

An Introduction to Git

Git gud



BuildingBloCS September Conference

Hello!

I'm Saumil

- I'm a secondary 3 student at Bukit Batok Secondary School
- This workshop will be my second workshop
- I'm mostly a iOS dev. Download hit app Grass on the App Store!!



Hello!

I'm Yuhang

- Year 5 at Anglo-Chinese School Independent
- Does not actually take computing in school
- Likes penguins. Like, a lot.
- Mainly does front-end app development



Hello!

I'm Brayden

- Also Year 5 at Anglo-Chinese School Independent
- First time as workshop speaker
- Plays a bit too much CS2 (5hrs/day)
- Mainly focused on web development, with some dabbling in controls engineering.



Workshop Objectives

You will leGIT learn all of this

- Be adept at moving around directories in your terminal
- Be proficient in basic Git repository management
- Collaboration with others using GitHub



Agenda

- What is Git?
- Configuring Git CLI
- Version control commands
 - Hands on with a picture of a cat
- Branches
 - Hands-on git branching + branching
- Recap

What is Git?

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



Textbook definitions aren't always the same for programmers

Dictionary
Definitions from [Oxford Languages](#) · [Learn more](#)

 **git**
/git/

noun ▲ DEROGATORY • INFORMAL
noun: **git**; plural noun: **gits**

an unpleasant or contemptible person.
"that mean old git"

Origin

1930s: variant of get (sense 2 of the noun).

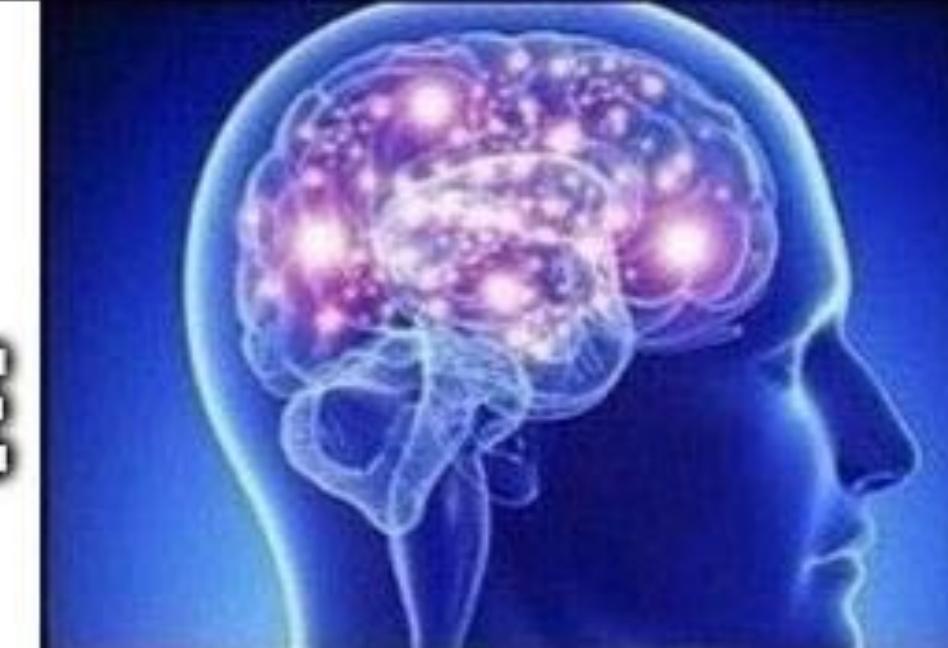
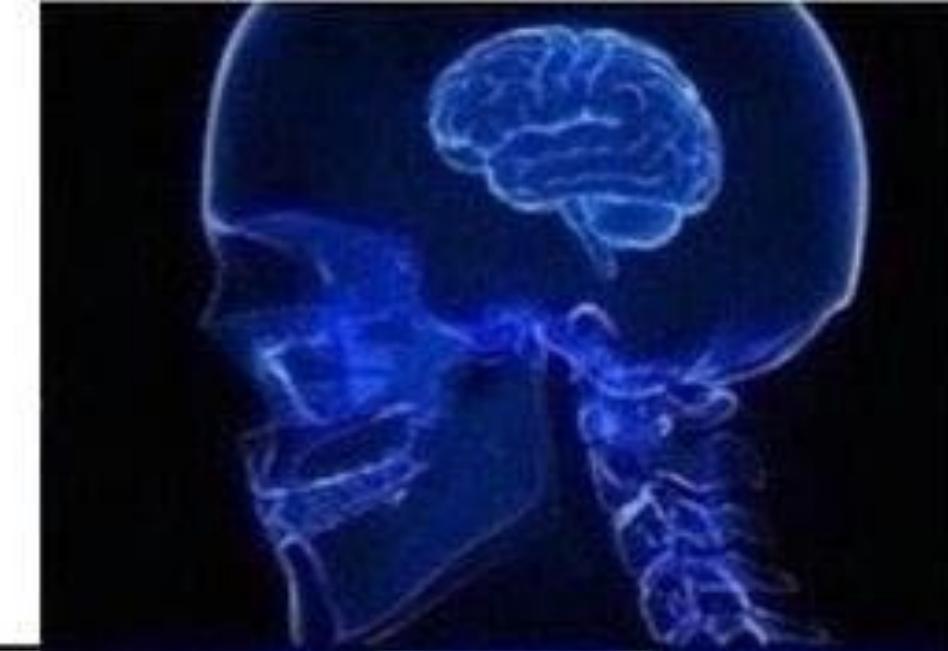
When
a docu-
school

**USING
AIRDROP**

**USING
GOOGLE DRIVE**

**USING A
PRIVATE DISCORD**

**USING
GITHUB**



rate on
er in
u use?

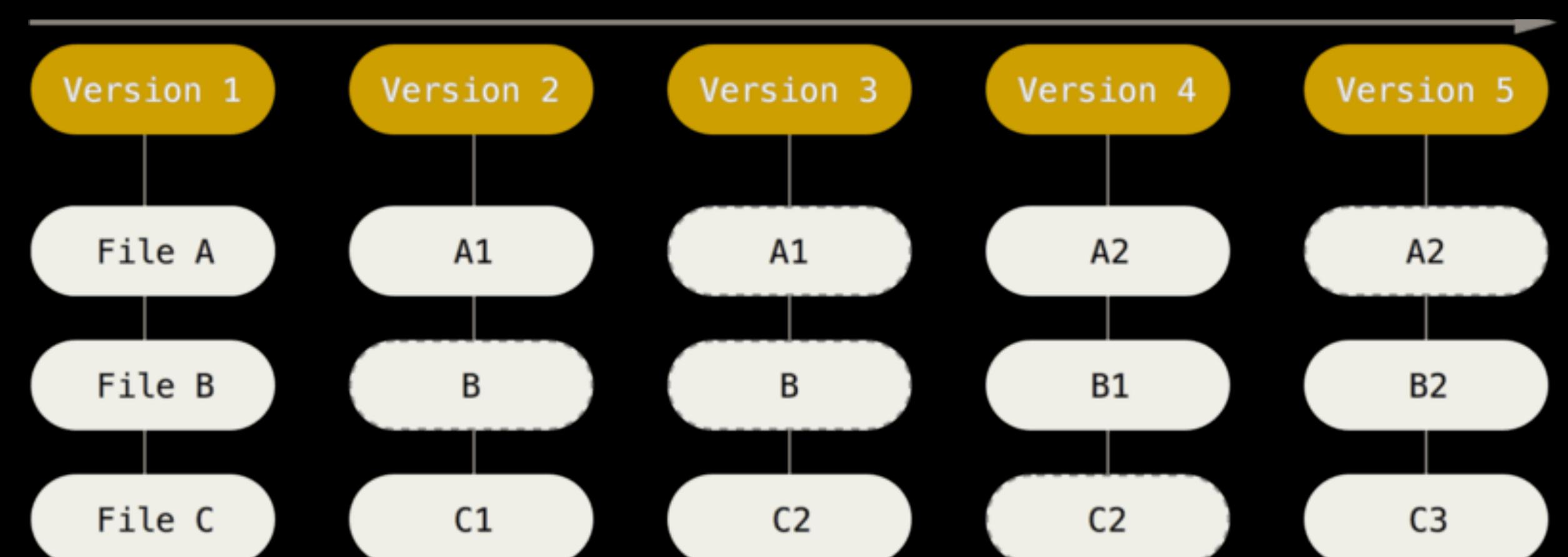
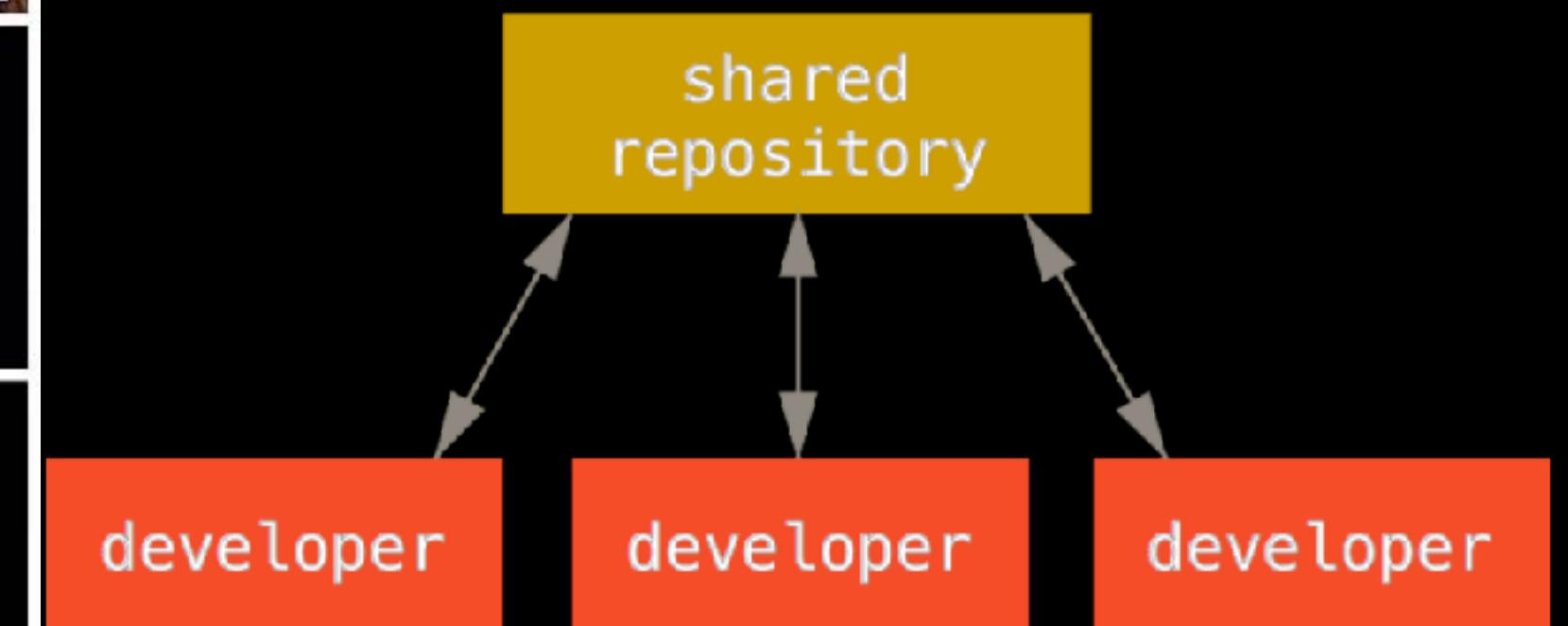
What is Git?

- It allows you to share your code
- It is a content tracker
- More importantly its a fast, scalable and distributable version control system
- Git by itself ≠ GitHub/BitBucket/GitLab/Azure devOps/Beanstalk
 - Although aforementioned services function using Git they aren't Git by themselves as Git is simply a file tracking system



More on Git

- Git functions by hosting a repository
 - Think of this repository as something that takes detailed "snapshots" of your code
 - This means that Git doesn't overwrite unchanged files and only changes the files you change



Git Vs GitHub

- Github is a Git repository host.
- Provides a GUI for Git functions
- GitHub is the easiest to use (and most accessible of all other git providers)
- So how do we use GitHub?



Crash course on GitHub time!!

A close-up photograph of a young man with dark hair and glasses, wearing a blue jacket over a red sweater. He is looking slightly to his left with a thoughtful expression. The background is blurred, showing what appears to be an office or study environment.

What is
GitHub?

More on Github

- GitHub is much more than code.
- Discussions
- Communities
- Organisations
- Co-pilot
- Campus experts



Arctic Code Vault

(You're supposed to show off
[GitHub.com](#) now)

GitHub Profiles

- Think of this as a LinkedIn or a portfolio
- Any/all tech hiring people will look at your profile to judge you more!

