

# Intro to Git

Jon Ambler  
Bioinformatician at CIDRI-Africa  
Based at the University of Cape Town  
South Africa

# Overview

- Goals:
  - To know what is possible
  - Have a feel for how it works, and what to use it for
  - Don't worry about memorising commands
- Topics
  - What is git?
  - How to use it
  - How to deal with when things go wrong
  - How to use it in teams
  - How it fits in to the greater picture

# What is Git?

- A version control system (VCS)
  - Git, Subversion, Perforce
- What is a VCS?
  - Allows you to keep track of changes to a file / files
  - Allows for non-linear work
- Used for:
  - Working collaboratively on the same project
  - Keeping work safe
  - Keeping a history of changes (Versioning)
  - Code, or writing! (Works great with LaTeX)

# What is Git?

- Git stores streams of snapshots
- Other VCS systems store changes
- Git stores snapshots, with unchanged files represented as links to the last unchanged version of the file.

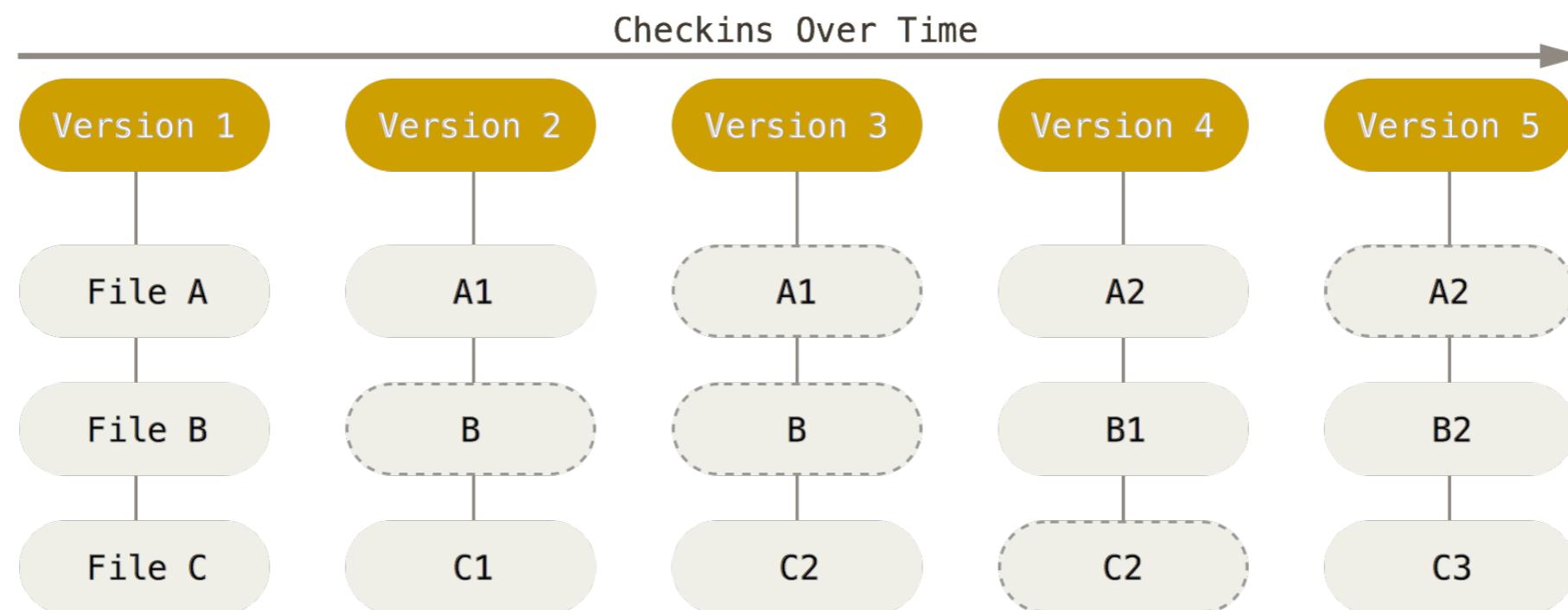
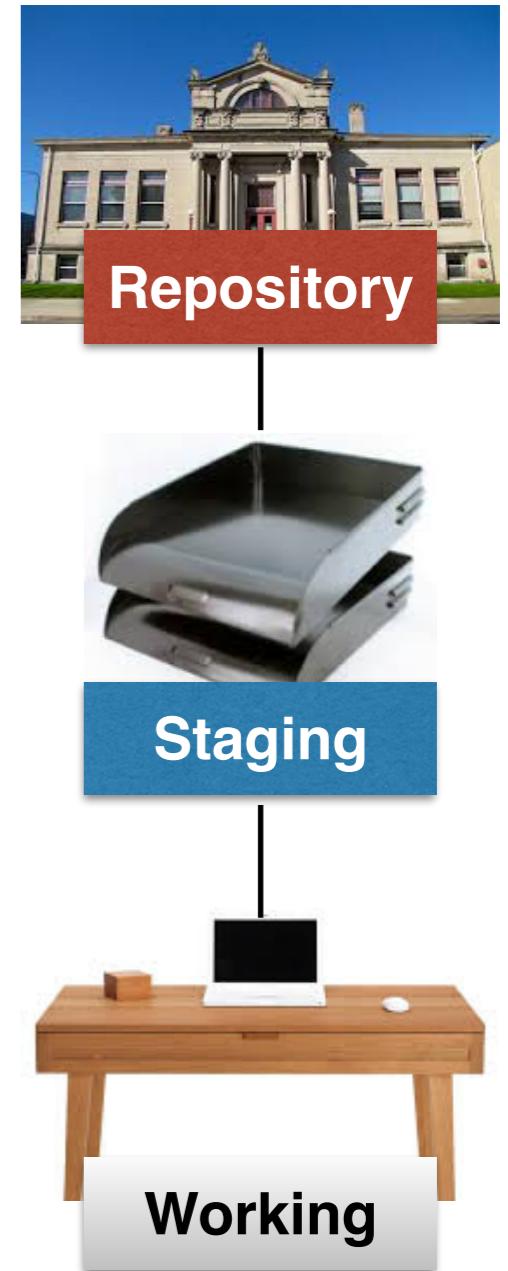


Image source: <https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F>

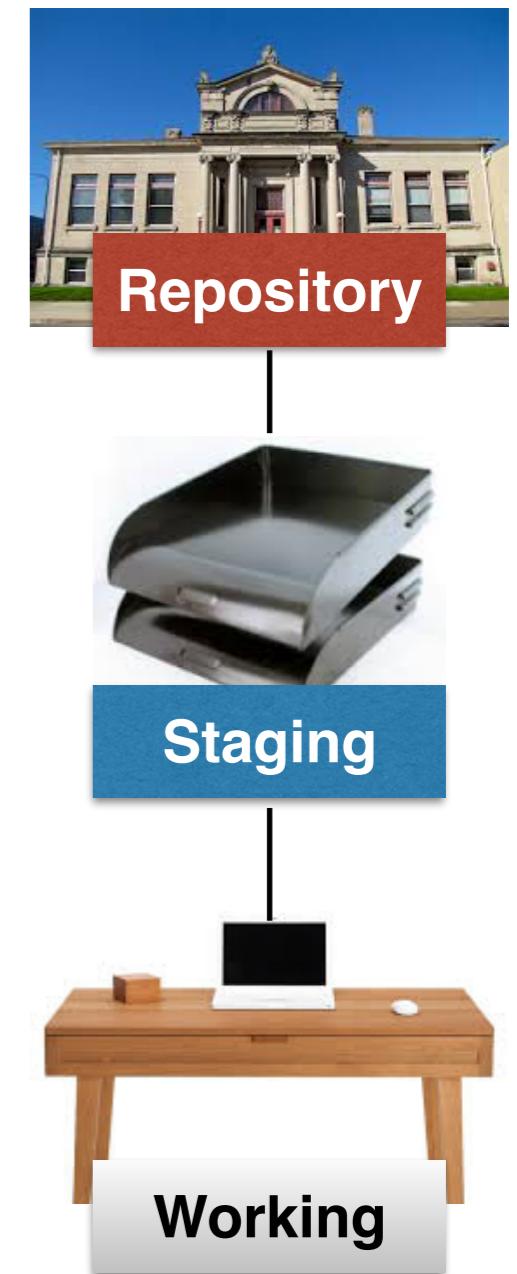
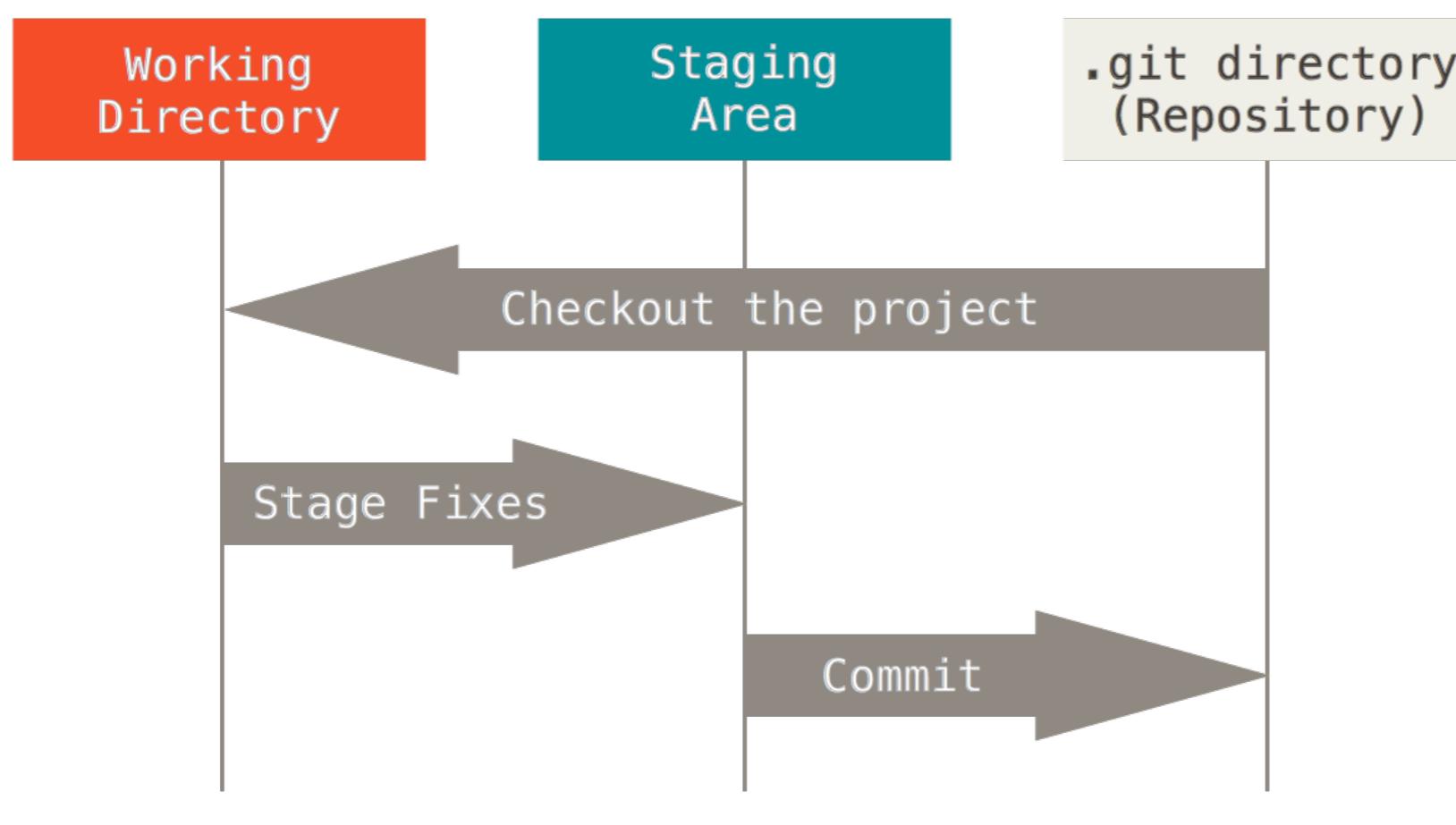
# Git usage

