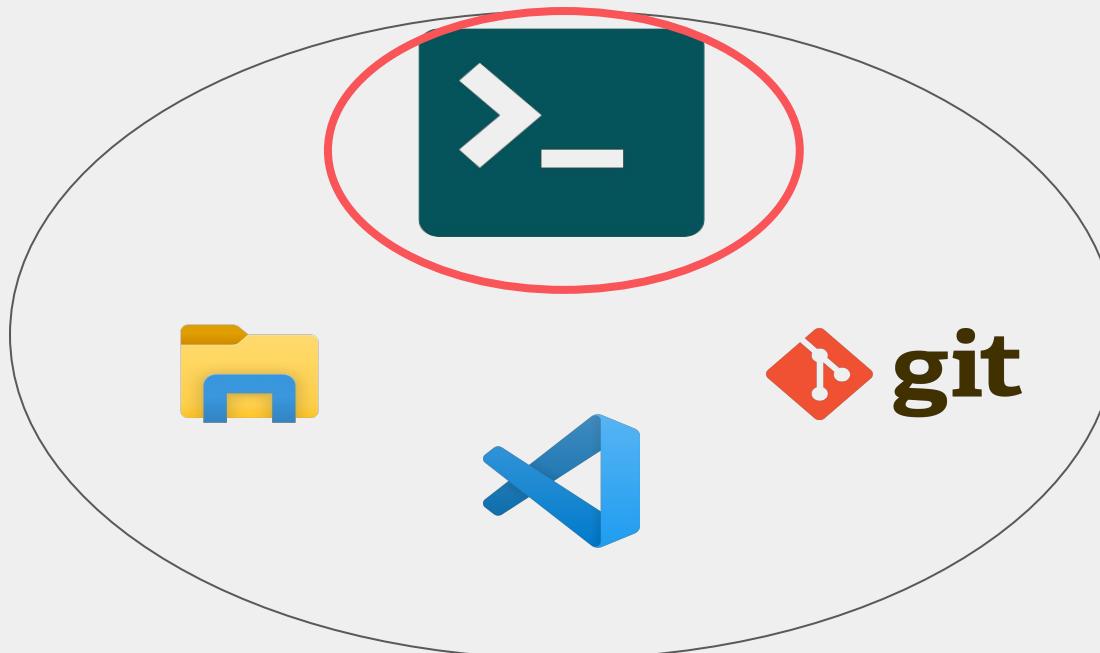
Problems with coding in projects

1. Synchronization of files
 - a. Push commits to a Git server (a single, unified source of truth)
 - b. Pull collaborator's changes before modifying on your computer
2. Collaboration of files
 - a. Use an algorithm to merge files based on commits.
 - b. Fall back on human intervention if algorithm fails (usually, different edits on same lines)
3. Version history
 - a. Keep a log of the changes to a file!
 - b. Allows us to easily revert to any previous version.

Woohoo!

So how do we actually use Git?

The command line



Computer's "secret menu"



Some useful terminology...

- **File**: a file (basic unit of information storage)
- **Directory**: a folder, which can hold files and other directories
- **Terminal**: the command line window (often used as a synonym for cmd)

In the command line...

- The “.” symbol is shorthand for “current directory”
- The “..” symbol is shorthand for “parent directory”
- The “~” symbol is shorthand for “home directory”

Demo!



kschen@kfc: ~



kschen@kfc:~\$ |



kschen@kfc: ~



kschen@kfc:~\$ |

“Who am I?”

(In this case it's showing {user}@{workstation}.)

```
kschen@kfc:~$ pwd
/home/kschen
kschen@kfc:~$ |
```

pwd: print working directory (“where am I?”)