The image shows two GitHub profiles side-by-side. On the left is the profile of a user named **Sim Yu Bin Brayden**, which features a large circular profile picture of a brown tabby cat. The profile includes sections for 'Popular repositories', 'Contribution activity' (showing contributions from June to December 2024), and 'Highlights'. On the right is the profile of a user named **saumilthecode**, which features a circular profile picture of a dog. This profile includes sections for 'Popular repositories', 'Achievements', 'Highlights', and 'Contribution activity' (showing contributions from December 2020 to December 2024).

Overachievers (you don't need to be like them)

cytronicoder

Overview Repositories 18 Projects Packages Stars 38

Type / to search



Zeyu Yao
cytronicoder · he/him

Unfollow

I occasionally work on bioinformatics research and lead Hack Club shenanigans.

64 followers · 39 following

@aimed-lab
Singapore
20:20 - same time
novodoodle@gmail.com
<https://www.cytronicoder.com/>
<https://orcid.org/0009-0000-4920-0159>
@cytronicoder
cytronical
in/cytronicoder

cytronicoder / README.md

README undergoing renovation; check back soon!

Hi there! I'm Peter. 🚀

👋 I'm a student developer currently studying in Singapore. I started programming at 13, and in the past year, I've built a website that [tracks more than 1.8k available taxis](#) in Singapore, worked with AI to create a fully functional [LSTM model](#) that can predict stock prices, and did so much more.

Achievements

- Platinum division in USACO
- Received Honor Roll of Distinction (top 1%) for the AMC 8 and Certificate of Achievement for AMC 10
- First place in Singapore for
 - BEBRAS International Challenge on Informatics and Computational Thinking (BEBRAS, 2022)
 - Singapore Math Global Assessments Final (SINGA, 2022)
 - Design Thinking with Robotics and Computational Thinking International Competition Finals (DrCT, 2021)
 - Singapore and Asian Schools Math Olympiad (SASMO, 2023)

Experience

- Founder of the SAIS Hack Club, aiming to provide SAIS students with an experience that inspires, educates, and develops future leaders in computer science. We currently have 20+ members working on projects and learning new skills!
 - Check out the [State of the SAIS Hackers 2023: Part 1](#), a recap of what we have done and are working on in the future.
- Current head of the outreach team at the [International Junior Honour Society Student Leadership Council](#); currently working to provide support and guidance to IJHS members involved in CSPs through an online mentorship program
 - Former head of the Development and Implementation team; organised workshops, webinars, and community events aimed at improving the welfare of IJHS fellows
 - Former head of the web development team at the SLC; spearheaded in deploying the IJHS community website to Wordpress

Projects I've worked on

- Connect 4 solver during [Fundamentals of Computer Science](#); it got beaten by another algorithm so then I made [a better one](#) (takes a while to run for some initial position though)

Overachievers (you don't need to be like them)

★ PRO

Organizations



Block or Report

Active 🔥

- [Cogniforge](#) - learning for all through collaborative projects and peer-to-peer collaboration
- [Various Hack Club shenanigans](#) in my scrapbook

In the back burner 😓

- [Notify](#), a site where you can write "notes" for other people to sign and leave a message
- A [case for the Sprig console](#) to make it into an actual, portable gaming controller

🌐 Things I work with

✍ Design



🌐 Languages and frameworks



NEXT

📦 Package managers



👉 Editors



🚀 Deployment



🤝 Collaboration

I believe that collaborating on projects is a great way to learn from each other and build up experience; feel free to hit me up on Discord (cytronicoder#4975) or email me at novodoodle@gmail.com if you want to work on a project together.

📢 Socials



I'm sorry for the accent in the
next video

Shell 6 5

C 1

44 contributions in the last year

2024



Learn how we count contributions

Less More

2023

2022

2021

2020

2019

Contribution activity

December 2024



2 contributions in private repositories

Dec 3



Projects

Well documented projects with good readme.md's

The image shows two GitHub repository pages side-by-side.

Astro Starter Kit: Minimal (Left Repository)

- Code** tab is active.
- Commits** section shows 128 commits by **icepcp**, with the latest commit being "add figma and gh as partners" (c28c8d3) 2 weeks ago.
- Files** section lists files like `package.json`, `pnpm-lock.yaml`, `vite.config.ts`, etc.
- README** section contains the following text:

```
npm create astro@latest -- --template minimal
```
- Contributors**: Seasoned astronaut? Delete this file. Have fun!
- Project Structure** button is visible.

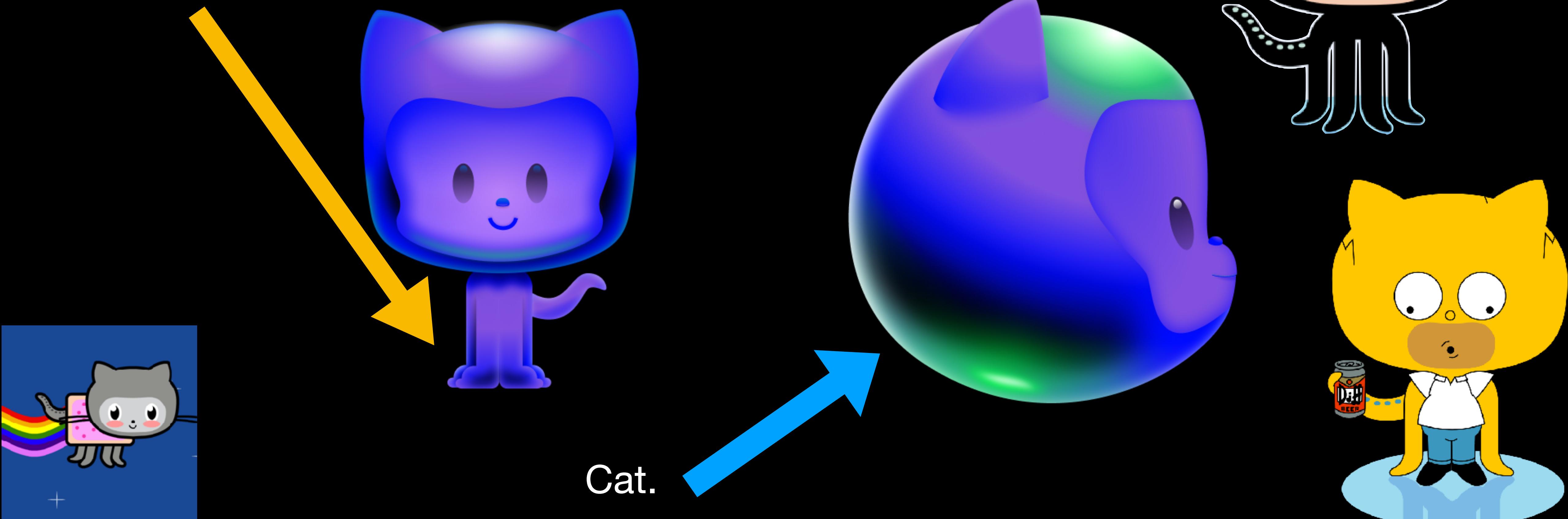
Size Limit (Right Repository)

- About** section:
 - BuildingBloCS's Newest Website!
 - buildingblocs.sg**
 - Readme
 - Activity
 - Custom properties
 - 4 stars
 - 2 watching
 - 3 forks
 - Report repository
- File List**: package.json, pnpm-lock.yaml, pnpm-workspace.yaml, vite.config.ts.
- README** section:
 - Size Limit** by Cult of Martians
 - Size Limit is a performance budget tool for JavaScript. It checks every commit on CI, calculates the real cost of your JS for end-users and throws an error if the cost exceeds the limit.
 - ES modules and tree-shaking support.
 - Add Size Limit to GitHub Actions, Circle CI or another CI system to know if a pull request adds a massive dependency.
 - Modular to fit different use cases: big JS applications that use their own bundler or small npm libraries with many files.
 - Can calculate the time it would take a browser to download and execute your JS. Time is a much more accurate and understandable metric compared to the size in bytes.
 - Calculations include all dependencies and polyfills used in your JS.
- Contributors**: 8 contributors, +80 contributors.
- Deployments**: 7 deployments, +2 deployments.
- Languages**: JavaScript 98.2%, TypeScript 51.2%, Astro 43.1%, JavaScript 3.2%, CSS 2.5%.

Octocat - (the main one is called Mona)

Many many different forms!

8 legs represent interconnectedness
and collaboration within the GitHub
ecosystem



Since the Octocat is a registered trademark of GitHub, using an Octocat as your logo or icon is not allowed.

Can I use an octocat as my avatar?

You can use an octocat as your personal avatar, but not for your company or a product you're building. You are welcome to show your <3 for GitHub, but not appear as if you == GitHub.

Can I submit an octocat to the Octodex?

We make quite a few octocats here at GitHub. Since they probably wouldn't be seen by more than a few people internally, we created the Octodex as a way to show them to the world. Everything on this list is official GitHub artwork and is under GitHub's trademark license, so we only include submissions from people affiliated with GitHub.

Can I create a product containing an octocat?

You may not use an octocat, created by GitHub or by you, for products or merchandise without written permission from GitHub. This includes, but is not limited to t-shirts, toys, stickers, etc.

Do all the octocats have a purpose?

Not all of them. The original list was comprised entirely of octocats that were used for legitimate purposes, but over time the list began to fill up with nonsense. By the time [memes](#) started to make their way onto the list, we knew it was hopeless to try to keep them all practical.

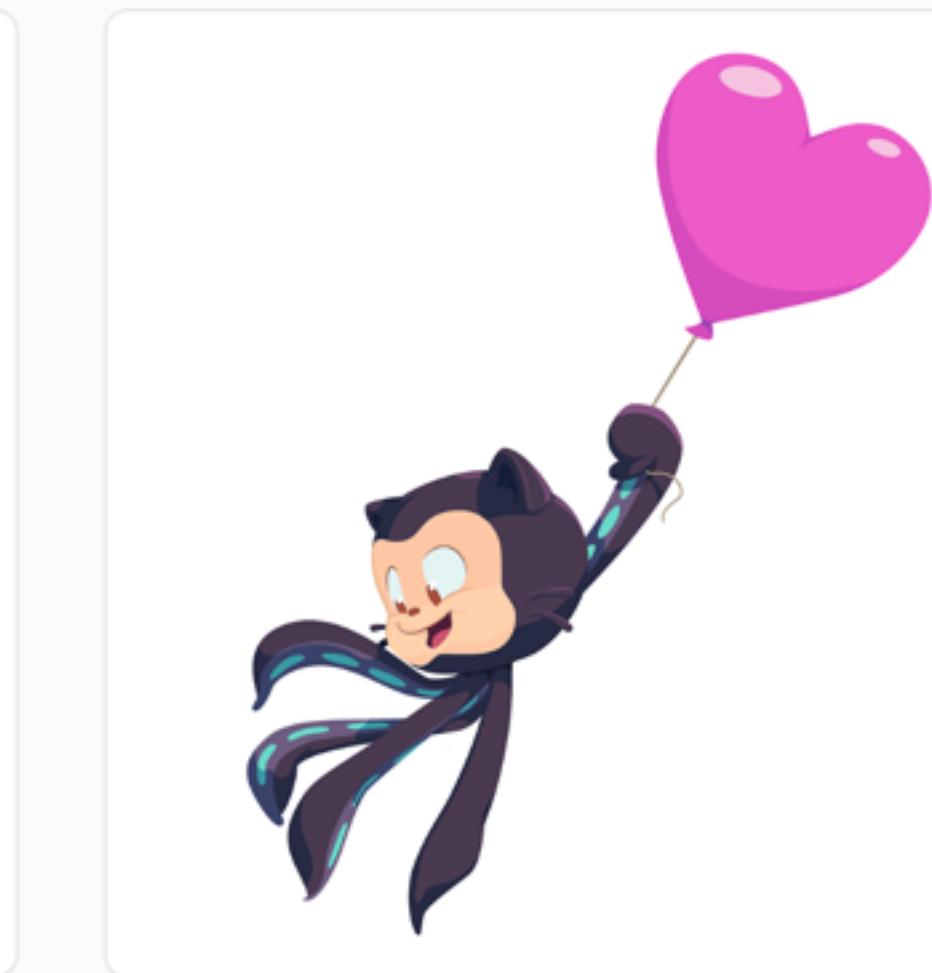
Legal speak

GITHUB®, the GITHUB® logo design, OCTOCAT® and the OCTOCAT® logo design are exclusive trademarks registered in the United States by GitHub, Inc. The OCTOCAT design is the exclusive property of GitHub, Inc and has been federally registered with the United States Copyright Office. All rights reserved. No adaptation or use of any kind of any of our registered trademarks or copyrights, or any other contents of this website, is allowed without the express written permission of GitHub, Inc.