- Can be used from:
  - The command line
  - A browser
  - A GUI
  - Built into interactive development environments (IDEs)



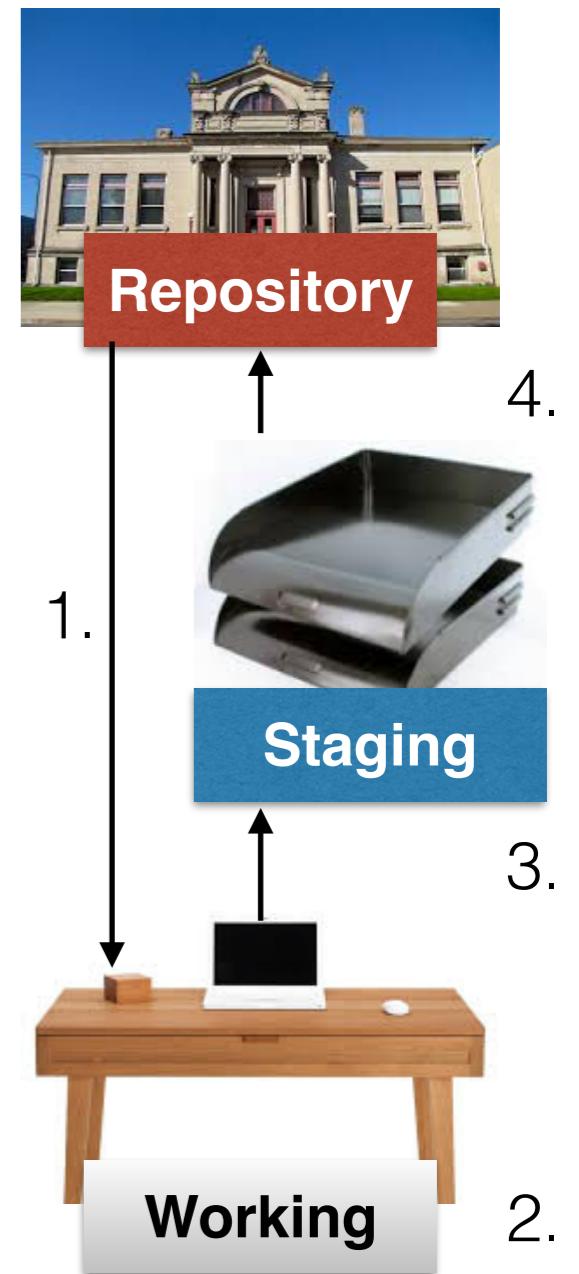
# Three states in Git

- A file can be committed, modified, or staged



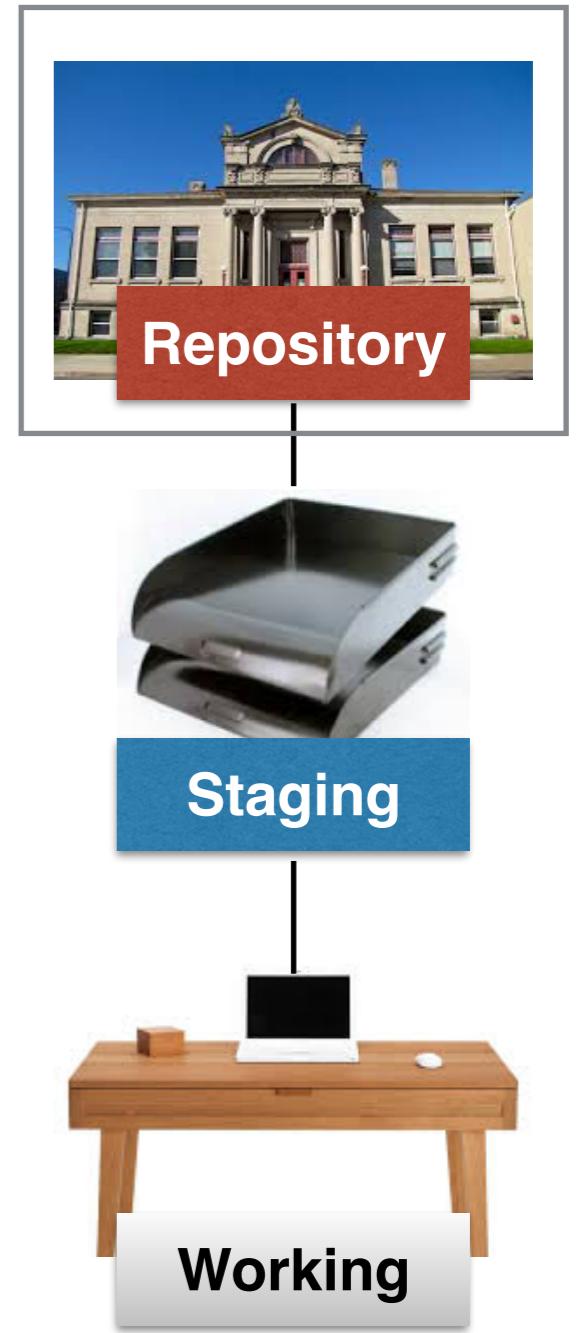
# Git workflow

1. Pull latest version from the repository
2. Modify the files in your working directory
3. Stage the files you have modified
4. Commit the files in your staging area to your Git repository



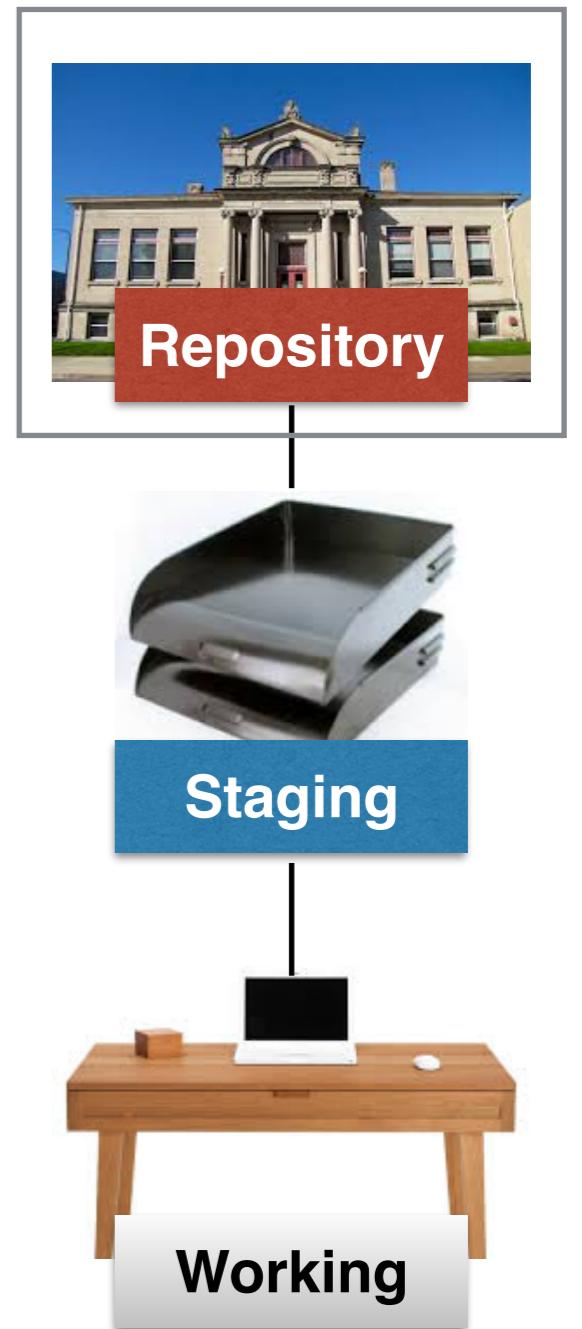
# Repositories

- Local
  - Where you are working
  - For sensitive data
- Remote
  - Github, Bitbucket, GitLab
  - Safer backup-wise
  - Needed for collaboration
  - Files are not private (Free version)
- Github
  - Git repository hosting service
  - Now owned by Microsoft



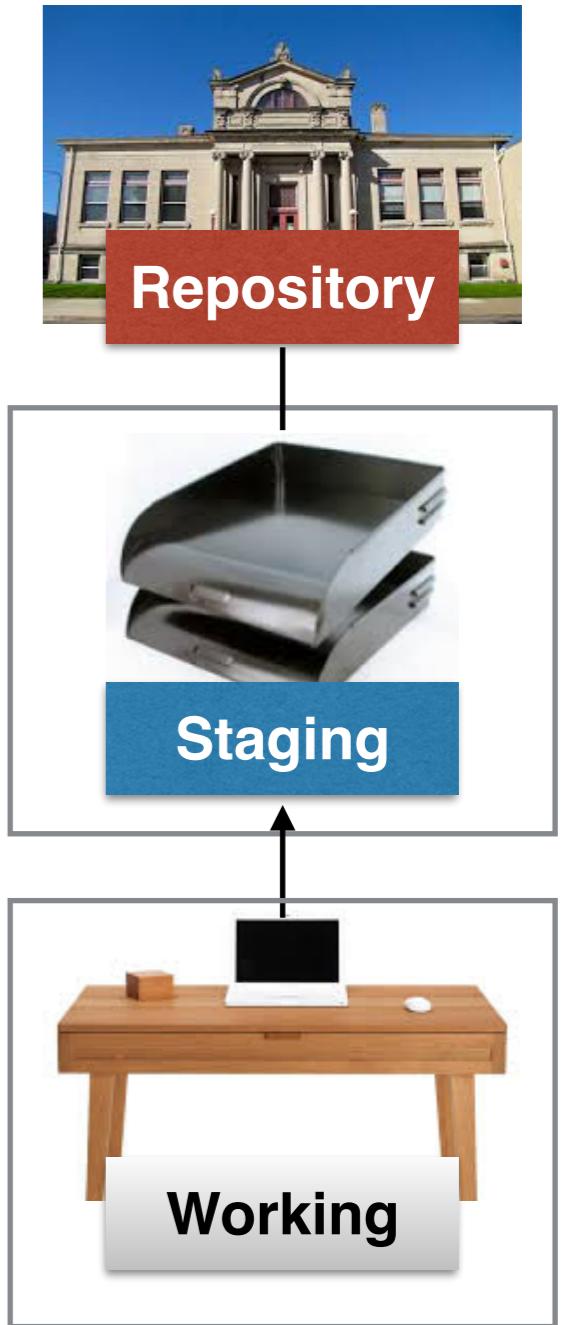
# Git usage

- Creating a local repo:
  - \$ git init
- Creates a .git folder
- Not visible except though ls -al