Most terminals will also show your current directory here.

kschen@kfc: ~

```
kschen@kfc:~$ pwd
/home/kschen
kschen@kfc:~$ mkdir weblab ← mkdir: make directory
kschen@kfc:~$ |
```

```
kschen@kfc:~/weblab × + ▾ - □ ×
kschen@kfc:~$ pwd
/home/kschen
kschen@kfc:~$ mkdir weblab
kschen@kfc:~$ cd weblab ←
kschen@kfc:~/weblab$ |
```

cd: change directory (“move here”)

kschen@kfc: ~/weblab X + - □ ×

```
kschen@kfc:~$ pwd  
/home/kschen  
kschen@kfc:~$ mkdir weblab  
kschen@kfc:~$ cd weblab  
kschen@kfc:~/weblab$ pwd  
/home/kschen/weblab ← We are now in the newly-created weblab directory!  
kschen@kfc:~/weblab$ |
```

```
kschen@kfc: ~/weblab × + ▾
kschen@kfc:~$ pwd
/home/kschen
kschen@kfc:~$ mkdir weblab
kschen@kfc:~$ cd weblab
kschen@kfc:~/weblab$ pwd
/home/kschen/weblab
kschen@kfc:~/weblab$ clear| ← Let's reduce some clutter on the terminal and continue.
```



kschen@kfc: ~/weblab

X + <

- □ ×

```
kschen@kfc:~/weblab$ touch file.js ← touch: create a new file  
kschen@kfc:~/weblab$ |
```

```
kschen@kfc: ~/weblab      X  +  V
kschen@kfc:~/weblab$ touch file.js
kschen@kfc:~/weblab$ ls ←
file.js
kschen@kfc:~/weblab$ |
```

ls: list (“show me what’s in this directory”)

kschen@kfc: ~/weblab

```
kschen@kfc:~/weblab$ touch file.js
kschen@kfc:~/weblab$ ls
file.js
kschen@kfc:~/weblab$ touch file2.js
kschen@kfc:~/weblab$ ls
file.js  file2.js ←
kschen@kfc:~/weblab$ |
```

ls shows us what's in the directory!

```
kschen@kfc: ~/weblab × kschen@kfc: ~/weblab × | + ×  
kschen@kfc:~/weblab$ touch file.js  
kschen@kfc:~/weblab$ ls  
file.js  
kschen@kfc:~/weblab$ touch file2.js  
kschen@kfc:~/weblab$ ls  
file.js file2.js  
kschen@kfc:~/weblab$ mkdir newdir ← Let's make a new directory called newdir.  
kschen@kfc:~/weblab$ ls  
file.js file2.js newdir ← We can see it here!  
kschen@kfc:~/weblab$ |
```

```
kschen@kfc: ~/weblab
```

```
kschen@kfc: ~/weblab
```

```
kschen@kfc:~/weblab$ touch file.js
kschen@kfc:~/weblab$ ls
file.js
kschen@kfc:~/weblab$ touch file2.js
kschen@kfc:~/weblab$ ls
file.js file2.js
kschen@kfc:~/weblab$ mkdir newdir
kschen@kfc:~/weblab$ ls
file.js file2.js newdir
kschen@kfc:~/weblab$ mv file.js newdir
kschen@kfc:~/weblab$ ls
file2.js newdir
kschen@kfc:~/weblab$ ls newdir
file.js
kschen@kfc:~/weblab$ |
```

mv: move (“move this source to this destination”)

It's no longer in our current directory!

But if we peek inside *newdir*...it's there!

```
kschen@kfc: ~/weblab
```

```
kschen@kfc:~/weblab$ ls
```

```
file2.js  newdir
```

```
kschen@kfc:~/weblab$ mv file2.js file2blah ←
```

What happens if our destination isn't a directory?

```
kschen@kfc:~/weblab$ |
```

```
kschen@kfc: ~/weblab
```

```
kschen@kfc:~/weblab$ ls  
file2.js  newdir  
kschen@kfc:~/weblab$ mv file2.js file2blah  
kschen@kfc:~/weblab$ ls  
file2blah  newdir  
kschen@kfc:~/weblab$ |
```

The file is moved *from* the current directory, *to* the current directory – but with a different name...