For more information regarding the authorized uses of these items, please [contact GitHub Support](#).



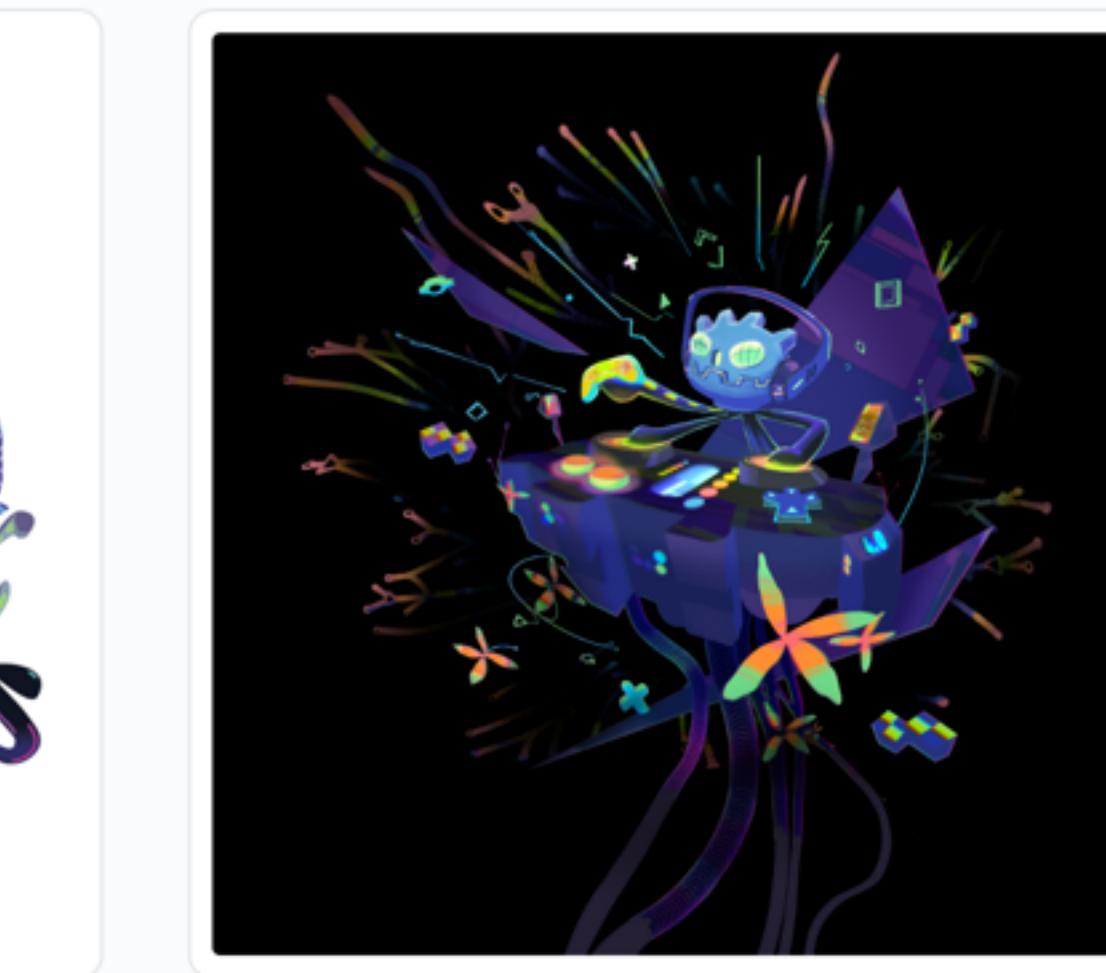
#157: Universetocat



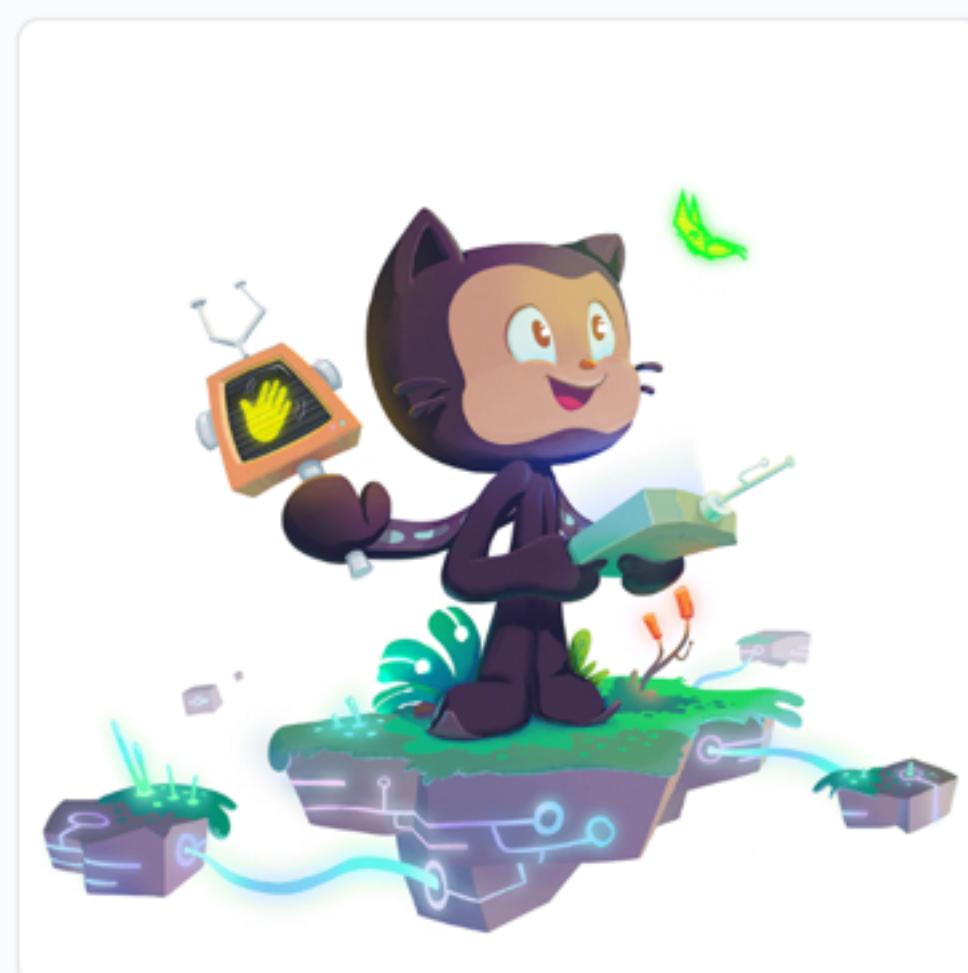
#156: Sponsortocat



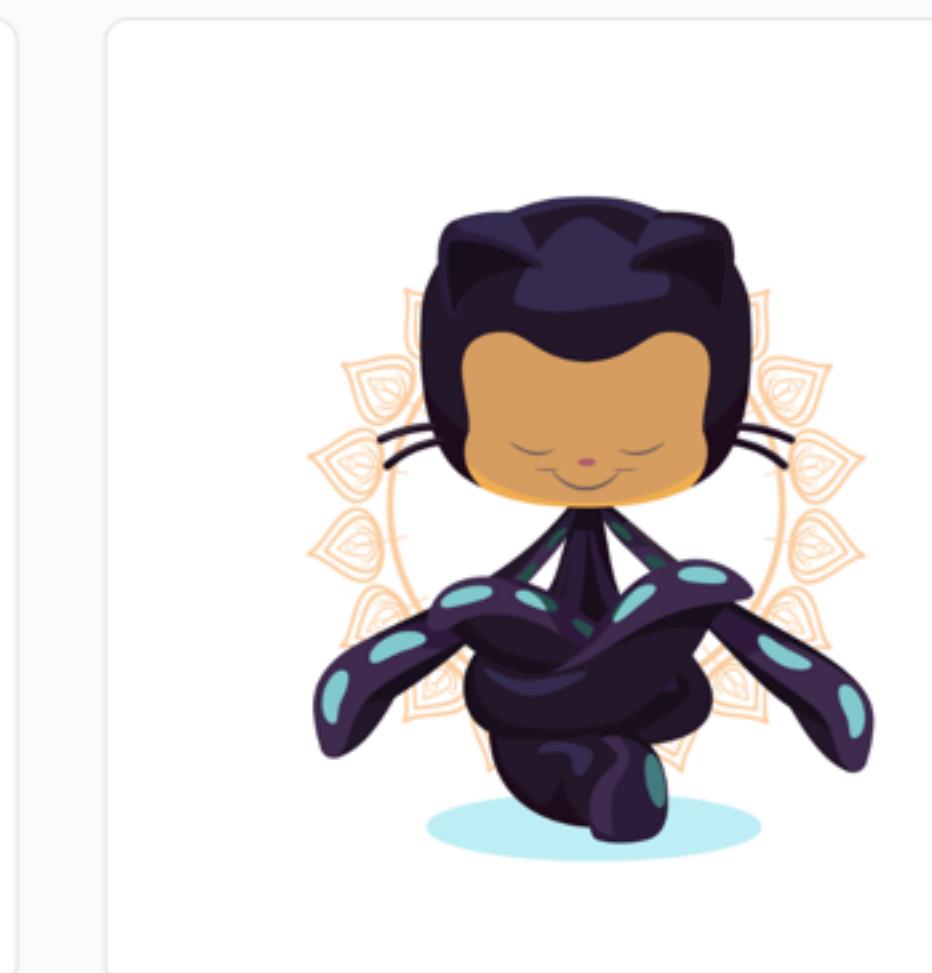
#155: Parentocats



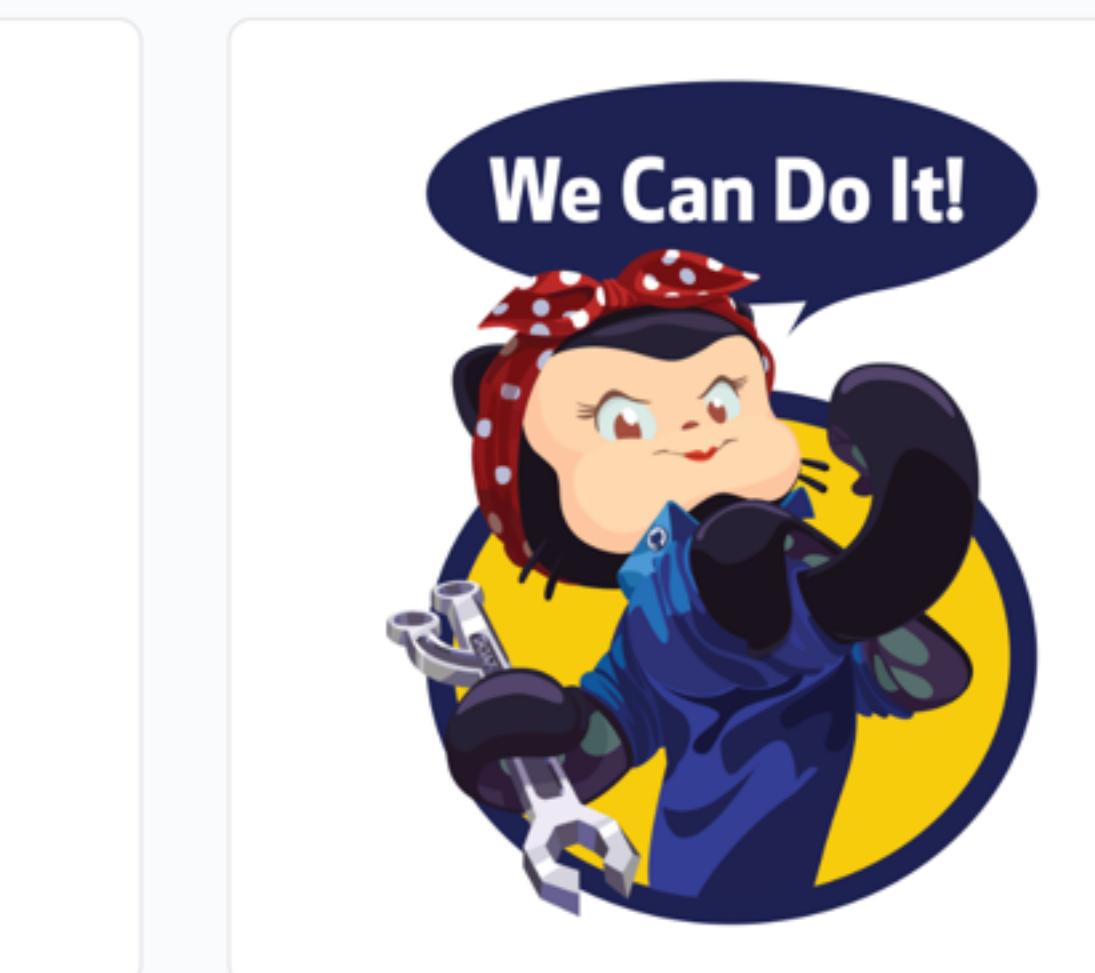
#154: Godotocat



#153: NUXtocat



#152: Yogitocat



#151: Mona the Rivertocat



#150: Manufactoretocat

Add a little plush to your life



shop.github.com

Signing up for GitHub

A guided example,

Some things that I might forget

- A GITHUB USERNAME IS PERMANENT. YOU CAN NOT CHANGE IT. THINK WISELY BEFORE MEMEING ON THE NAME
- Pick a username you will be comfortable revealing to your future boss.
- Shorter is better than longer.
- For the love of god please do not use uppercase. This will make it slightly harder and more annoying for you to clone and do stuff you/others need to type out

GitHub.com

Go here

Back to the hash reality of Git

Configuring Git CLI

Don't have it?