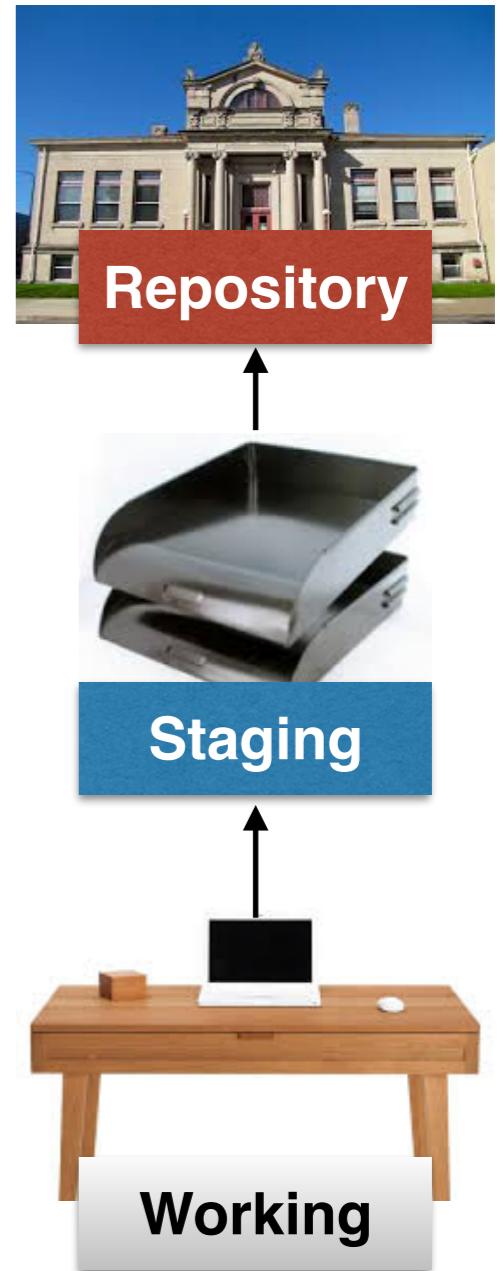
# Git usage

- Add files to be tracked:
  - \$ git add <file>
- Checking what is the state of the files in the dir
  - \$ git status



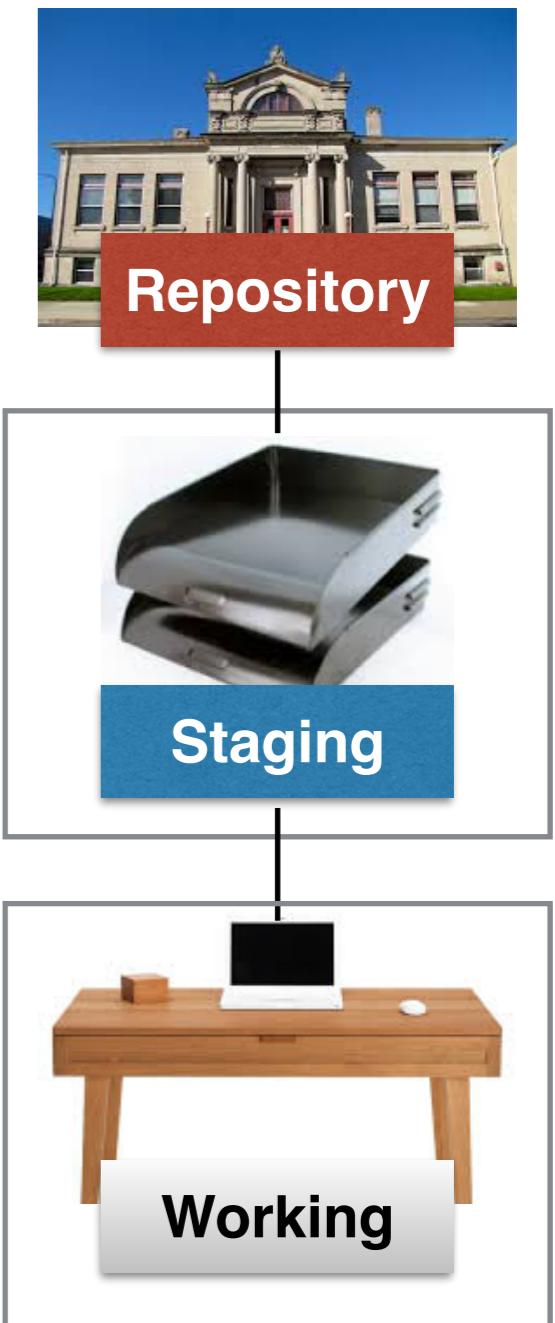
# Git usage

- To add a file to staging from the working area
  - \$ git add <file>
- To move a file from staging to the local repository
  - \$ git commit -m "A descriptive message"



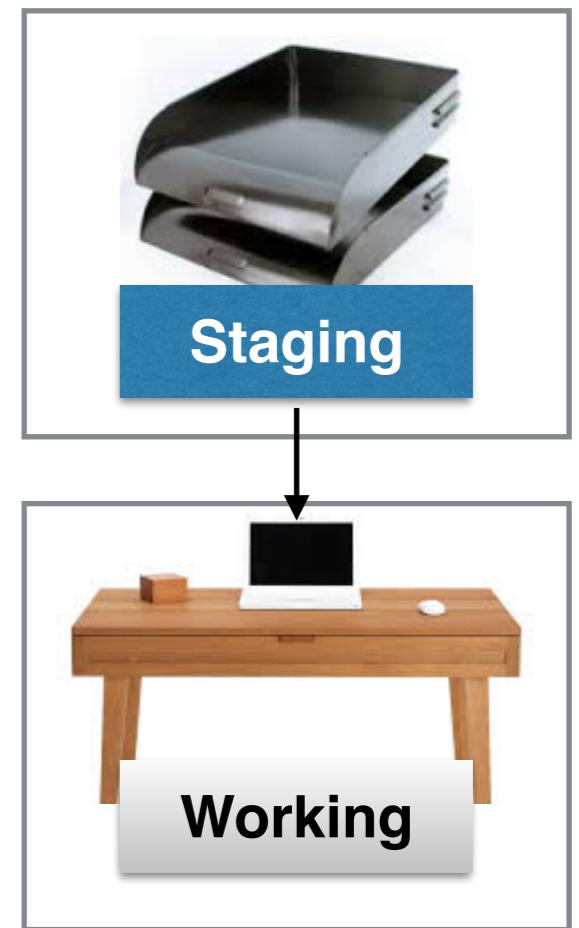
# Git usage

- If you start working on a staged file:
  - You will be working on a different file to the one that is staged
  - If you run git commit, the staged file, not the one you are working on will be committed to the repo

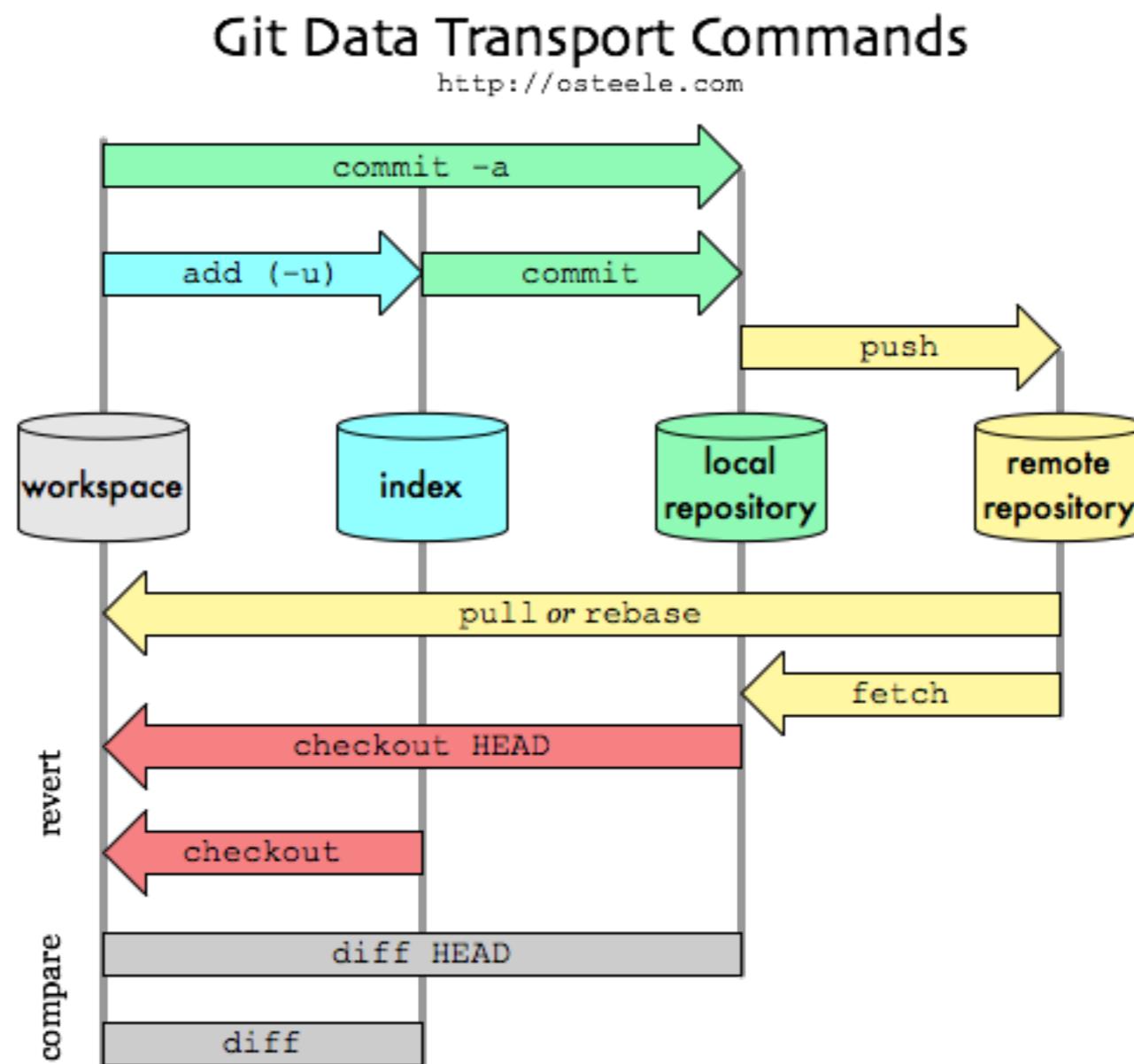


# Git usage

- The `.gitignore` file
  - Tells git to ignore files / folders / filetypes
  - They will not be shown when you type
    - `git status`