Basically, just a rename!

```
kschen@kfc: ~/weblab      kschen@kfc: ~/weblab      - □ ×  
kschen@kfc:~/weblab$ ls  
file2.js  newdir  
kschen@kfc:~/weblab$ mv file2.js file2blah  
kschen@kfc:~/weblab$ ls  
file2blah  newdir  
kschen@kfc:~/weblab$ cp file2blah file2meh ← cp: copy ("copy this source to this destination")  
kschen@kfc:~/weblab$ ls  
file2blah  file2meh  newdir  
kschen@kfc:~/weblab$ |
```

Creates a copy of the file!



kschen@kfc: ~/weblab



kschen@kfc: ~/weblab

- | + < >

— □ ×

```
kschen@kfc:~/weblab$ ls
file2.js  newdir
kschen@kfc:~/weblab$ mv file2.js file2blah
kschen@kfc:~/weblab$ ls
file2blah  newdir
kschen@kfc:~/weblab$ cp file2blah file2meh
kschen@kfc:~/weblab$ ls
file2blah  file2meh  newdir
kschen@kfc:~/weblab$ |
```

Now, let's do some cleanup.

kschen@kfc: ~/weblab

```
kschen@kfc:~/weblab$ ls
file2.js  newdir
kschen@kfc:~/weblab$ mv file2.js file2blah
kschen@kfc:~/weblab$ ls
file2blah  newdir
kschen@kfc:~/weblab$ cp file2blah file2meh
kschen@kfc:~/weblab$ ls
file2blah  file2meh  newdir
kschen@kfc:~/weblab$ rm file2blah file2meh ←
kschen@kfc:~/weblab$ ls
newdir
kschen@kfc:~/weblab$ |
```

rm: remove (“remove file(s)”) ←

```
kschen@kfc: ~/weblab
```

```
kschen@kfc:~/weblab$ ls  
file2.js  newdir  
kschen@kfc:~/weblab$ mv file2.js file2blah  
kschen@kfc:~/weblab$ ls  
file2blah  newdir  
kschen@kfc:~/weblab$ cp file2blah file2meh  
kschen@kfc:~/weblab$ ls  
file2blah  file2meh  newdir  
kschen@kfc:~/weblab$ rm file2blah file2meh  
kschen@kfc:~/weblab$ ls  
newdir  
kschen@kfc:~/weblab$ rmdir newdir
```

rmdir: remove directory

```
rmdir: failed to remove 'newdir': Directory not empty
```

We can't use rmdir on a non-empty directory. What sequence of commands should we use to empty newdir?

```
kschen@kfc: ~/weblab × kschen@kfc: ~/weblab × | + ×
kschen@kfc:~/weblab$ rmdir newdir
rmdir: failed to remove 'newdir': Directory not empty
kschen@kfc:~/weblab$ cd newdir ← Go into newdir
kschen@kfc:~/weblab/newdir$ ls
file.js
kschen@kfc:~/weblab/newdir$ rm file.js ← Remove the file in newdir
kschen@kfc:~/weblab/newdir$ cd .. ← Go back to the parent directory (weblab)
kschen@kfc:~/weblab$ ls
newdir
kschen@kfc:~/weblab$ rmdir newdir ← Now we can remove newdir!
kschen@kfc:~/weblab$ ls
kschen@kfc:~/weblab$ |
```

Recap

pwd	where am I?
ls	list files/directories
cd <directory>	change directory
mkdir <name>	make directory with specified name
mv <source> <destination>	move from source to destination
cp <source> <destination>	copy from source to destination
rm <file>	remove (<i>permanently delete</i>) file
rmdir <directory>	remove directory

So let's try using Git!

```
weblab-test — albertbowenxing@C02D51K7MD6M — -zsh — 94x30
..t/weblab-test

[→ test mkdir weblab-test
[→ test cd weblab-test
[→ weblab-test touch index.html
[→ weblab-test git init
Initialized empty Git repository in /Users/albertbowenxing/test/weblab-test/.git/
[→ weblab-test git:(master) x
```

weblab-test — albertbowenxing@C02D51K7MD6M — -zsh — 94x30
./t/weblab-test

```
[→ weblab-test git:(master) ✘ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

nothing added to commit but untracked files present (use "git add" to track)
→ weblab-test git:(master) ✘ ]
```

```
weblab-test ~ albertbowenxing@C02D51K7MD6M -- zsh -- 94x30
./t/weblab-test

[→ weblab-test git:(master) ✘ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

nothing added to commit but untracked files present (use "git add" to track)
[→ weblab-test git:(master) ✘ git add index.html
[→ weblab-test git:(master) ✘ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html

[→ weblab-test git:(master) ✘ git commit -m "oMG i mAdE mY fIRsT Git CoMMIT"
[master (root-commit) 9e23323] oMG i mAdE mY fIRsT Git CoMMIT
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 index.html
[→ weblab-test git:(master) ✘
```

```
weblab-test — albertbowenxing@C02D51K7MD6M — -zsh — 94x30
./t/weblab-test

[→ weblab-test git:(master) git status
On branch master
nothing to commit, working tree clean
[→ weblab-test git:(master) git log
commit 9e233234851e8d23e44781775c948fdc16e5d94e (HEAD → master)
Author: Albert B Xing <axing@mit.edu>
Date:   Thu Dec 30 00:20:04 2021 -0500

    oMG i mAdE mY fIRsT Git ComMIT
[→ weblab-test git:(master)]
```