- Head to git-scm.com/downloads and download the file recommended for your system (this step is only necessary on windows and some linux distros)

The screenshot shows the 'Downloads' section of the git-scm.com website. At the top, there are links for macOS, Windows, and Linux/Unix. Below these, a large image of a Mac monitor displays the 'Latest source Release 2.45.0' with a 'Download for Mac' button. To the left, there's a sidebar with links for About, Documentation, Downloads (GUI Clients, Logos), and Community. A note about the 'Pro Git book' is also present.



How check?

- Type in **git version** into terminal to check what version you have.
- You should have version 2.39.3 onwards
- (Honestly versions shouldn't matter much any version is fine for today)

```
[base] saumil@Saumils-MBP ~ % git version  
git version 2.39.3 (Apple Git-146)
```

- The latest version is 2.47.0 (as of end November)

Version Control

Basics of Version Control

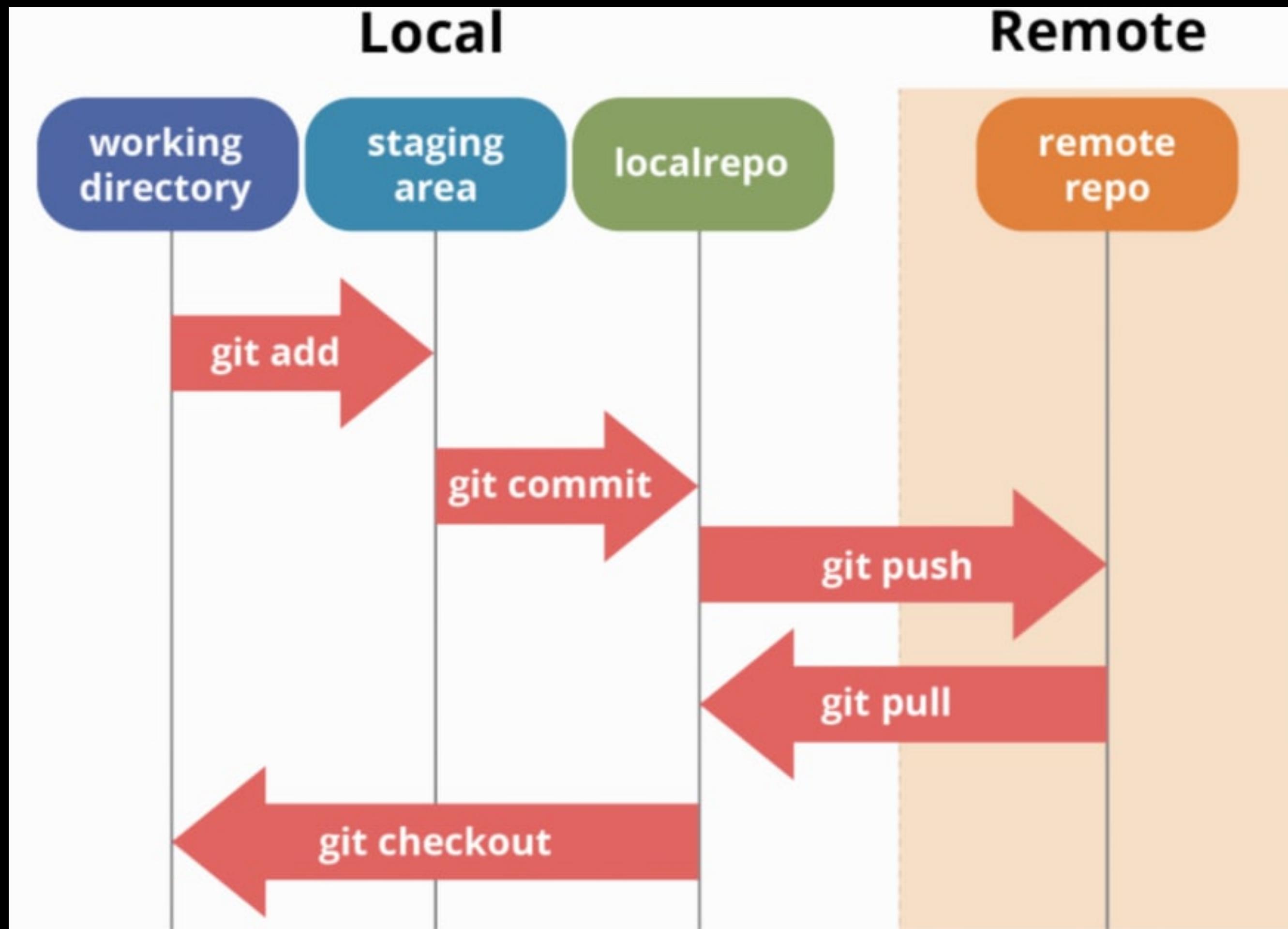
- Inherently intertwined with the core functionality of Git
- Keeps track of versions across different “timelines” or branches

“Oh no my code ain’t working no more”

“Why?”

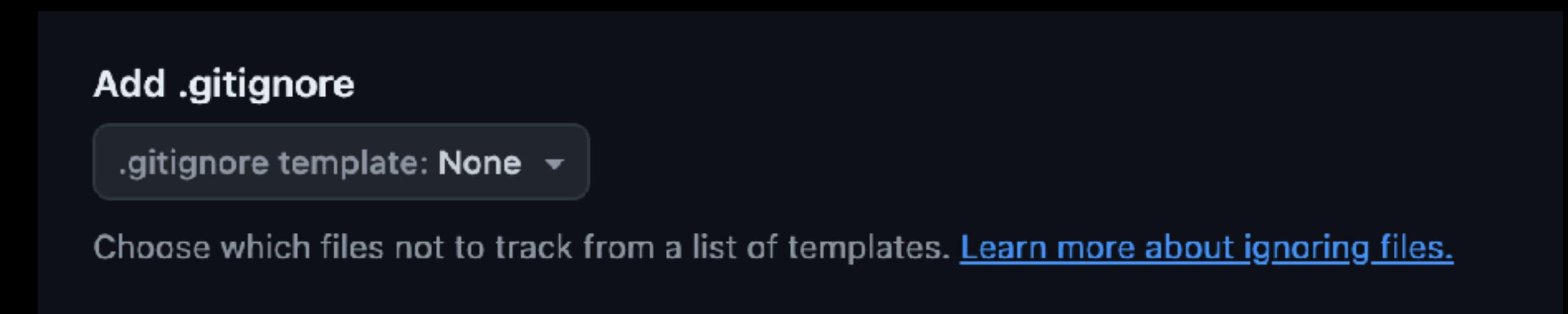
“No clue”

Using Git



.gitignore

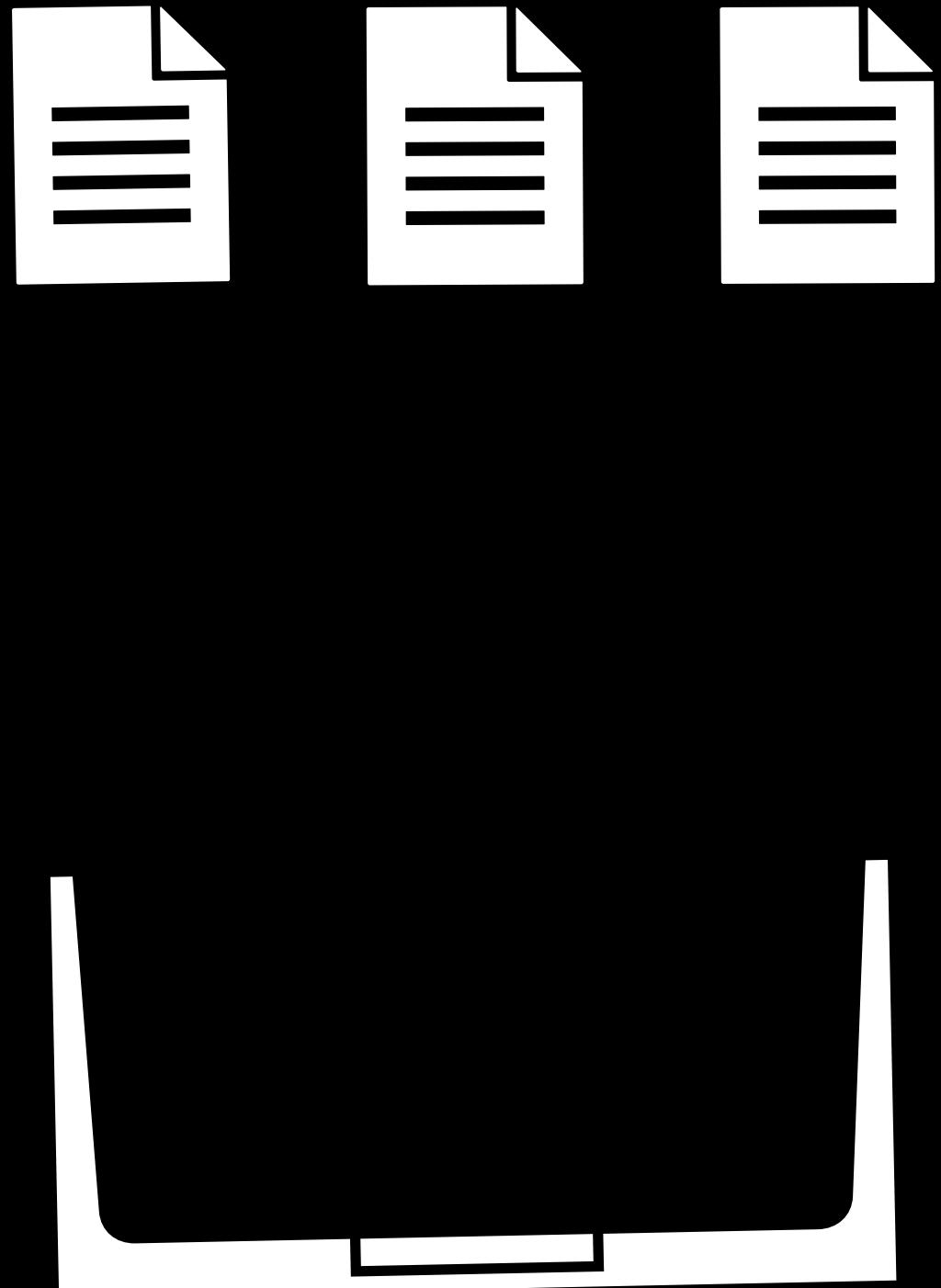
- Certain files you don't commit
 - Metadata, .env files, API keys etc.
- Different programming languages have different default files that should be commonly ignored



```
1  # Xcode
2  #
3  # gitignore contributors: remember to update Global/Xcode.gitignore, Objective-C.gitignore & Swift.gitignore
4
5  ## User settings
6  xcuserdata/
7
8  ## compatibility with Xcode 8 and earlier (ignoring not required starting Xcode 9)
9  *.xcscmblueprint
10 *.xcccheckout
11
```

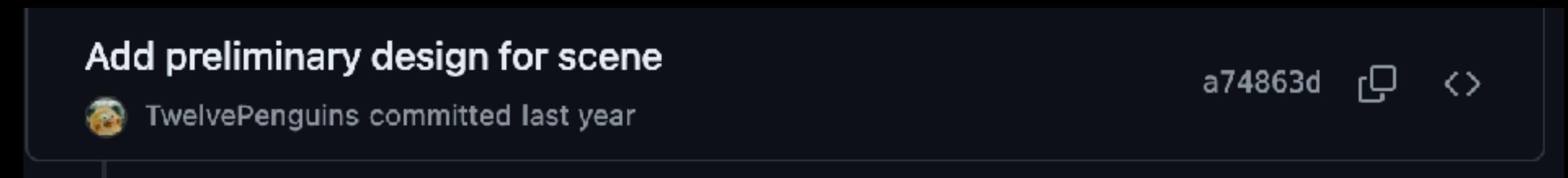
Git Add

- Multiple files in a single repository
- Stage files and add files to a commit with `git add`
- Syntax: `git add <path>`
- Flags:
 - `-f`: force add otherwise ignored files
 - `-A`: add all files
 - `-i`: interactively add files