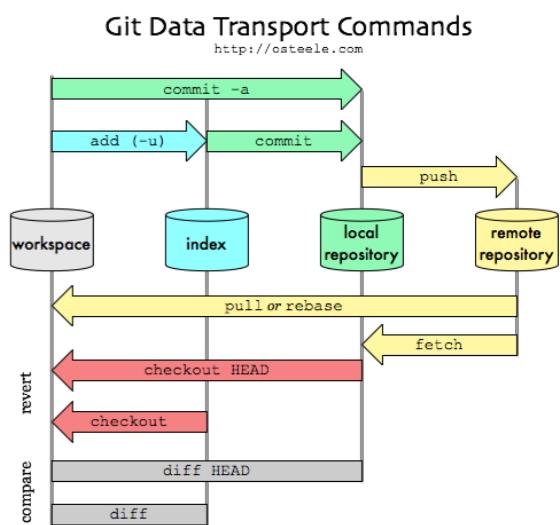
# Remote repositories



# Remote repositories

- Create a Github account
- To work on an existing repository:
  - `git clone <git url to clone>`

A screenshot of a GitHub repository page. At the top, there are navigation links: Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings. Below this, the repository summary shows 4 branches and 0 tags. A merge pull request from 'jambler24/testing' is listed. The main area displays commit history with changes like 'Added gitignore', 'Fixed readme', and 'Added lower case function'. A 'Code' dropdown menu is open, showing cloning options: 'Clone' (HTTPS, SSH, GitHub CLI) with the URL [https://github.com/jambler24/git\\_le](https://github.com/jambler24/git_le), 'Open with GitHub Desktop', and 'Download ZIP'.

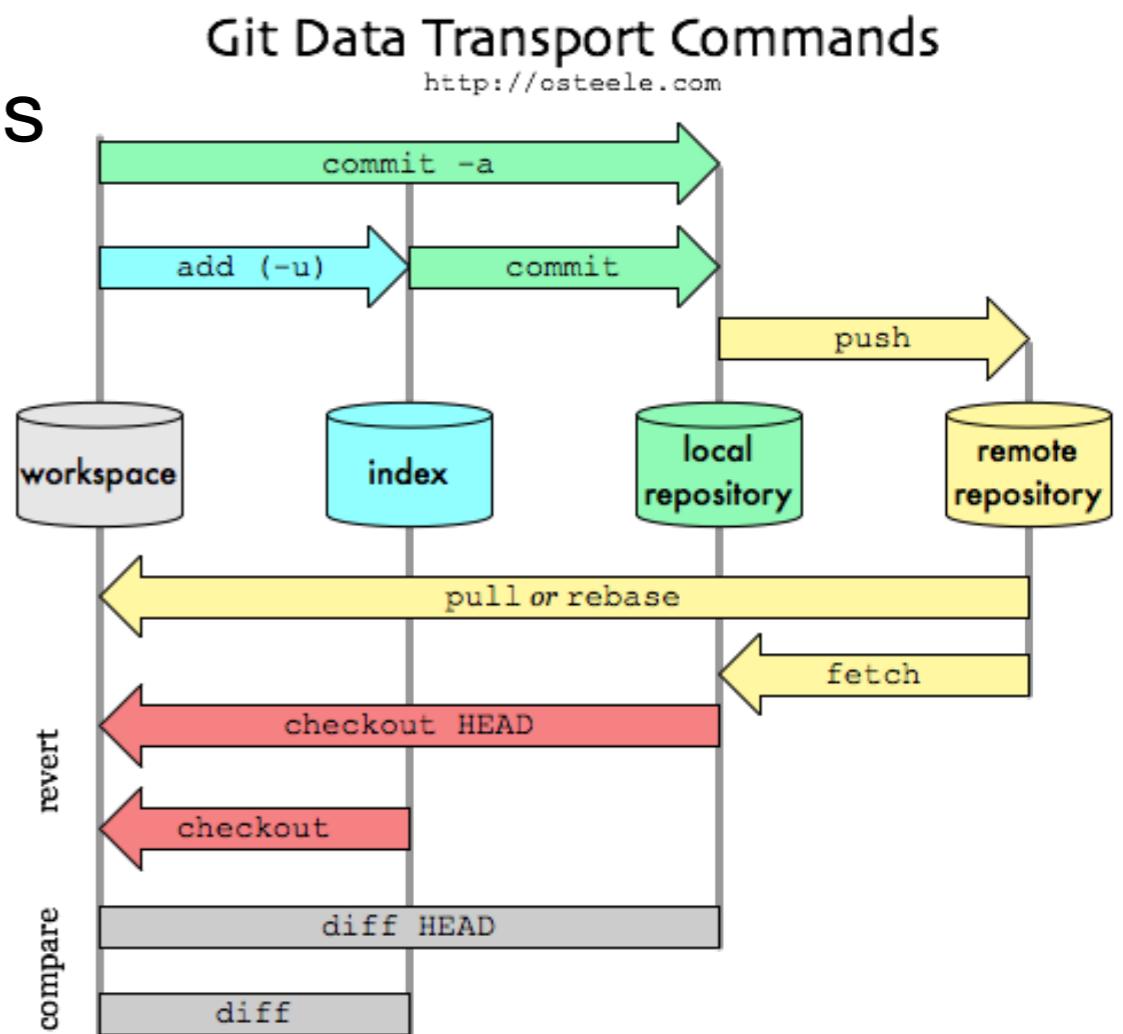


# Remote repositories

- \$ git config

- Configuration stored at 3 levels

- System (--system)
- User (--global)
- Repository



# Remote repositories

- Setting the user identity
  - `$ git config --global user.name "John Doe"`
  - `$ git config --global user.email johndoe@example.com`
- These must be the same as your Github details

# Remote repositories

- Get the data from the remote repo, add it to your local repo:

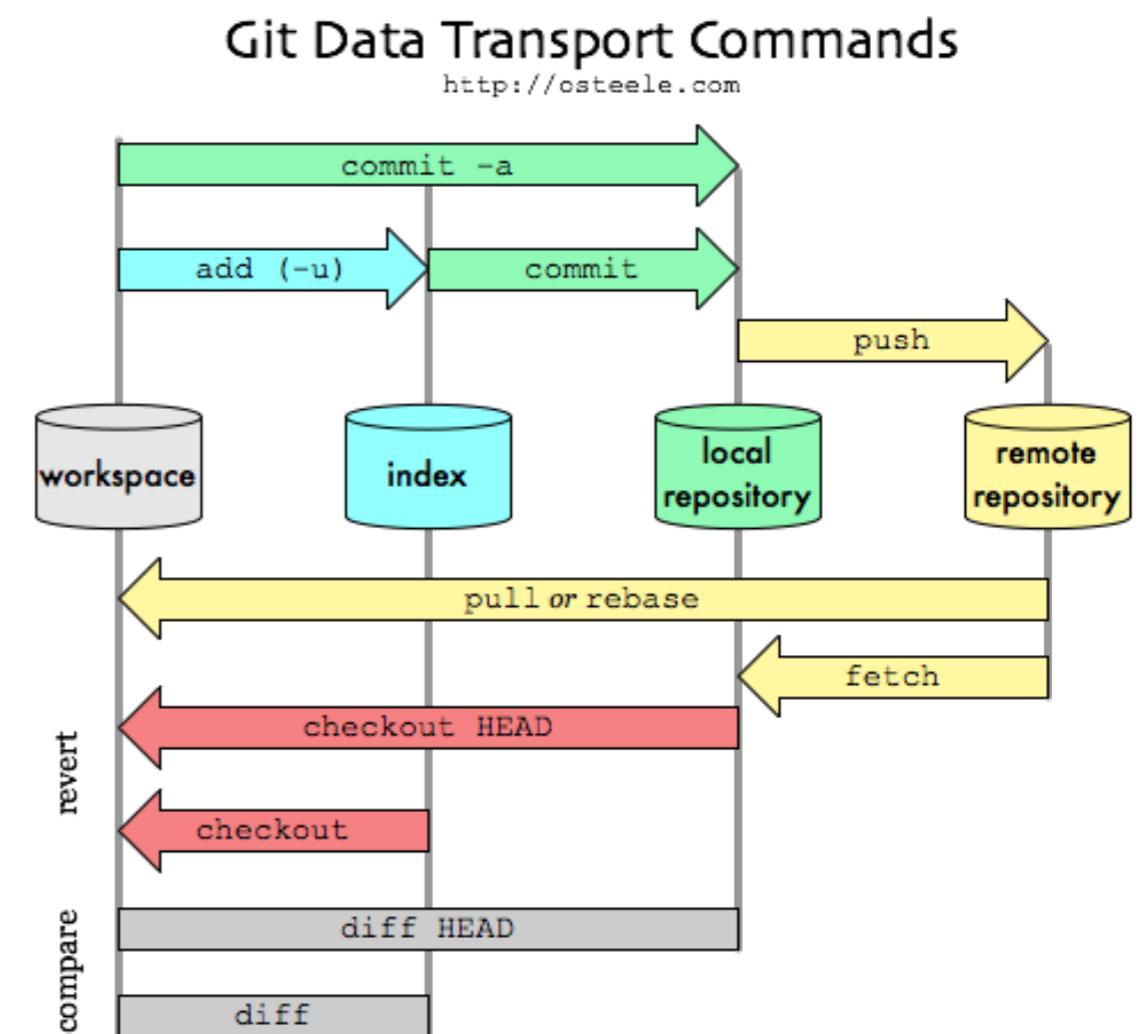
- \$ git fetch

- Get the data from the remote repo, and check it out

- \$ git pull

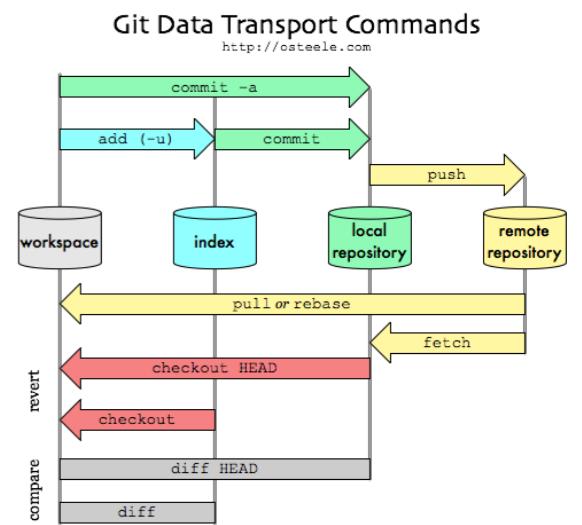
- Send the files from your local repo to the remote repo:

- \$ git push

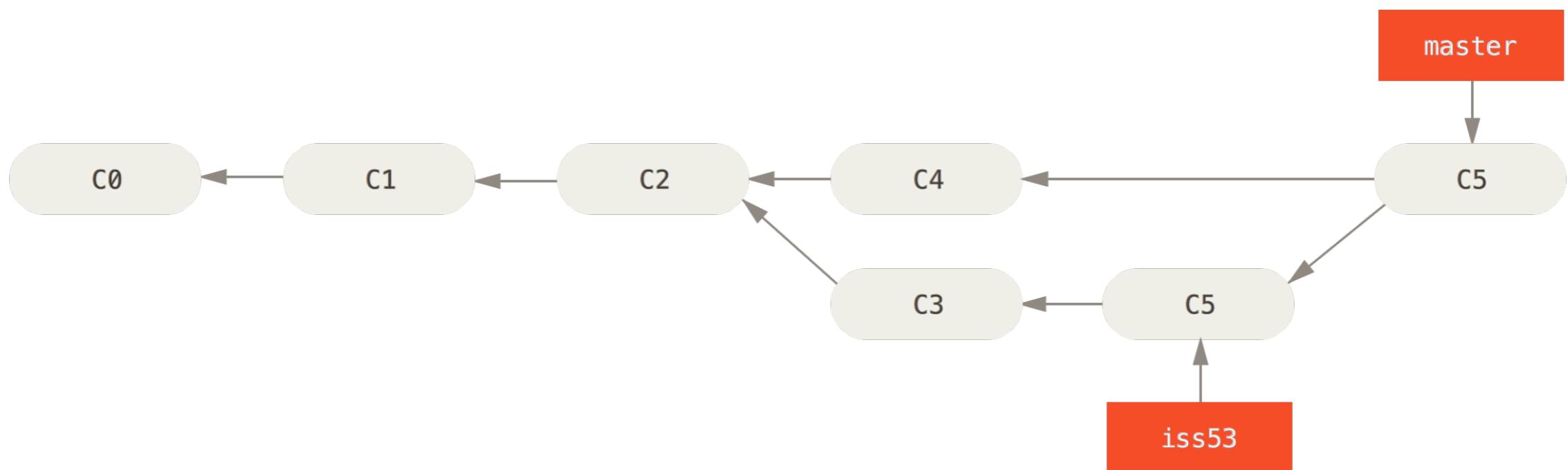


# Remote repositories

- Check what remotes you have added:
  - `$ git remote -v`
- Manually add new remote
  - `$ git remote add <optional shortname> <url>`

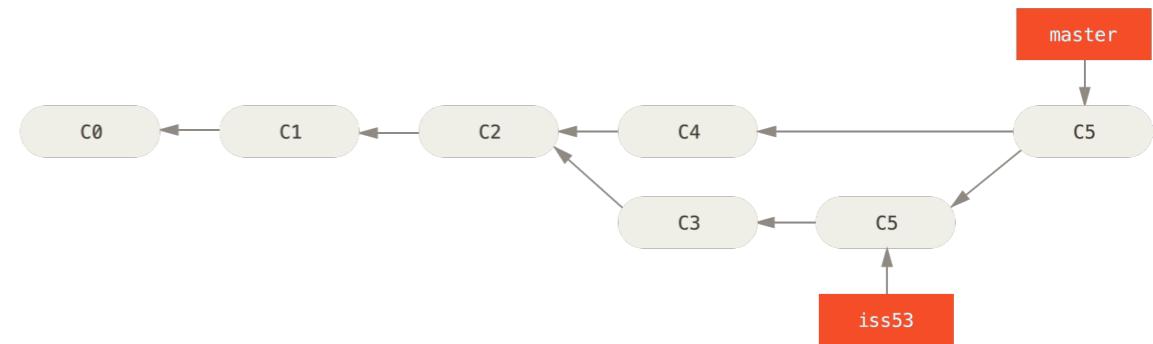


# Branches



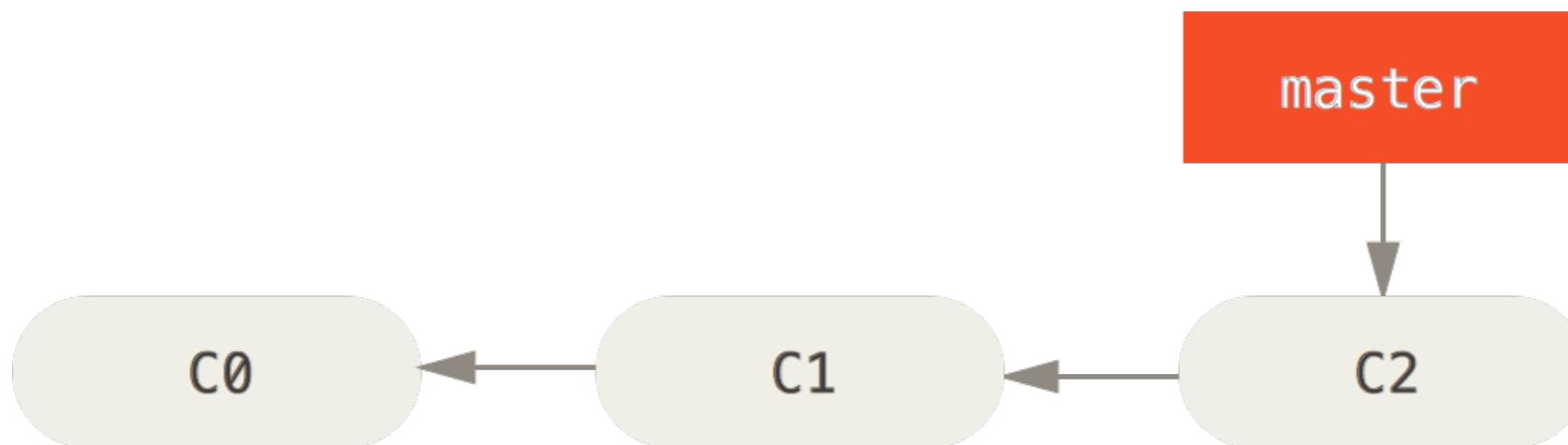
# Branches

- Branches are pointers to a particular commit
  - English: Branches are where code diverges, and each branch has a name
- Allow you to work on parts of large projects individually
- Keep a stable working version of the code
  - This is known as the master / main branch



# Branches

- So far:

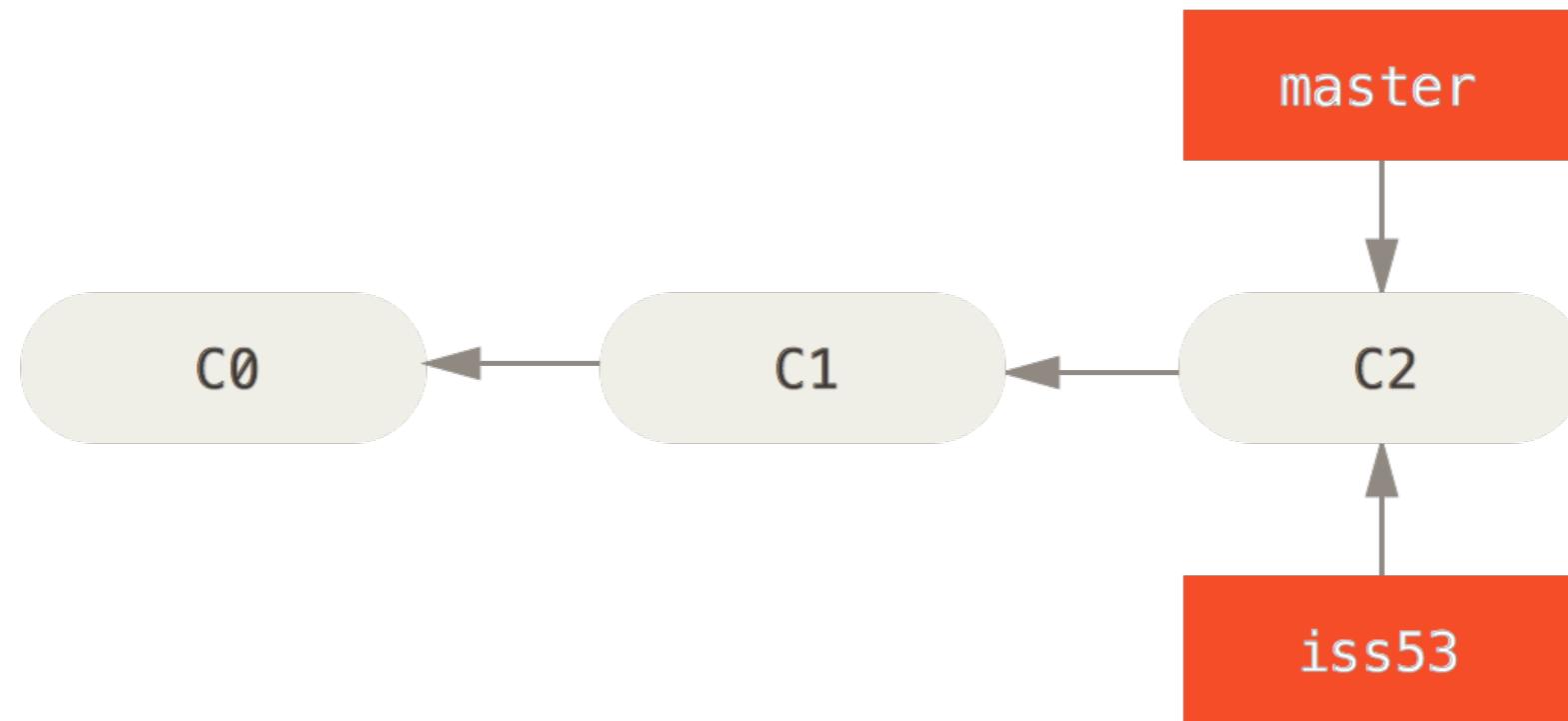


# Branches

- Create a new branch:
  - \$ git branch <branch name>
- Switch to the new branch
  - \$ git checkout <branch name>
- Create & switch to new branch
  - \$ git checkout -b <branch name>
- Wait, where am i?
  - \$ git branch