Now, we find a space on a Git server to store all this information.

The world's leading software development platform · GitHub · Google Chrome
github.com



Why GitHub? Enterprise Explore Marketplace Pricing

Search GitHub

Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more](#).

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.



GitHub x +

github.com

Search or jump to... ✓

Pull requests Issues Marketplace Explore

xingalbert

Repositories New

Find a repository...

- JinayJain/hackmit
- xingalbert/resume
- xingalbert/xingalbert.github.io
- xingalbert/testneuromorphism
- xingalbert/6.08-Final-Project
- xingalbert/weblab-project
- xingalbert/hello-world

Show more

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Your teams

Find a team...

weblab-class/penguins-n-phriends

All activity

JinayJain starred JinayJain/timekeeper 12 days ago

JinayJain/timekeeper
Telling the time from an image with computer vision (PyTorch + OpenCV)

Python ⭐ 3 Updated Dec 25

JinayJain made JinayJain/deep-racing public 20 days ago

JinayJain/deep-racing
Self-driving racecar using reinforcement learning (proximal policy optimization) in PyTorch

Python Updated Dec 28

daniel-moon32 started following gracewzhang 21 days ago

Grace Zhang gracewzhang
4 repositories 11 followers

Follow

legitatx starred mehdihadeli/awesome-software-architecture 24 days ago

mehdihadeli/awesome-software-architecture
A curated list of awesome articles, videos, and other resources to learn and practice about software architecture, patterns, and principles.

⭐ 6.6k Updated Dec 28

legitatx starred mjhea0/awesome-fastapi 28 days ago

mjhea0/awesome-fastapi
A curated list of awesome things related to FastAPI

Star

Welcome to GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

Go to Global Campus

Explore repositories

python-gino/gino
GINO Is Not ORM - a Python asyncio ORM on SQLAlchemy core.

Python ⭐ 2.3k

teamhide/fastapi-boilerplate
FastAPI boilerplate for real world production

Python ⭐ 91

sysid/sse-starlette
Starlette-based Server-Sent Events (SSE) implementation

Python ⭐ 145

Explore more →



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template
Start your repository with a template repository's contents.

No template ▾

Owner *  xingalbert / **Repository name *** 

Great repository names are short and memorable. Need inspiration? How about [symmetrical-octo-garbanzo](#)?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

xingalbert/weblab-test

github.com/xingalbert/weblab-test

Search or jump to... Pull requests Issues Marketplace Explore

xingalbert / weblab-test Public

Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:xingalbert/weblab-test.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# weblab-test" >> README.md  
git init  
git add README.md  
git commit -m "first commit"  
git branch -M main  
git remote add origin git@github.com:xingalbert/weblab-test.git  
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:xingalbert/weblab-test.git  
git branch -M main  
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

ProTip! Use the URL for this page when adding GitHub as a remote.

So now we go back to our command
line...

The image shows a dark-themed terminal window with a black background and white text. The title bar at the top reads "weblab-test — albertbowenxing@C02D51K7MD6M — -zsh — 94x30". Below the title bar, the command ".t/weblab-test" is displayed. In the main area, two lines of text are visible: "weblab-test git:(master) git remote add origin git@github.com:xingalbert/weblab-test.git" and "weblab-test git:(master)". The text is in a monospaced font, and the cursor is positioned at the end of the second line.

```
weblab-test git:(master) git remote add origin git@github.com:xingalbert/weblab-test.git
weblab-test git:(master)
```

A screenshot of a macOS terminal window titled "weblab-test". The window has a dark theme. The command ".t/weblab-test" was run, followed by the command "git push --set-upstream origin master". The output shows the process of pushing three objects to the "master" branch of the "github.com:xingalbert/weblab-test.git" repository. The push is successful, and the local "master" branch is set up to track the remote "master" branch.

```
weblab-test git:(master) git push --set-upstream origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 234 bytes | 234.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:xingalbert/weblab-test.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
weblab-test git:(master)
```

Now if we refresh our [github.com](#)
page...



xingalbert/weblab-test

github.com/xingalbert/weblab-test

Search or jump to... ?