Git Commit

- Requires specifically determined checkpoints/milestones
- Takes staged files and commits it to the local repository
- Each commit has a unique ID called a hash
- Requires a commit message
 - Should be concise and written in present tense
 - “Add ContentView”; “Update Home Page UI”

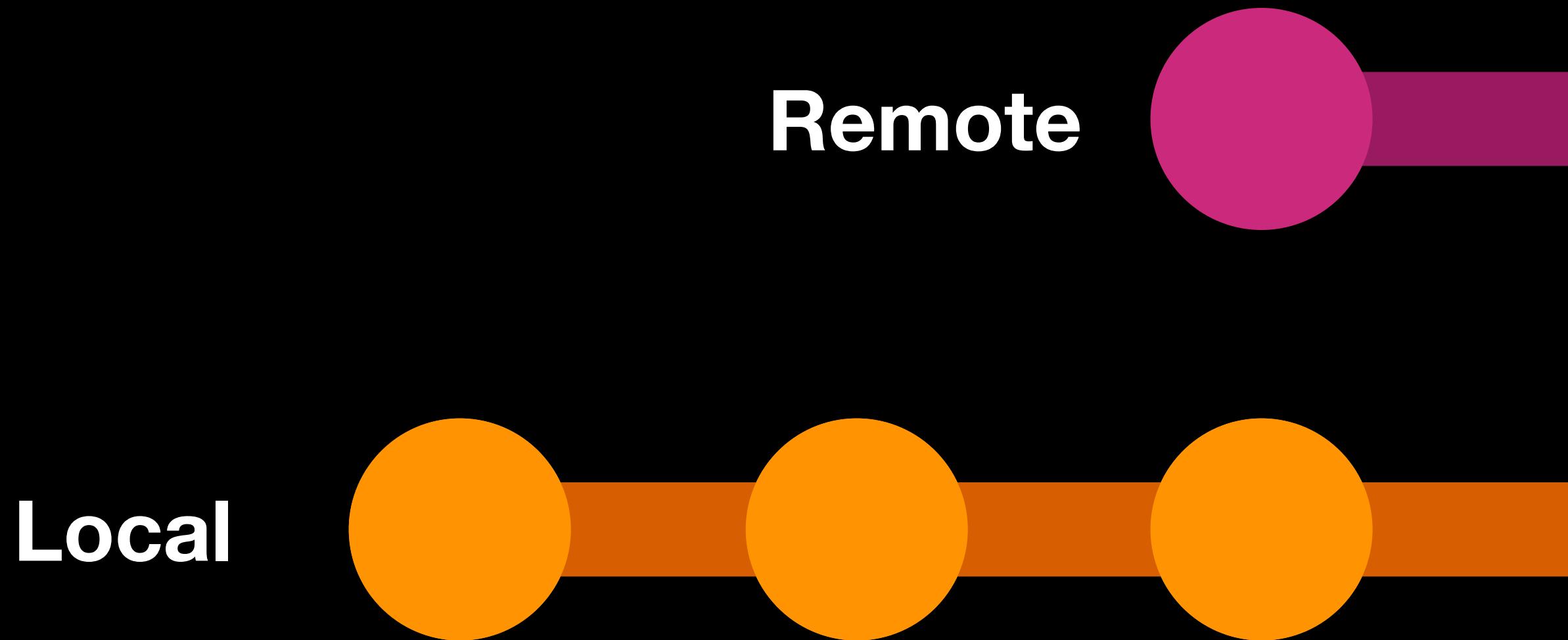


Git Commit

- Syntax: `git commit`
- Flags:
 - `-m <msg>`: Commit with the following message
 - `-a`: Stage all files that have not been staged and commit
 - `-c <commit>`: reuse authorship information of a commit that's passed in, while editing the log message
 - `--amend`: replaces the latest commit with the new commit
 - Recommended to only be used if commits have not been pushed to remote yet

Git Push

- Updates all new commits from the local repository to the remote repository
- Your friends can now access your code from another device!
- Syntax: `git push -u <remote name> <branch name>`
 - `git push -u origin main`
 - `git push`
- Flags:
 - `--all`: push all branches



Git Pull

- Updates all new commits from the remote repository to the local repository
- You accessing your friend's code from your device
- Combination of **git fetch** and **git merge**



Git Pull

- Important to check for uncommitted changes with `git status` before pulling
- Pull before push
- Syntax: `git pull`
- Flags:
 - `--rebase`: updates branch with remote commits first, then local commits

Git Checkout

- Very flexible command that allows you to switch branches or restore files
- Discard uncommitted changes in a file to last commit
- Syntax: `git checkout -- <bad filename>`
- Flags:
 - `-p`: Interactively select chunks of code to restore

hello.md

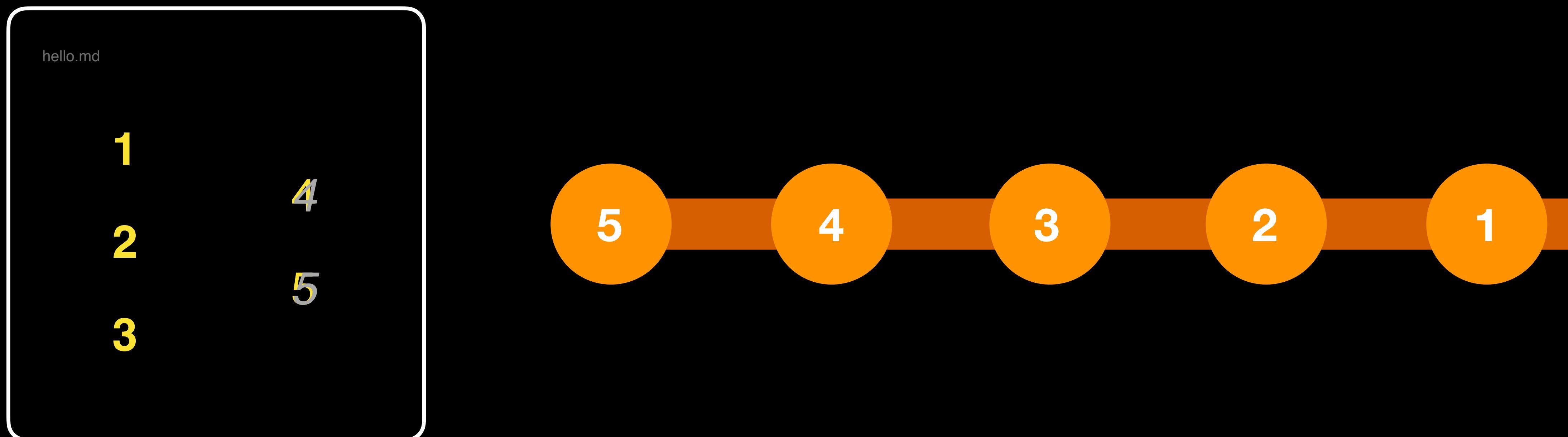
never gonna

give you up



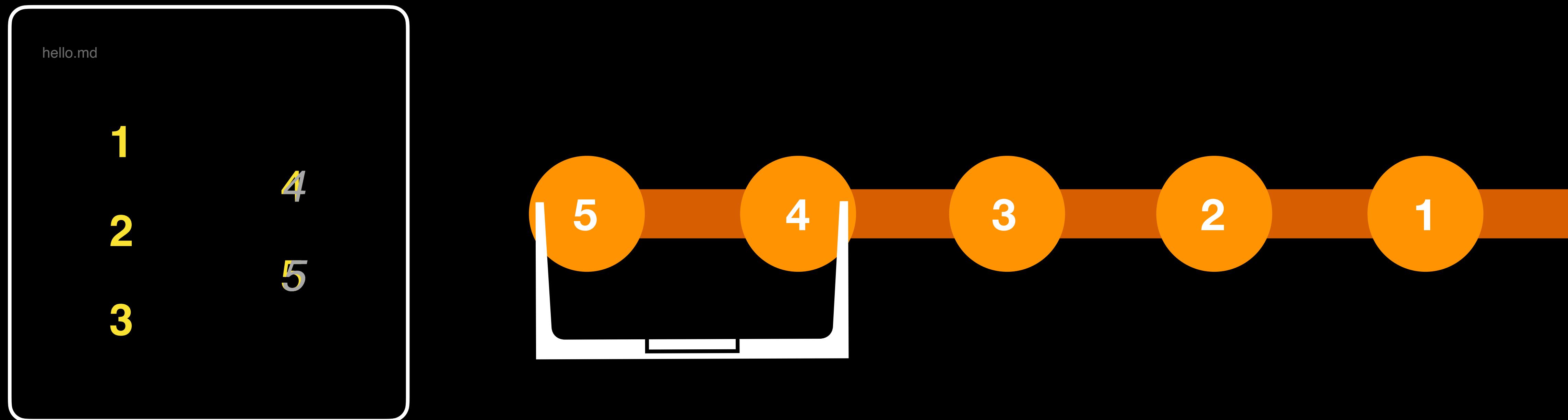
Git Reset

- Reset current branch head to a certain commit
- Preserves the progress of the commits on working direct by default
- Syntax: `git reset <mode> <target commit>`



Git Reset

- Modes:
 - **--hard**: deletes progress of reseted commits
 - **--soft**: preserves the progress of commits as staged changes



Git Reflog

- Reference log for all updates to the local branch for a certain period of time
- Combined with `git reset` to recover changes
- Syntax: `git reflog <command>`
- Commands:
 - `list`
 - `delete` - delete individual entries of the reflog

Git Status

- Displays current state of repository
- Includes information such as the current branch, staged files etc.

```
              p/sit-status (master)
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
```

Git Log

- Shows a list of commits

```
/Desktop/git (master)
$ git log --stat
commit 24d7a4c4cb142ac7a6aa4c2fb4b05e7545aa8fe3 (HEAD -> master)
Author: SoftwareTestingo Admin <softwaretestingo@email.com>
Date:  Tue Jul 26 17:15:31 2022 +0530

    Added Content

abc.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit e28c08ad9e4ea7c26fe9fbfeec89b1c073d347b8
Author: SoftwareTestingo Admin <softwaretestingo@email.com>
Date:  Tue Jul 26 17:15:04 2022 +0530

    Added Content

abc.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 9c2bd2af982f5cf837da69950c23859b41b63303
Author: SoftwareTestingo Admin <softwaretestingo@email.com>
Date:  Tue Jul 26 17:14:34 2022 +0530

    Added Content

abc.txt | 1 +
1 file changed, 1 insertion(+)

commit 10eaa0c918d7ee2b6e21a0fc0c7dcbc578cc37f4
Author: SoftwareTestingo Admin <softwaretestingo@email.com>
Date:  Tue Jul 26 15:19:48 2022 +0530

    file added

abc.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)

/Desktop/git (master)
$
```

Setting up a repository

- Cloning a remote repository that you've created on GitHub
 - `git clone <remote url>`
- Making a directory a git repository
 - `git init <directory>`
- Adding a remote to an existing local git repository
 - `git remote add origin <remote url>`

Hands on with Git

Set Up

- Apply what you've learnt in the past few minutes

- Save your octocat it to your downloads folder
 - Yuhan recommends octopenguins
- Rename the picture your name + surname