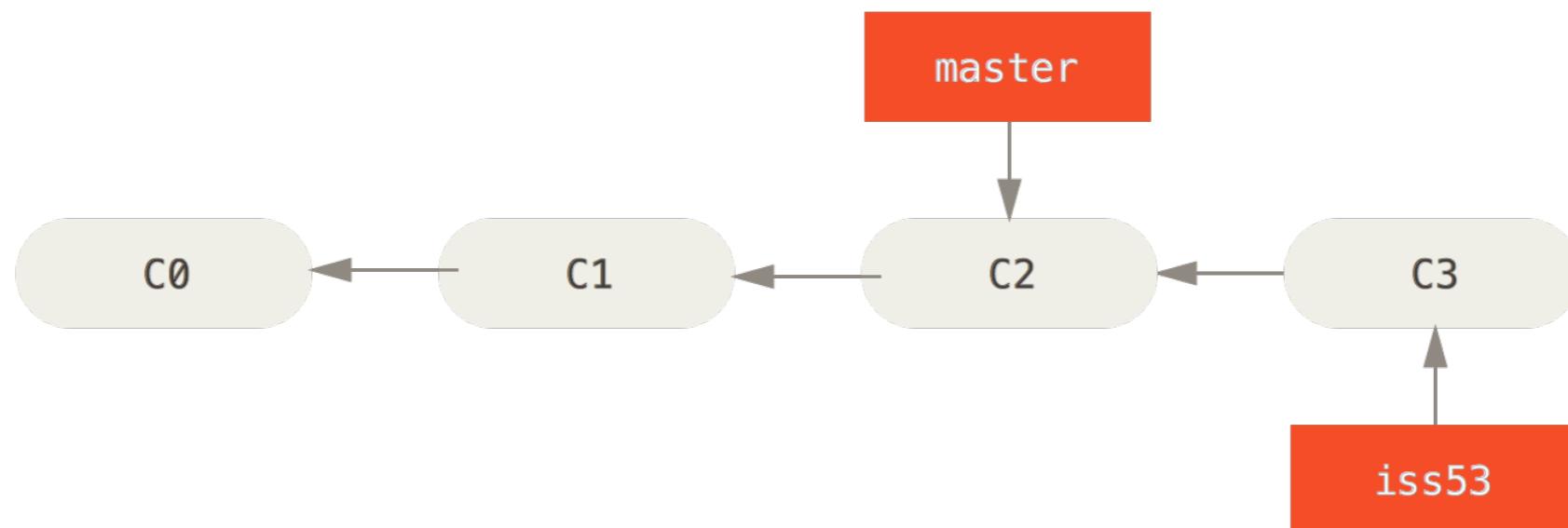
# Branches

- Adding a branch does this:



# Branches

- Doing a commit on a branch does this:

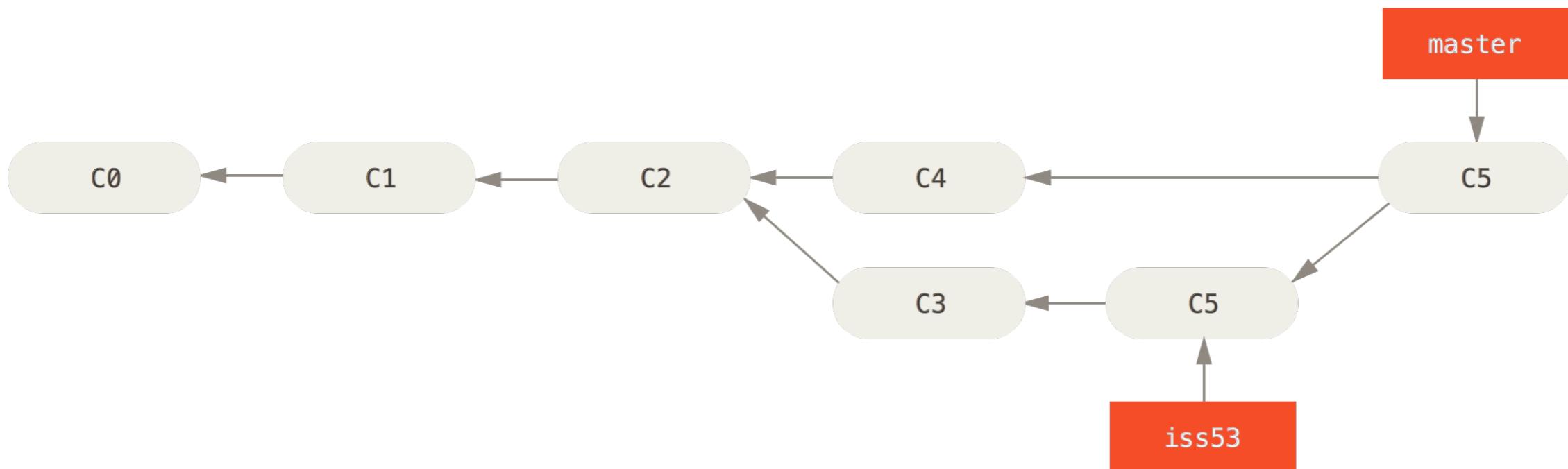


# Branches

- Merging branches
  - `$ git checkout master`
  - `$ git merge <branch name>`
- If the merge is successful, and you are done with the branch
  - `$ git branch -d <branch name>`

# Branches

- Changes from the branch are merged into the master branch



Version Control: Local Changes Log

Branch: All User: All Date: All Paths: All

Unified viewer ▾ 1 difference

Merge branch 'remove-captcha-field-from-cajon Ambler 2020-03-24  
Merge branch 'range-input-slider' into 'release-Jon Ambler 2020-03-24  
Update max age Nico van Zyl 2020-03-22  
Remove value label Nico van Zyl 2020-03-22  
Add age value label Nico van Zyl 2020-03-22  
Implement range slider Nico van Zyl 2020-03-22  
Merge branch 'add-terms-and-conditions-to- Roan Song 2020-03-22  
Merge branch 'refactor-data-capture-form' in Roan Song 2020-03-22  
Merge branch 'feedback-fixes-phase-1' into Nico van Zyl 2020-03-22  
Updates to UserInfoForm Roan Song 2020-03-21  
Removed captcha f .../remove-captcha-f... Roan Song 2020-03-21  
Improve password i .../feedback-fixes-p... Nico van Zyl 2020-03-19  
Fix link Nico van Zyl 2020-03-19  
Add capture success page content Nico van Zyl 2020-03-19  
autocomplete improvements Nico van Zyl 2020-03-19  
Added new table a .../add-terms-and-co... Roan Song 2020-03-18  
Merge branch 'new-capture-ui' into 'release-2 Dean Kayton 2020-03-18  
Fix safari button bug o.../origin & new-captur... Nico van Zyl 2020-03-17  
More style improvements Nico van Zyl 2020-03-17  
Fix kit model bug Nico van Zyl 2020-03-17  
Style improvements Nico van Zyl 2020-03-17  
Improve QR Scanner UX Nico van Zyl 2020-03-17  
Update legal tab content Nico van Zyl 2020-03-17  
Reduce opacity on disabled steps Nico van Zyl 2020-03-17  
Advanced validation Nico van Zyl 2020-03-17

File tree:

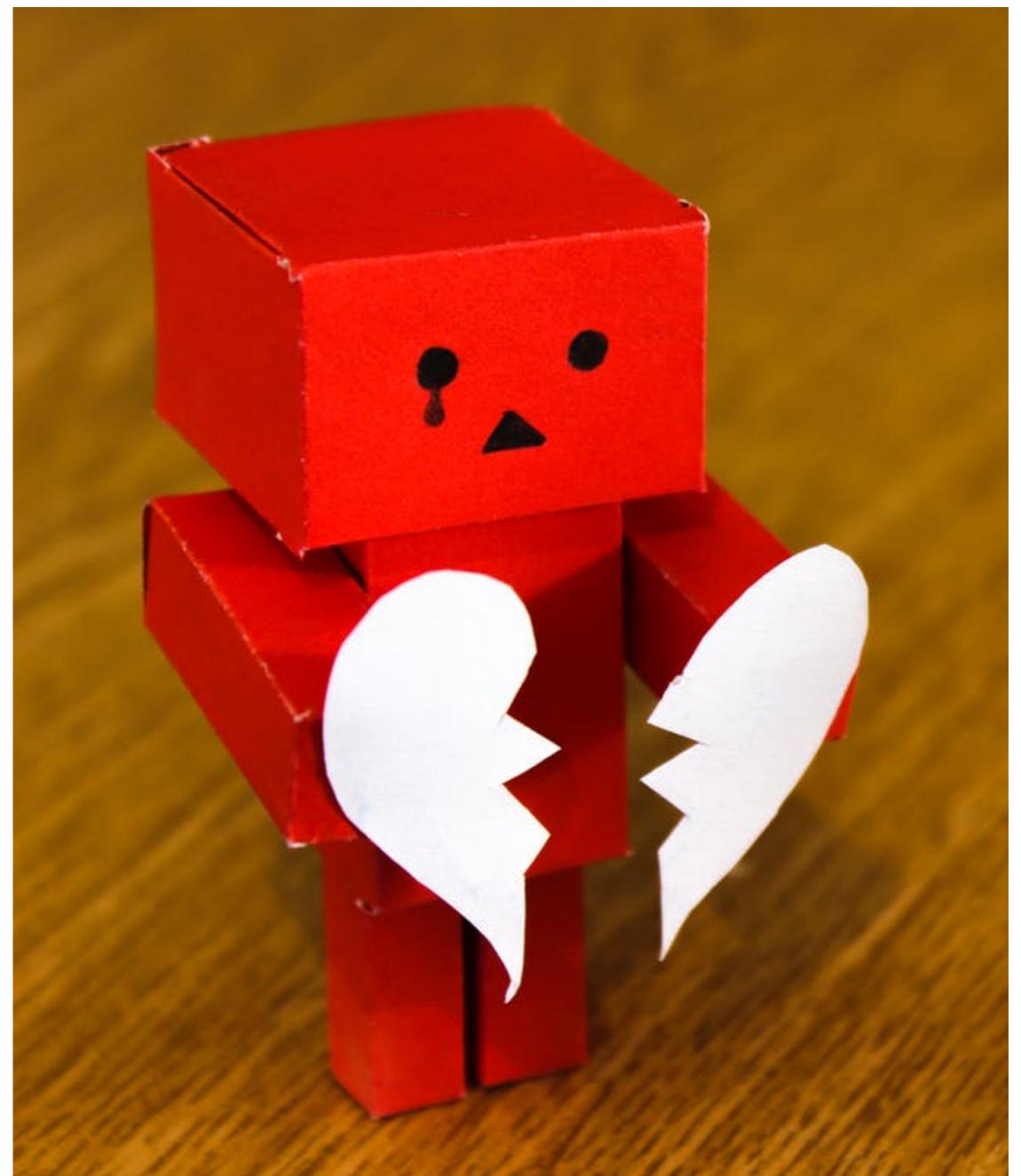
- /Users/panix/Library/Mol
- └ microbiome\_app 1 file
  - forms.py
- └ static/app 2 files
  - dist 1 file
    - app.js
  - src 1 file
    - app.js
- └ templates 1 file
  - capture\_form.html

Code editor:

```
#captcha = ReCaptchaField(widget=ReCaptchaWidget)  
field_order = ['sampleID', 'user_name', 'user_email', 'user_email_confirm']  
  
sampleID = forms.CharField(max_length=128, required=True)  
user_name = forms.CharField(max_length=256, required=True)  
user_email = forms.CharField(max_length=256, required=True)  
user_email_confirm = forms.CharField(max_length=256, required=True)
```

# When things break

- Git forgives, but never forgets



# When things break

- Retrieving old versions
  - First, find the correct version
    - `git log`
  - This gives you all the commits made and the messages associated.
  - `git checkout <commit hash>`
  - `git add <Restored file>`
  - `git commit -m "Fixed it!"`