Pull requests Issues Marketplace Explore

Unwatch 1 Fork 0 Star 0

xingalbert / weblab-test Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

 xingalbert oMG i mAdE mY fIRsT Git ComMIT 9e23323 17 minutes ago 1 commit

 index.html oMG i mAdE mY fIRsT Git ComMIT 17 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

No description, website, or topics provided.

0 stars 1 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)



© 2021 GitHub, Inc.

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing

API

Training

Blog

About

Awesome! Now let's go sleep.

A day later...

A screenshot of a dark-themed code editor interface, likely Visual Studio Code. The interface includes a top bar with window control buttons (red, yellow, green), a title bar reading "index.html — weblab-test", and a bottom status bar showing file details like "Ln 1, Col 1" and encoding "UTF-8".

The left sidebar contains the "EXPLORER" view, which lists "OPEN EDITORS" (containing "index.html") and a folder "WEBLAB-TEST" also containing "index.html". Below these are icons for "Search", "File Explorer", "Terminal", "Git", "Recent Files", and "Help". A dropdown menu from the "Help" icon lists "OUTLINE", "TIMELINE", and "TYPESCRIPT TUTOR".

The main workspace shows a single open editor tab for "index.html" with the number "1" at the top left. The editor area is currently empty.

A screenshot of a Microsoft Edge browser window displaying a code editor interface. The title bar reads "index.html — weblab-test". The left sidebar contains icons for file operations, search, repository, and user profile, with a gear icon showing a "1" indicating notifications. The main area shows an "EXPLORER" sidebar with "OPEN EDITORS" and "WEBLAB-TEST" sections, both listing "index.html" with a modified status indicator "M". The main editor pane displays the following HTML code:

```
<h1>is this how you write a website?</h1>
<div>I have three flamboyant plumbers</div>
```

The bottom status bar shows "Ln 2, Col 44" and "Spaces: 4" along with file format indicators "UTF-8", "LF", "HTML", and "Prettier".

A screenshot of a macOS terminal window titled "weblab-test". The window shows the command ".t/weblab-test" being run. The output of the "git status" command is displayed, indicating the user is on the "master" branch and their local changes are up-to-date with "origin/master". It also lists a modified file, "index.html".

```
weblab-test git:(master) ✘ git status
On branch master
Your branch is up to date with 'origin/master'.

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

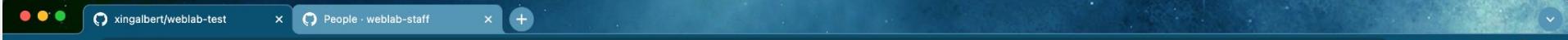
no changes added to commit (use "git add" and/or "git commit -a")
→ weblab-test git:(master) ✘
```

A screenshot of a macOS terminal window titled "weblab-test". The window shows a command-line session where a user is committing changes to a local repository. The session starts with ".t/weblab-test" and ends with a successful commit message.

```
.t/weblab-test
[weblab-test git:(master) ✘ git add .
[weblab-test git:(master) ✘ git commit -m "i mADe a wEBsiTE"
[master aac6022] i mADe a wEBsiTE
 1 file changed, 2 insertions(+)
[weblab-test git:(master) ✘
```

```
weblab-test — albertbowenxing@C02D51K7MD6M — zsh — 94x30
./t/weblab-test

→ weblab-test git:(master) git push
To github.com:xingalbert/weblab-test.git
! [rejected]      master -> master (fetch first)
error: failed to push some refs to 'git@github.com:xingalbert/weblab-test.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
→ weblab-test git:(master)
```



Search or jump to...