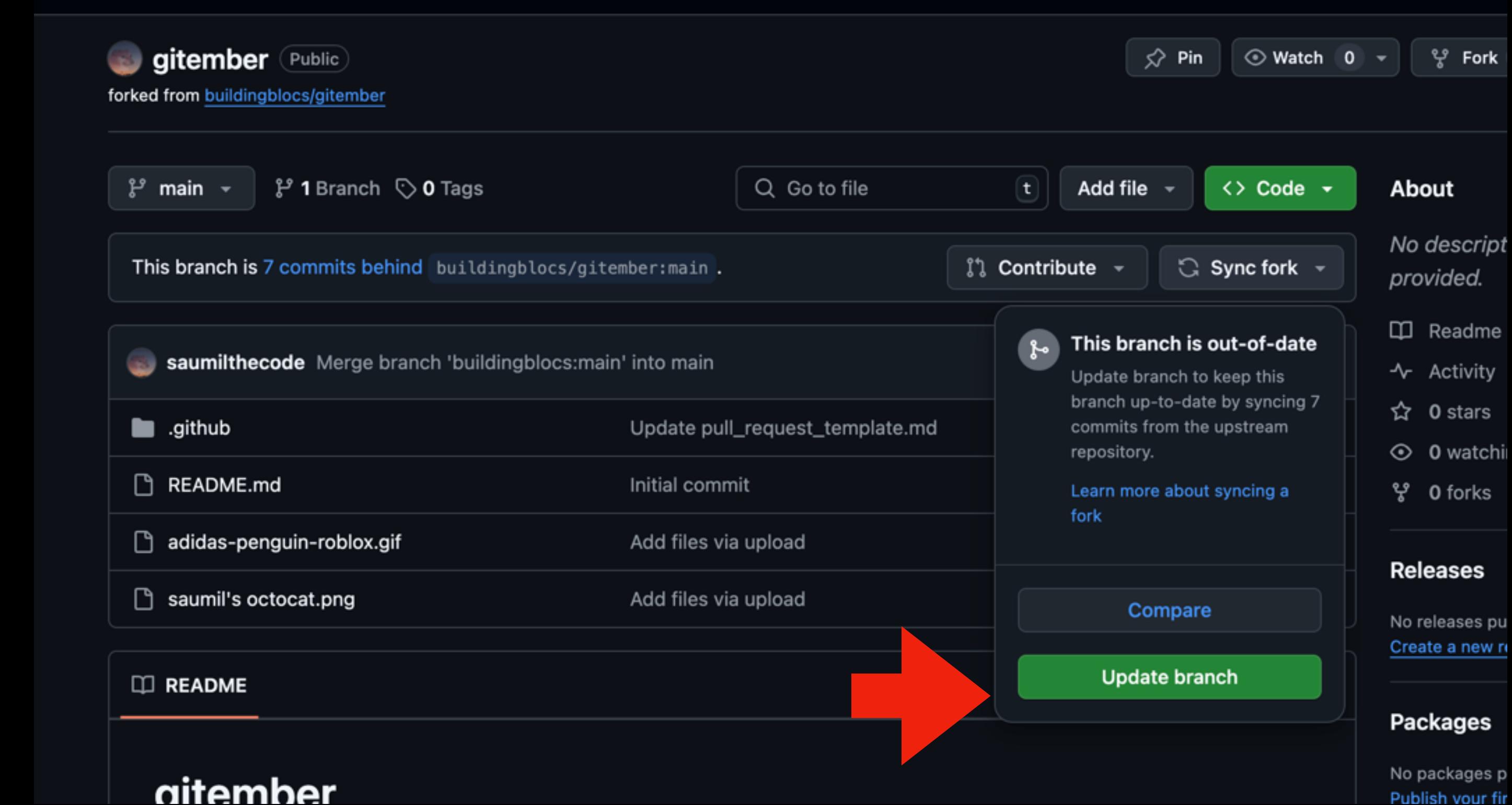
<https://github.com/buildingblocks/gitember>

Main instructions

- Fork on GitHub, - github.com/buildingblocks/gitember
- In terminal - git clone <your fork url>
- Then `cd gitember`
- Drag your file from downloads into gitember
- RENAME FILE NO SPACES (linus had broke the repo)
- `git add .` (this adds all the files in the directory IE:yourimage)
- `git commit -m "my penguin"
- `git push -u origin main`

If broken,

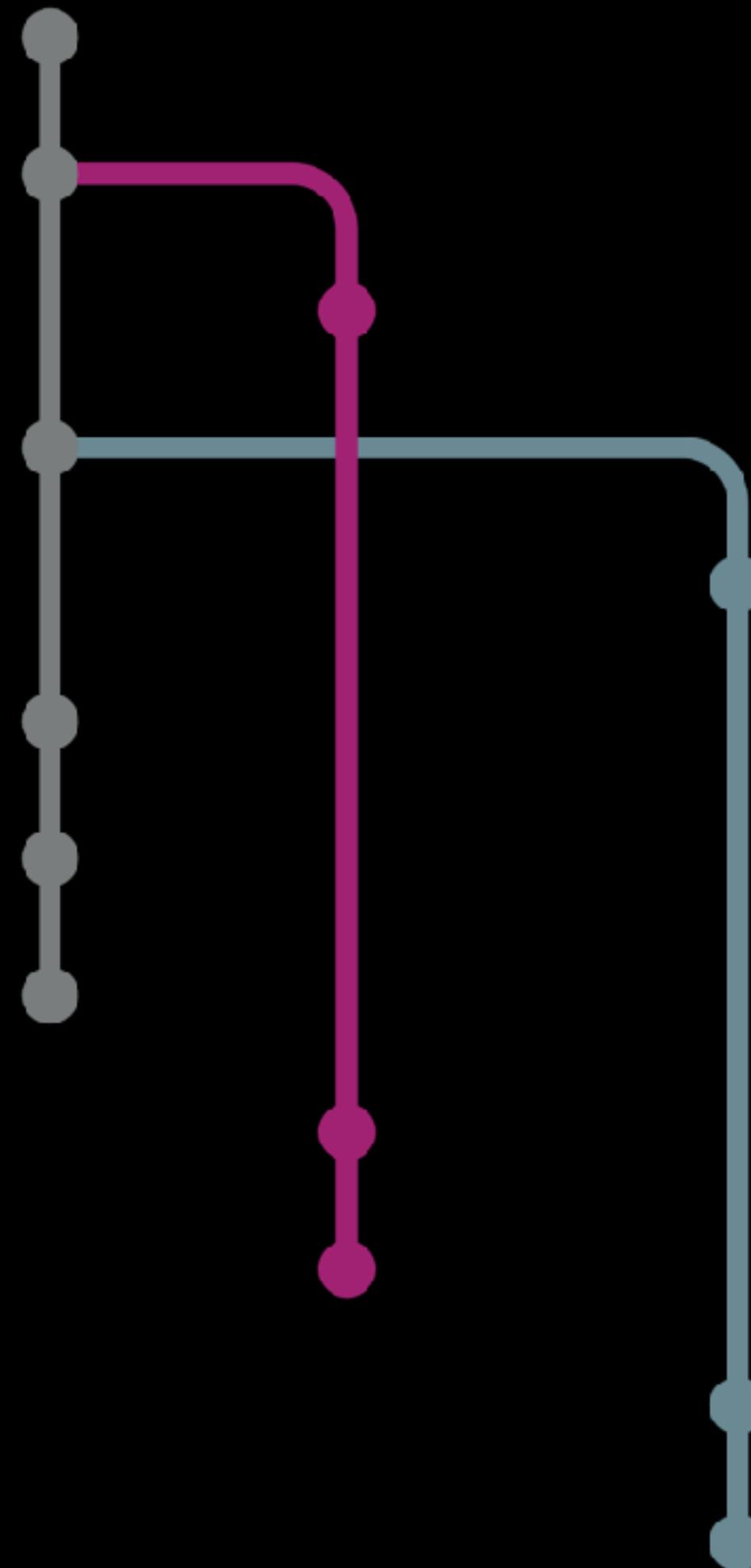
- Go to your fork on [GitHub.com](#)
- SYNC FORK
- In terminal cd to folder and git pull
- Should work



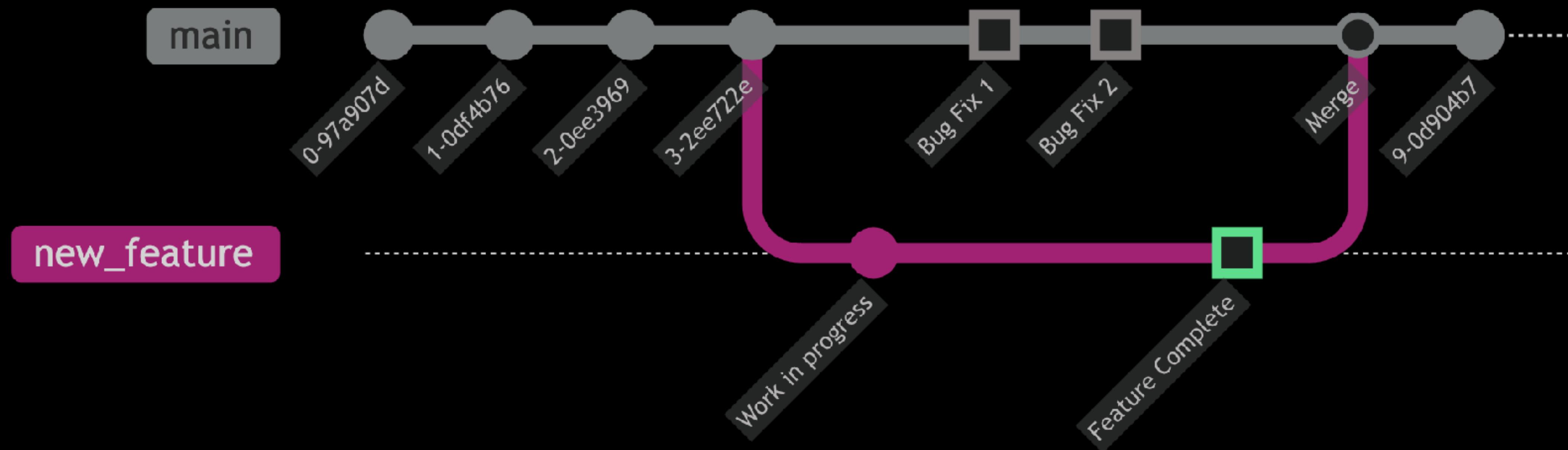
Branches

What are branches?

- Isolated environment separate from the main codebase
- Ensures stability of the main branch during the development of new features and bug fixing
- Easier tracking of development

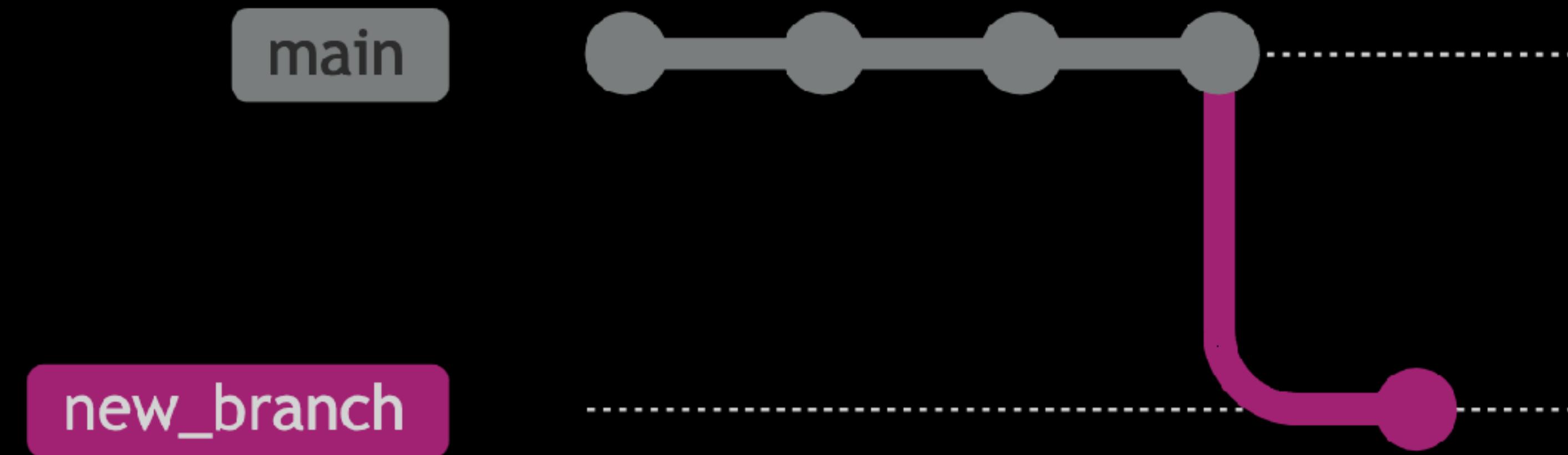


Branches



Working with branches

- Create branches with `git branch`
- Switch branches with `git checkout`



Git Branch

- Create a new branch with `git branch <name_of_new_branch>`
- Syntax: `git branch <name_of_branch>`
- Flags:
 - `-d`: delete branch
 - `-l`: lists all branches
 - `--show-current`: shows current branch
 - `-m <old_branch> <new_branch>`: move/rename a branch

Git Checkout (for switching branches)

- Switch branches (or restore files)
- Syntax: `git checkout <name_of_branch>`
- Flags:
 - `-b <name>`: create new branch and switch

Check which branch you're on

- Use `git branch --show-current` command
- Check from `git status`
- Check from `git log`
 - `commit <commit hash> (HEAD -> master, new_branch)`



Current branch

Merging branches

- Merge branches with `git merge`
- Combine changes from another branch to current branch