# When things break

Find the correct version

- git log

This gives you all the commits made and the messages associated.

```
commit 157dcdde57e3a8fe3989113754380ada07a86fda
Author: Jon <jambler24@gmail.com>
Date:   Wed May 18 13:04:48 2016 +0200
```

Added new version of the notes

```
commit 9578fa5e6bb3c707660baec5e8e9ac9ccd689505
Author: Jon <jambler24@gmail.com>
Date:   Thu May 5 11:53:05 2016 +0200
```

Added the slides from other lectures

# When things break

- Conflicts
  - What are they?
    - They happen when git is unable to automatically decide what the merged version of the code looks like
    - Like having 2 supervisors / reviewers review your thesis / paper
  - How do you deal with them?
    - Avoid it (Give your thesis to one supervisor at a time)
    - Get everyone together and review it manually, then give back the intended version

```
# git_lecture
<<<<< HEAD
## Lowercase function info

How to use
=====
## Sections

## Credits

More info
>>>> bugfix-002-typo-docstring
```

# When things break

- Accidentally added files
  - You may accidentally add a database containing sensitive data
  - Or add your entire hard drive
    - `git add /*`
  - To remove them
    - Before commit
      - `git reset`
    - After commit
      - `git rm --cached file1.txt`
      - `git commit -m "remove file1.txt"`
      - `git push origin branch_name`

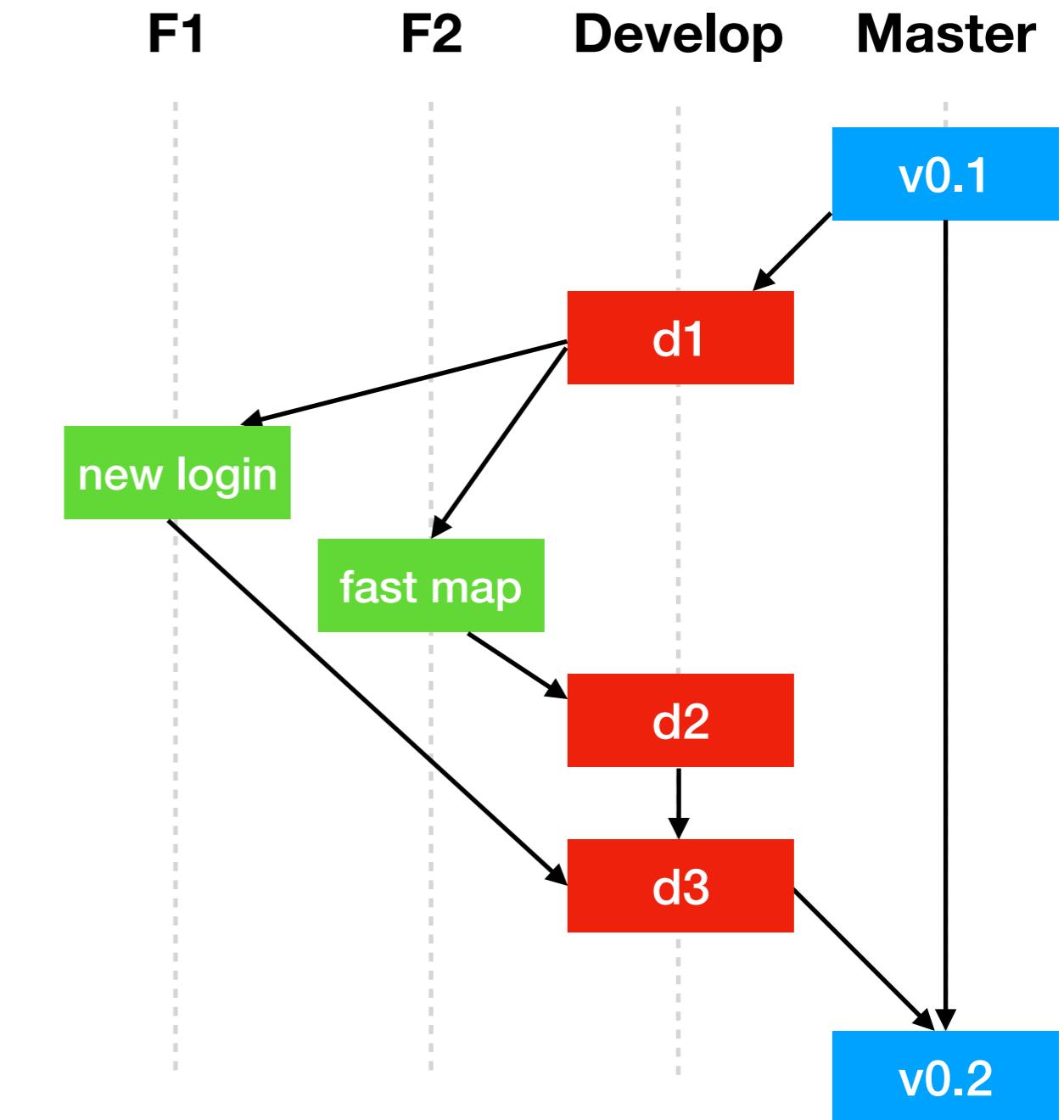
# Working in teams

- Best practices
  - The main branch, testing, and feature branches
  - Naming schemes
  - Merging / Pull requests
  - Code Reviews
  - Kanban

# Working in teams

- Best practices

- The main branch, testing, and feature branches
- Main / master
  - Stable, working code
  - Always available for users
- Testing / develop
  - For doing testing
  - For merging in features
- Feature
  - New functions, enhancements, or bug fixes



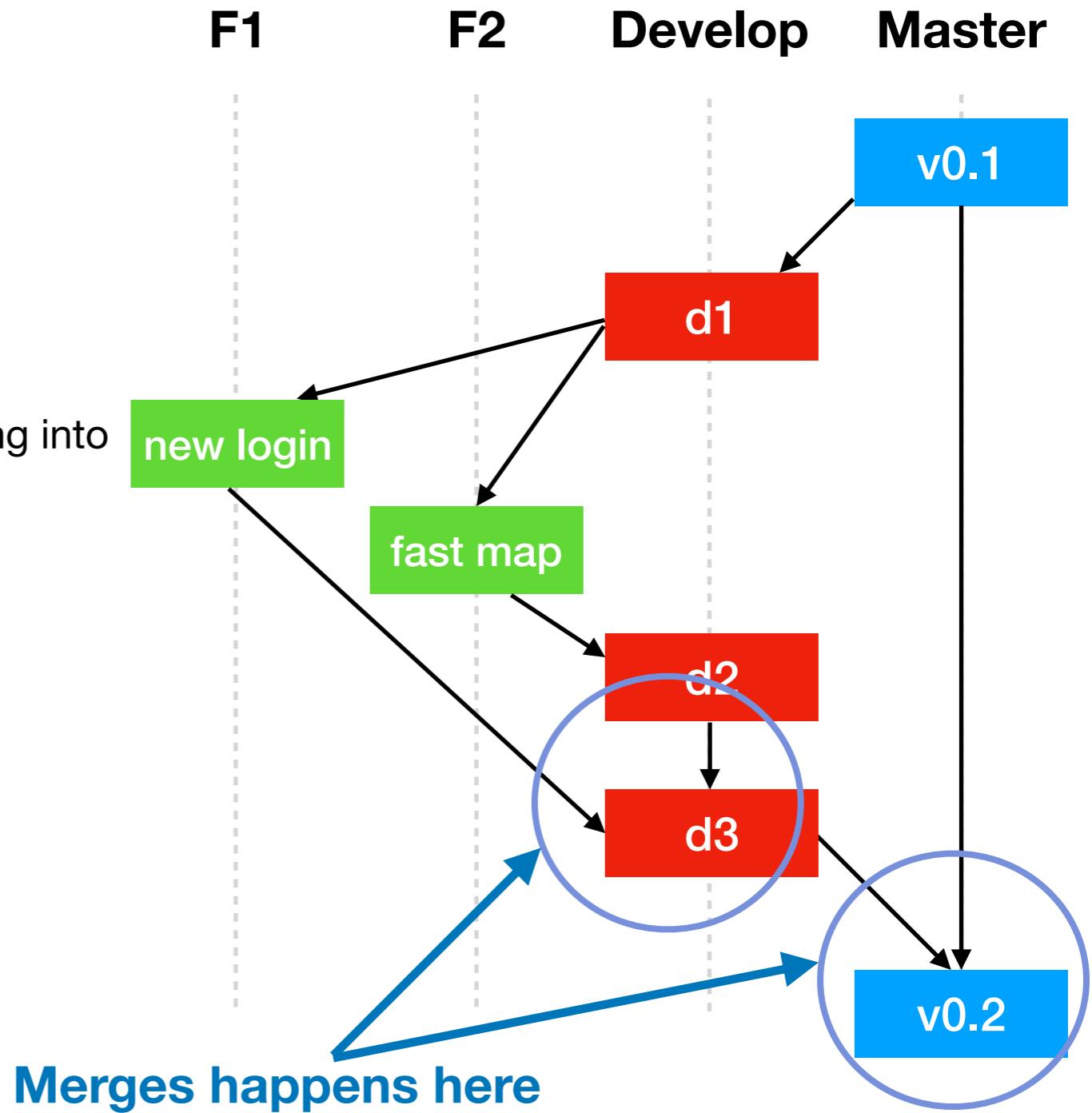
# Working in teams

- Best practices
    - Naming schemes
      - Use “-“ or “\_”
      - Description
      - Prefix / suffix
        - feature, bugfix, merge
      - Link to task / issue
        - See Kanban board
- Examples:
- develop-903-new-function
- bugfix-412-login-issue