Pull requests Issues Marketplace Explore

Unwatch 1 Fork 0 Star 0

xingalbert / weblab-test Public

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

master

1 branch

0 tags

Go to file

Add file

Code



kevinsunchen Kevin's Commit ...

f8d3f6e 21 minutes ago 2 commits

index.html

Kevin's Commit

21 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

- xingalbert Albert Xing
- kevinsunchen kevin chen

```
weblab-test — albertbowenxing@C02D51K7MD6M — zsh — 94x30
./t/weblab-test

[→ weblab-test git:(master) git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:xingalbert/weblab-test
  9e23323..f8d3f6e master      -> origin/master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
[→ weblab-test git:(master) x]
```

index.html — weblab-test

EXPLORER ...

OPEN EDITORS index.html !

WEBLAB-TEST index.html !

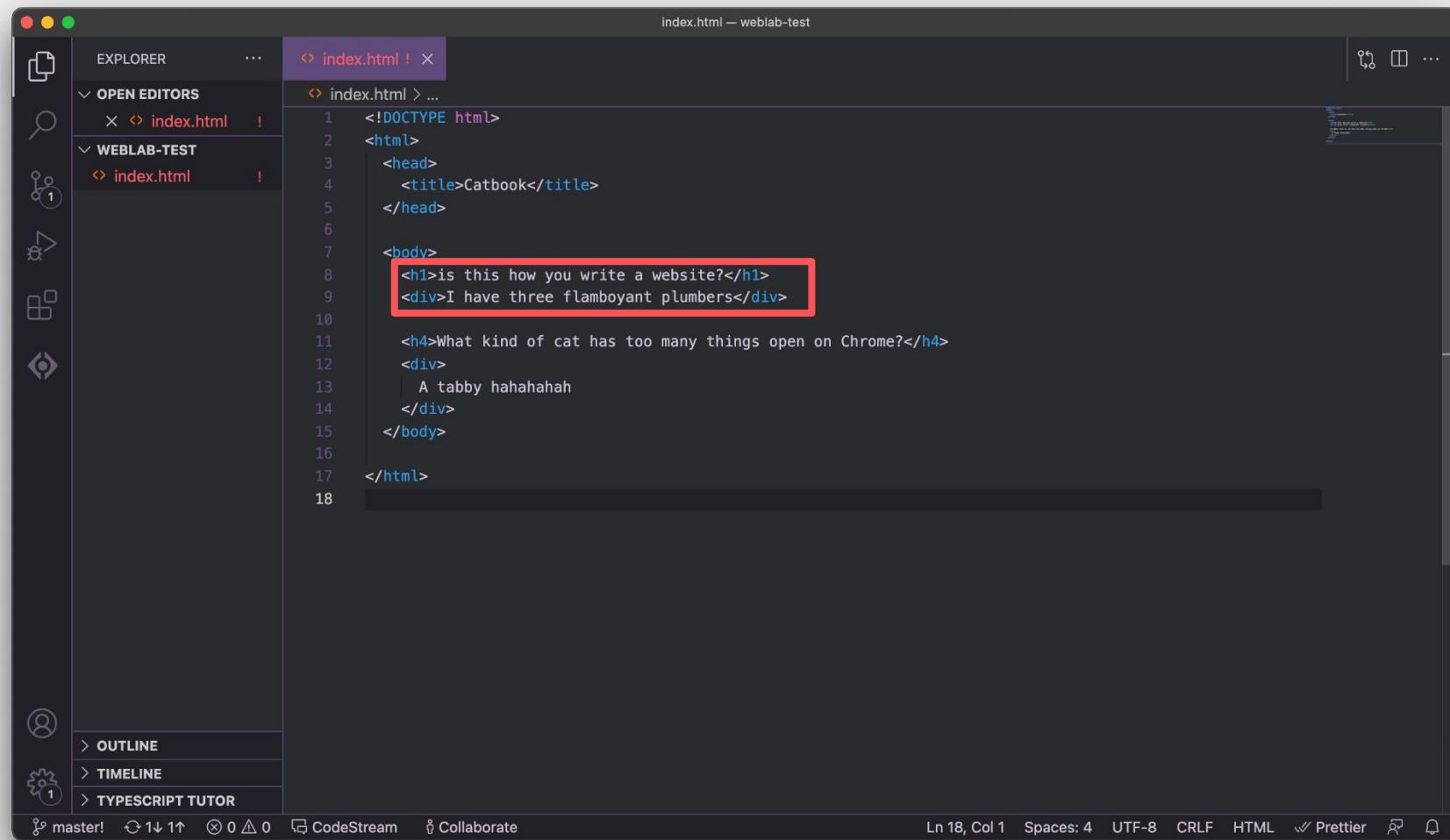
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

```
<<<<< HEAD (Current Change)
1 <h1>is this how you write a website?</h1>
2 <div>I have three flamboyant plumbers</div>
3 =====
4 <!DOCTYPE html>
5 <html>
6   <head>
7     <title>Catbook</title>
8   </head>
9
10  <body>
11    <h4>What kind of cat has too many things open on Chrome?</h4>
12    <div>
13      A tabby hahahah
14    </div>
15  </body>
16
17 </html>
18 >>>>> f8d3f6efd96ef2cdcca8885cc314609c39ed3fa5 (Incoming Change)
19
20
```