Git Merge

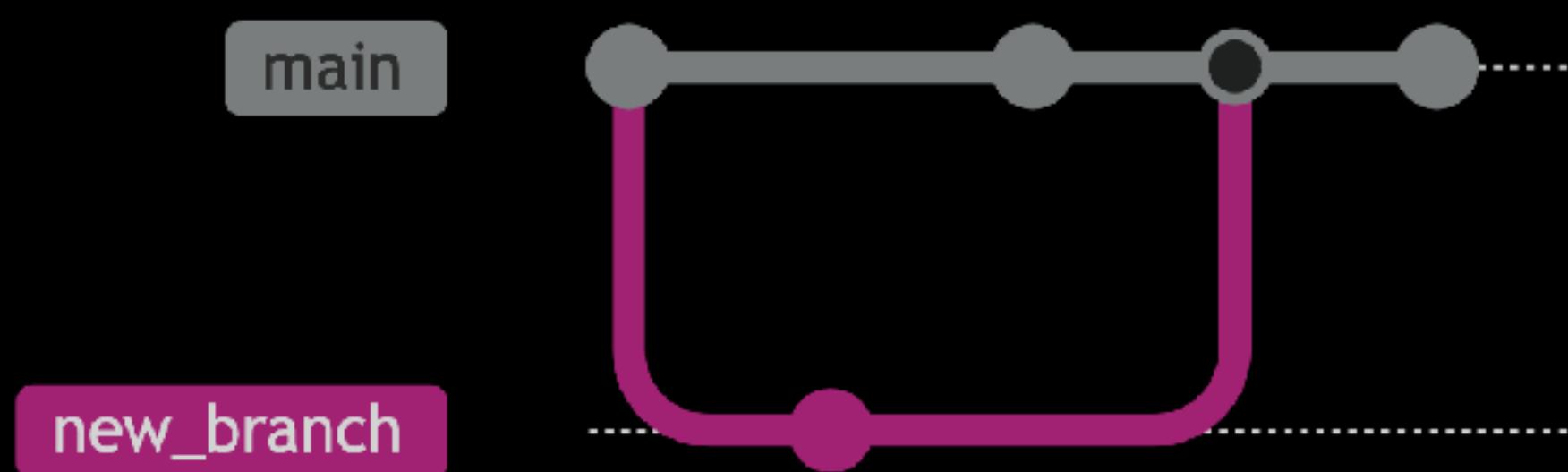
- Merge a branch into current branch by using `git merge <name_of_branch>`
- Syntax: `git merge`
- Flags:
 - `--abort`: abort after encountering a conflict
 - `--continue`: continue after resolving a conflict

Merging branches

- Ensure that you are on the right branch before using `git merge`

`git checkout main`

`git merge new_branch`



`git checkout new_branch`

`git merge main`

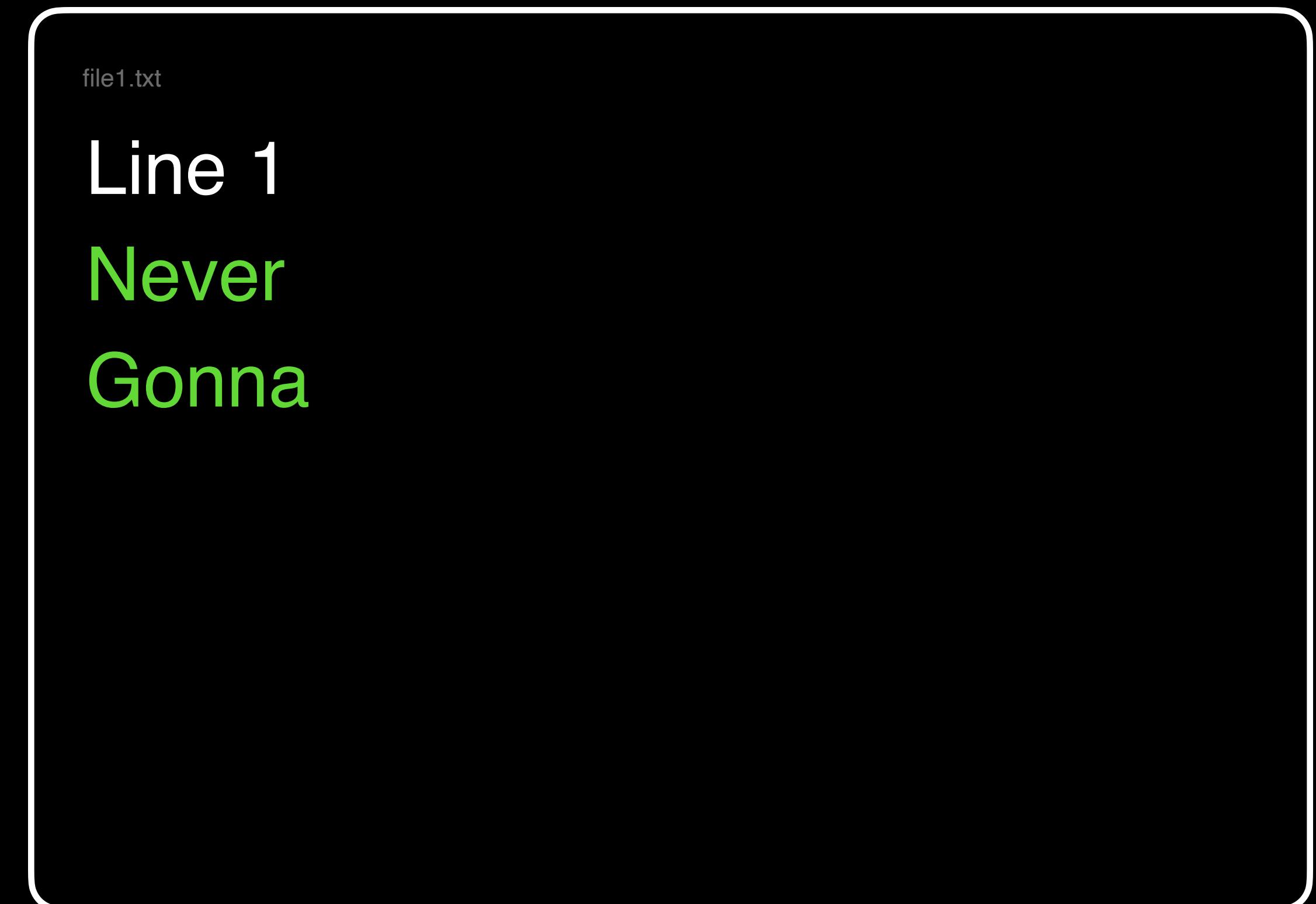
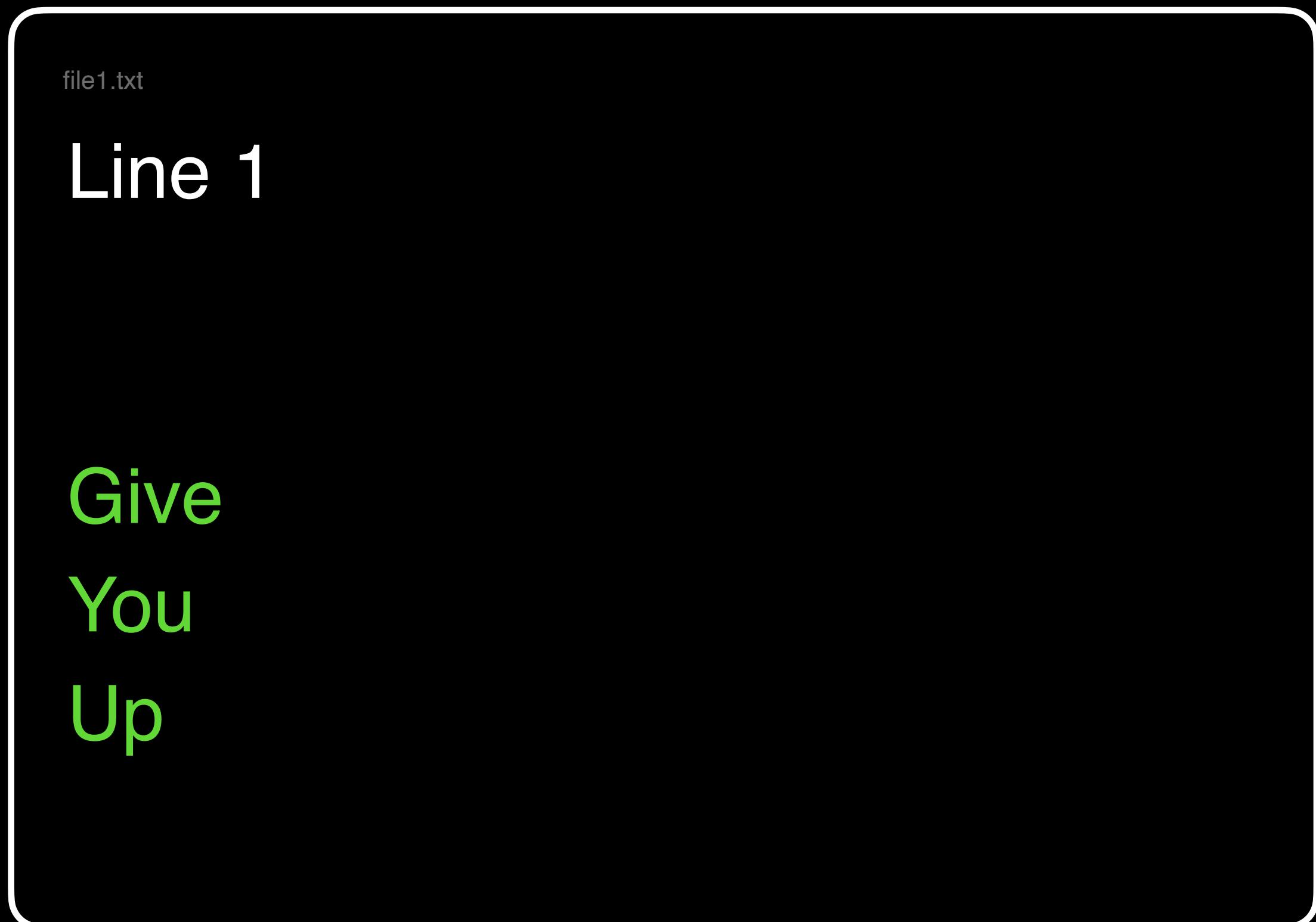


When you are gonna
merge two branches
after long time..



Example of how code will merge

- Merging file1.txt from two branches

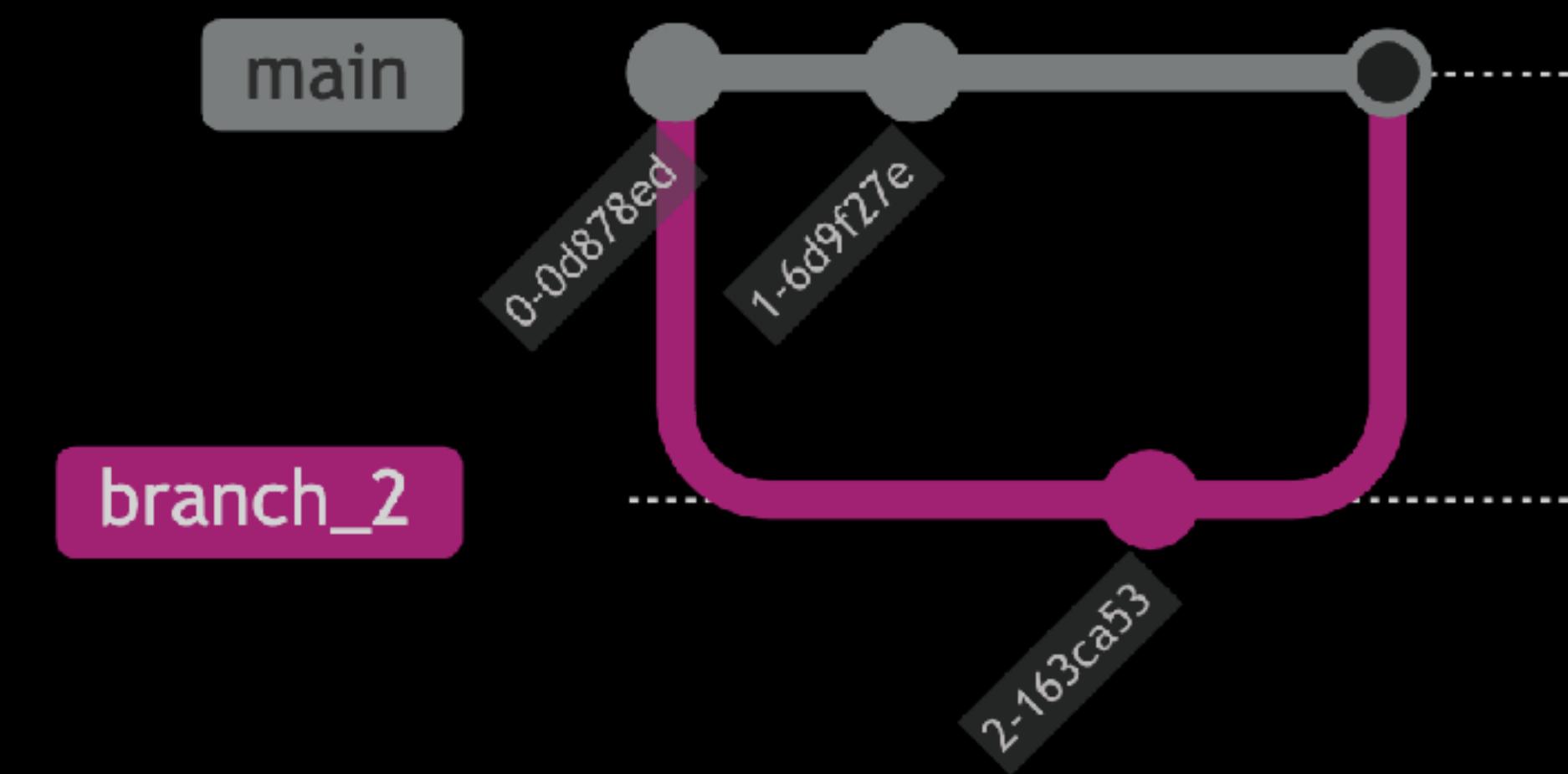


Example of how code will merge

- Merge result

file1.txt

```
Line 1
Never
Gonna
Give
You
Up
```



Merge conflicts

- Merge will "fail" when there is a conflict
- Happens when two different branches both contain changes in the same line of the same file
- Conflicts are resolved manually before it is successfully merged