My commit

Kevin's commit (from remote)

master! 1↓↑ 0△0 CodeStream Collaborate Ln 20, Col 1 Spaces: 4 UTF-8 CRLF HTML Prettier



index.html – weblab-test

EXPLORER ...

OPEN EDITORS

- index.html !

WEBLAB-TEST

- index.html !

```
<!DOCTYPE html>
<html>
  <head>
    <title>Catbook</title>
  </head>
  <body>
    <h1>is this how you write a website?</h1>
    <div>I have three flamboyant plumbers</div>
  <h4>What kind of cat has too many things open on Chrome?</h4>
  <div>
    | A tabby hahahahah
  </div>
</body>
</html>
```

OUTLINE

TIMELINE

TYPESCRIPT TUTOR

master! 1 1 0 0 CodeStream Collaborate Ln 18, Col 1 Spaces: 4 UTF-8 CRLF HTML Prettier

```
weblab-test — albertbowenxing@C02D51K7MD6M — zsh — 94x30
./t/weblab-test

[→ weblab-test git:(master) ✘ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")
[→ weblab-test git:(master) ✘ ]
```

```
weblab-test — albertbowenxing@C02D51K7MD6M — -zsh — 94x30
./t/weblab-test

→ weblab-test git:(master) ✘ git add index.html
→ weblab-test git:(master) ✘ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
    modified:   index.html

→ weblab-test git:(master) ✘
```

```
weblab-test - albertbowenxing@C02D51K7MD6M -- zsh - 94x30
./t/weblab-test

[→ weblab-test git:(master) ✘ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  modified:   index.html

[→ weblab-test git:(master) ✘ git commit -m "Fix merge conflict"
[master 9fdb104] Fix merge conflict
[→ weblab-test git:(master) ✘ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 685 bytes | 685.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:xingalbert/weblab-test.git
  f8d3f6e..9fdb104  master -> master
[→ weblab-test git:(master) ✘
```

```
weblab-test git:(master) git log
commit 9fdb104b7979044b876cc61fe2928cc01511ad51 (HEAD -> master, origin/master)
Merge: aac6022 f8d3f6e
Author: Albert B Xing <axing@mit.edu>
Date:   Thu Dec 30 01:34:27 2021 -0500

    Fix merge conflict

commit aac60226ed00c424c8acdb0543324a23dd3d7a46
Author: Albert B Xing <axing@mit.edu>
Date:   Thu Dec 30 01:21:18 2021 -0500

    i mADe a wEBsiTE

commit f8d3f6efd96ef2cdcca8885cc314609c39ed3fa5
Author: kevin chen <kschen@mit.edu>
Date:   Thu Dec 30 01:02:51 2021 -0500

    Kevin's Commit

    :))

commit 9e233234851e8d23e44781775c948fdc16e5d94e
Author: Albert B Xing <axing@mit.edu>
Date:   Thu Dec 30 00:20:04 2021 -0500

    oMG i mAdE mY fIRsT Git ComMIT
→ weblab-test git:(master)
```

Merge commit

My second commit

Kevin's commit

My first commit



Search or jump to...

Pull requests Issues Marketplace Explore



xingalbert/weblab-test

Public

Unwatch 1

Fork 0

Star 0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

master

1 branch

0 tags

Go to file

Add file

Code



xingalbert Fix merge conflict

9fdb104 18 minutes ago 4 commits



index.html

18 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

xingalbert Albert Xing

kevinsunchen kevin chen

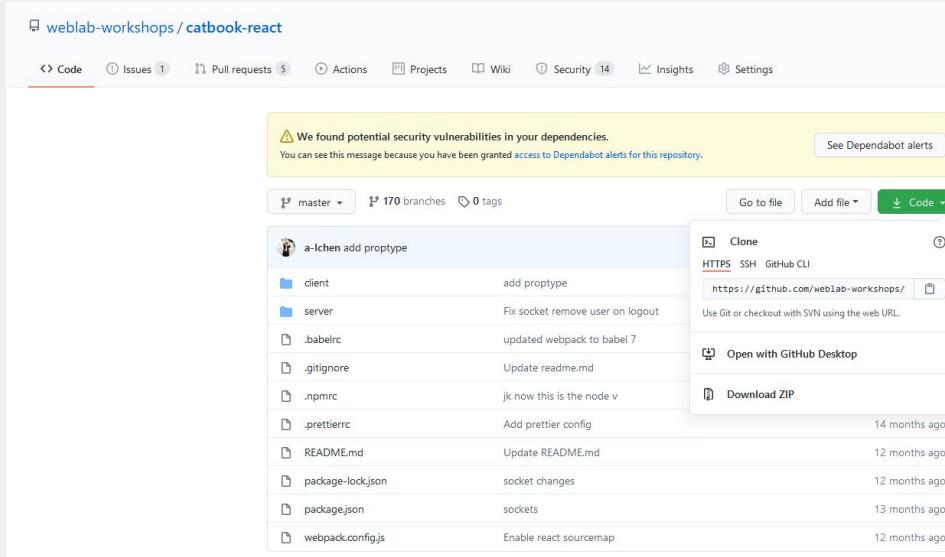
PHEW

One Last Thing....

Git the Workshop!

weblab.to/catbook

Cloning the repository!

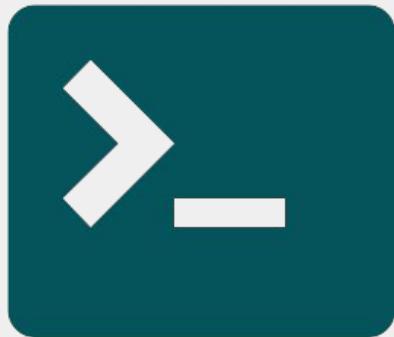


The screenshot shows a GitHub repository page for 'weblab-workshops / catbook-react'. The page includes a yellow warning box about potential security vulnerabilities in dependencies. Below the header, there are buttons for 'Code' (selected), 'Issues 1', 'Pull requests 5', 'Actions', 'Projects', 'Wiki', 'Security 14', 'Insights', and 'Settings'. A dropdown menu shows the 'master' branch is selected. The main area displays a list of files and their commit history. On the right side, a 'Clone' button is highlighted, with a tooltip showing 'HTTPS' and 'GitHub CLI' options, along with the URL 'https://github.com/weblab-workshops/catbook-react'. Below the clone button are links for 'Open with GitHub Desktop' and 'Download ZIP'. The commit history shows several recent changes by 'a-Ichen'.

File	Commit Message	Date
client	add proptype	14 months ago
server	Fix socket remove user on logout	12 months ago
.babelrc	updated webpack to babel 7	12 months ago
.gitignore	Update readme.md	13 months ago
.npmrc	jk now this is the node v	12 months ago
.prettierrc	Add prettier config	14 months ago
README.md	Update README.md	12 months ago
package-lock.json	socket changes	12 months ago
packagejson	sockets	13 months ago
webpack.config.js	Enable react sourcemap	12 months ago

- 1
- 2

Recap



git

Recap pt 2

To create a local repository:

```
git init  
git add <file>  
git commit -m "<commit message>"
```

To set the server address for a local repository:

```
git remote add origin <origin.git link>  
git push -u origin master
```

To send changes to the Git server:

```
git add <file>  
git commit -m "<commit message>"  
git push
```

To get updated files from server:

```
git pull  
(may need to resolve conflicts here)
```

Info commands:

```
git status  
git log
```

Cheatsheets

<https://weblab.mit.edu/resources/>

Next:

Lunch Break
HTML/CSS with Thanh!