CONFLICT (content): Merge conflict in file.txt

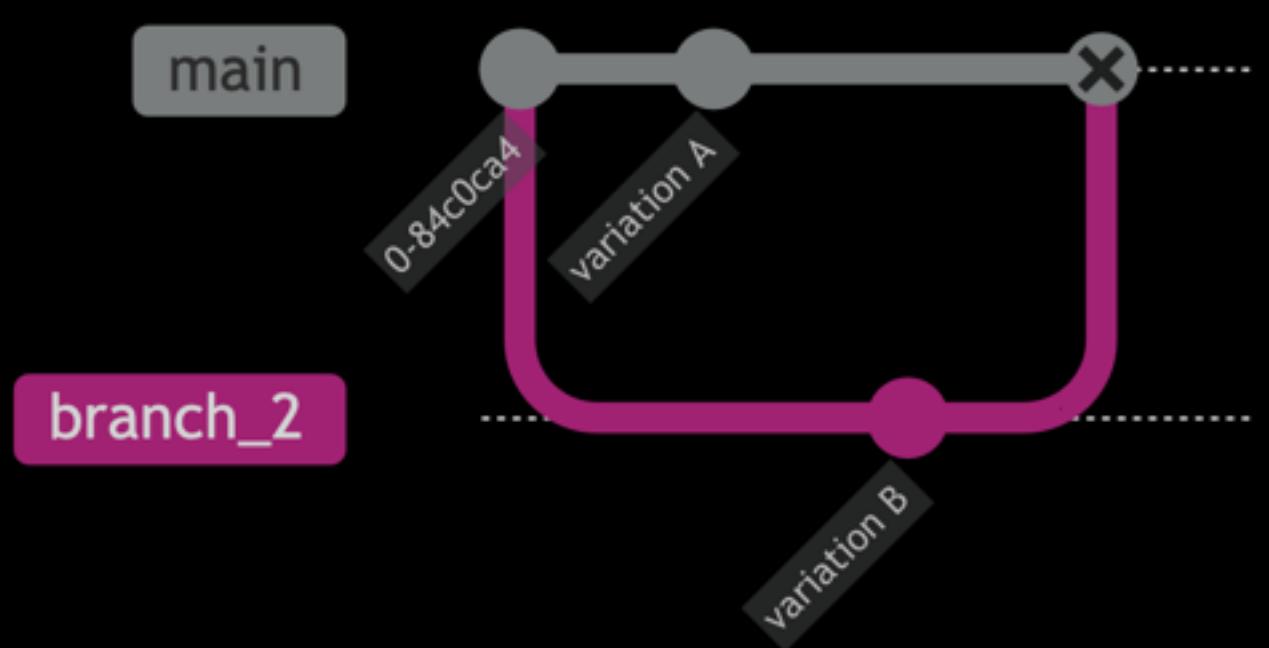
Automatic merge failed; fix conflicts and then commit the result.

Merge conflicts

- Example of conflicting files

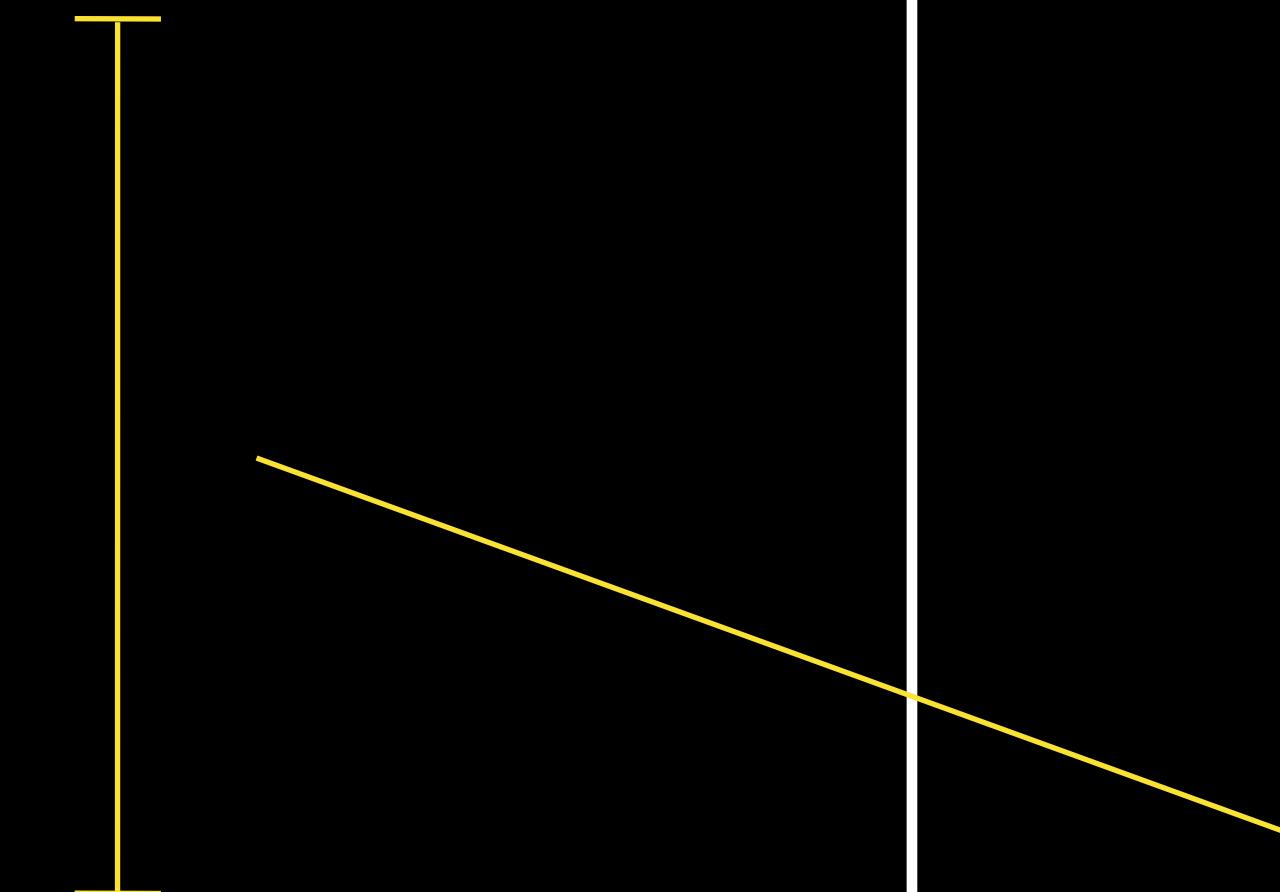
```
file1.txt  
Line 1 variation A  
  
Give  
You  
Up
```

```
file1.txt  
Line 1 variation B  
Never  
Gonna
```



Merge conflicts

```
file1.txt  
=><<<<< HEAD  
Line 1 variation A  
=====  
Line 1 variation B  
>>>>> branch-name  
Never  
Gonna  
Give  
You  
Up
```



```
<<<<< HEAD  
Line 1 variation A  
=====  
Line 1 variation B  
>>>>> branch-name
```

Resolving conflicts

- Open file with conflicts using any text editor
- Conflicted lines will be shown in the syntax below

file1.txt

```
<<<<< HEAD
Line 1 variation A
=====
Line 1 variation B
>>>>> branch-name
```

Beginning of conflict

Separator between the conflicting changes

End of conflict with name of branch being merged

```
JS example.js > ...
1 function processData(data) : void {
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2 <<<<< HEAD (Current Change)
```

Resolving conflicts

- Remove the unwanted lines and keep correct lines and save the file
- Stage the file again with `git add`, then proceed with `git merge --continue`
 - Or just commit again normally using `git commit -m <message>`

file1.txt

```
<<<<< HEAD
Line 1 variation A
=====
Line 1 variation B
>>>>> branch-name
```

file1.txt

```
Line 1 variation A
```

Hands-on git branching

Branching hands-on setup

- Apply what you've learnt in the past few minutes

- Create a new Github repository
- Find an ASCII art of an animal
 - <https://www.asciiart.eu/animals/>
- Paste it into a new plain text file in your repository
- Commit and push changes

Your task

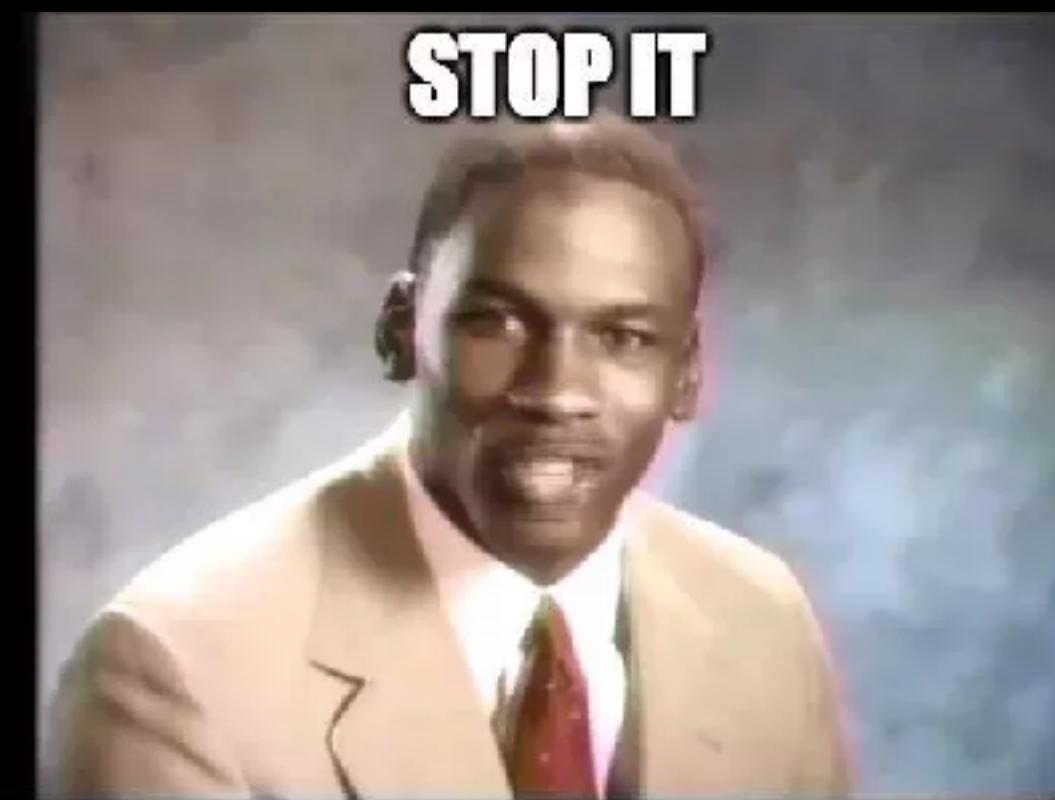
- Apply what you've learnt in the past few minutes

- Create a new branch
- In the new branch, change the eyes of your ascii art animal into different characters
- Commit and push
- In the main branch, change the eyes of your ascii art animal, different from the new branch
- Commit and push
- Try to merge new branch into main branch

Recap

Next steps

- If you want to learn more about Git, you can try out this fun game called <https://learngitbranching.js.org>
- GitHub's Git Cheatsheet (<https://training.github.com/>)
- If you wish to use a GUI version of git you can read up on `git-gui` and `gitk`
- or theres always the good old
when you get lost



See 'git --help'.

Thank you!

Feed us the back