# Working in teams

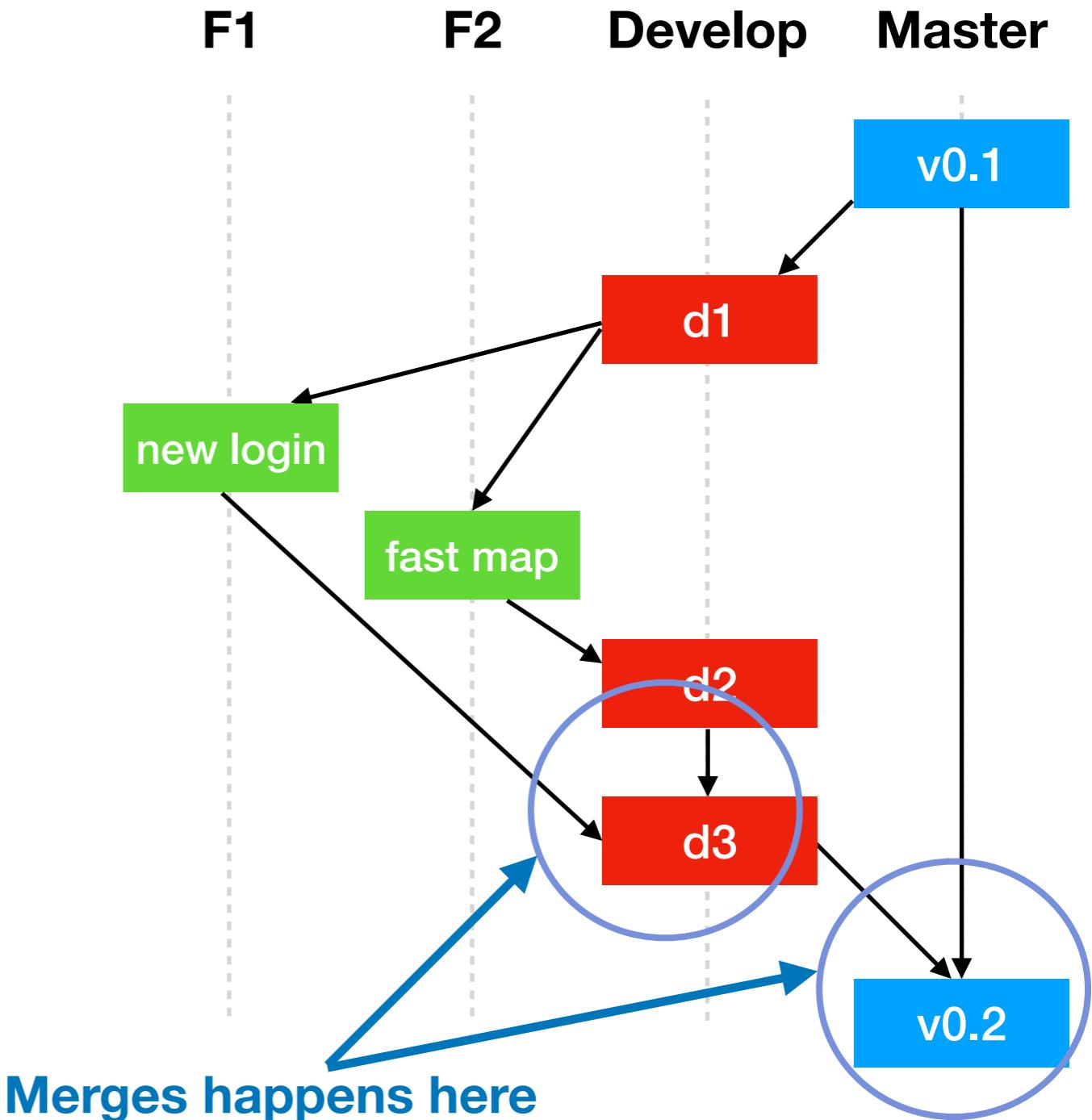
- Best practices

- Merging
  - Merge two branches together
  - Command line
    - Change to branch you are merging into with:
      - git checkout develop
      - Merge in the other branch
      - git merge d3
    - Push the change!
      - git push



# Working in teams

- Best practices
  - Pull requests
    - Merge two branches together
    - Github



🕒 testing had recent pushes 1 minute ago

Compare & pull request

🕒 bugfix-002-typo-docstring had recent pushes 1 minute ago

Compare & pull request

🕒 feature-001-lowercase-function had recent pushes 1 minute ago

Compare & pull request

🕒 main ▾

🕒 4 branches

🏷 0 tags

Go to file

Add file ▾

⬇ Code ▾



jambler24 Added a test program

3ffb8c1 3 days ago ⏲ 3 commits

📄 .gitignore

Added gitignore

3 days ago

📄 README.md

Initial commit

3 days ago

📄 program.py

Added a test program

3 days ago

README.md



# git\_lecture

# Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

# Testing #1

[Edit](#) [Open with ▾](#)[Open](#) jambler24 wants to merge 4 commits into [main](#) from [testing](#) [Conversation 0](#)[Commits 4](#)[Checks 0](#)[Files changed 2](#)[+15 -2](#) 

jambler24 commented now

[Owner](#)

...

Reviewers



No reviews

Still in progress? Convert to draft

Merging the 2 features:

- Lowercase function
- README edits

 jambler24 added 4 commits 15 minutes ago

195bae9

 Added lower case function

4ae6bf8

 Added info to README

30555f0

 Added info on lowercase function

24f181f

 Fixed readme

Assignees



jambler24

Labels

[documentation](#) [enhancement](#)

Projects



None yet

Milestone



No milestone

Linked issues



Successfully merging this pull request may close these issues.

 jambler24 added [documentation](#) [enhancement](#) labels now jambler24 self-assigned this now

# Testing #1

[Edit](#)[Open with ▾](#)

[Open](#) jambler24 wants to merge 4 commits into `main` from `testing`

[Conversation](#) 0[Commits](#) 4[Checks](#) 0[Files changed](#) 2

+15 -2

[Changes from all commits ▾](#) [File filter... ▾](#) [Jump to... ▾](#) [⚙️ ▾](#)

0 / 2 files viewed

[i](#)[Review changes ▾](#)

▼ 12 README.md

[Viewed](#) [...](#)

... @@ -1,11 +1,11 @@

1 - `# git_lecture`

1 + `# git_lecture`  
2 +  
3 + `## Lowercase function info`  
4 +  
5 + `How to use`  
6 +  
7 + `## Sections`  
8 +  
9 + `## Credits`  
10 +  
11 + `More info`

▼ 5 program.py

[Viewed](#) [...](#)

↑ @@ -4,7 +4,7 @@

4 DNA = "ACGACTCGAGCTAGCGTACGAGCTCAGCTAGC"

4 DNA = "ACGACTCGAGCTAGCGTACGAGCTCAGCTAGC"

5

5

6

6

7 - `def create_kmers(sequence, kmer_size):`

7 + `def create_kmers(sequence, kmer_size, make_lower=False):`

8     """\p

8     """\p

9         This function creates a list of kmers of a given length from a given string

9         This function creates a list of kmers of a given length from a given string

 **jambler24** self-assigned this 1 minute ago

 **jambler24** commented now

Author Owner  

Looking good!

Add more commits by pushing to the **testing** branch on **jambler24/git\_lecture**.

 **Continuous integration has not been set up**  
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**  
Merging can be performed automatically.

**Merge pull request** ▾ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

 Write Preview H B I         

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

 **Close pull request**  **Comment**

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Version Control: Local Changes Log			
<input type="text"/> Branch: All ▾ User: All ▾ Date: All ▾ Paths: All ▾			
Merge pull request #1 from jambler24/testing		origin & main Jambler* 2020-11-29 18:04	
Fixed readme		origin & testing Jon 2020-11-29 17:57	
<b>Added info on lowercase function</b>		origin & feature-001-lowercase-... Jon 2020-11-29 17:54	
<b>Added info to README</b>		origin & bugfix-002-typo-docst... Jon 2020-11-29 17:52	
<b>Added lower case function</b>			Jon 2020-11-29 17:48
<b>Added a test program</b>			Jon 2020-11-26 16:59
<b>Added gitignore</b>			Jon 2020-11-26 15:23
<b>Initial commit</b>			Jambler* 2020-11-26 15:15

# Working in teams

- Best practices
  - Code Reviews
    - Tie into merge requests
    - Team comes together to go over the code
    - Coder presents, but very informal
    - Make sure everyone understands what was done and why
  - Get feedback
  - Find bugs or optimisations
  - When needed, often when doing a merge into testing / develop

# Working in teams

- Best practices
  - Kanban
  - A workflow management method
  - Keep track of tasks
    - To do, in progress, done



How TOYOTA's original Kanban looked  
Source: TOYOTA Global Website

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

## Organize your issues with project boards

[Learn More](#)[Create a project](#)

Did you know you can manage projects in the same place you keep your code? Set up a project board on GitHub to streamline and automate your workflow.



### Sort tasks

Add issues and pull requests to your board and prioritize them alongside note cards containing ideas or task lists.



### Plan your project

Sort tasks into columns by status. You can label columns with status indicators like "To Do", "In Progress", and "Done".



### Automate your workflow

Set up triggering events to save time on project management—we'll move tasks into the right columns for you.



### Track progress

Keep track of everything happening in your project and see exactly what's changed since the last time you looked.



### Share status

Each card has a unique URL, making it easy to share and discuss individual tasks with your team.



### Wrap up

After you wrap up your work, close your project board to remove it from your active projects list. On to the next project!

 Code  Issues  Pull requests  Actions  Projects  Wiki  Security  Insights  Settings

## Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

### Project board name

Hackathon sprints!

### Description (optional)

Sprints for during the H3Africa Hackathon 2020!

### Project template

Save yourself time with a pre-configured project board template.

Template: Basic kanban ▾

**Create project**



Search or jump to...

Pull requests Issues Marketplace Explore



Project created from Basic kanban template.

jambler24 / git\_lecture

Unwatch 1

Star 0

Fork 0

Code Issues Pull requests Actions Projects 1 Wiki Security Insights Settings

### Hackathon sprints!

Updated 2 minutes ago

Filter cards

Add cards Fullscreen More

2 To do + ...

Enter a note

Add

Cancel

Make more slides on Kanban ...

Added by jambler24

More documentation on new lowercase function ...

Added by jambler24

2 In progress + ...

Break things!

Added by jambler24

Bug: No AI used in project. Find a use, it is on trend.

Added by jambler24

1 Done + ...

Implement lowercase function

Added by jambler24

Add column



Search or jump to...

Pull requests Issues Marketplace Explore



jambl24 / git\_lecture

Unwatch 1

Star 0

Fork 0

Code Issues 1 Pull requests Actions Projects 1 Wiki Security Insights Settings

Filters ▾

is:issue is:open

Labels 9

Milestones 0

New issue

1 Open  0 Closed

Author ▾

Label ▾

Projects ▾

Milestones ▾

Assignee ▾

Sort ▾

Does not run on windows 98 bug



2

#2 opened 19 minutes ago by jambl24

💡 ProTip! Add no:assignee to see everything that's not assigned.



© 2020 GitHub, Inc.

Terms

Privacy

Security

Status

Help

Contact GitHub

Pricing

API

Training

Blog

About

# Does not run on windows 98 #2

[Edit](#)[New issue](#)

jambler24 opened this issue 1 minute ago · 2 comments



jambler24 commented 1 minute ago

[Owner](#)[...](#)

Tried running this on windows 98, didn't work.

jambler24 self-assigned this now

jambler24 added the **bug** label now

jambler24 added this to To do in Hackathon sprints! now



jambler24 commented now

[Author](#)[Owner](#)[...](#)

On it!



jambler24 commented now

[Author](#)[Owner](#)[...](#)

Create win98 VM using Docker

Assignees



jambler24



Labels



Projects



To do ▾



Milestone

No milestone



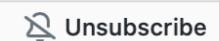
Linked pull requests

Successfully merging a pull request may close this issue.

None yet

Notifications

Customize



You're receiving notifications because you're

# Does not run on windows 98 #2

[Edit](#)[New issue](#)

jambler24 opened this issue 1 minute ago · 2 comments



jambler24 commented 1 minute ago

[Owner](#)[...](#)

Assignees



jambler24



Labels

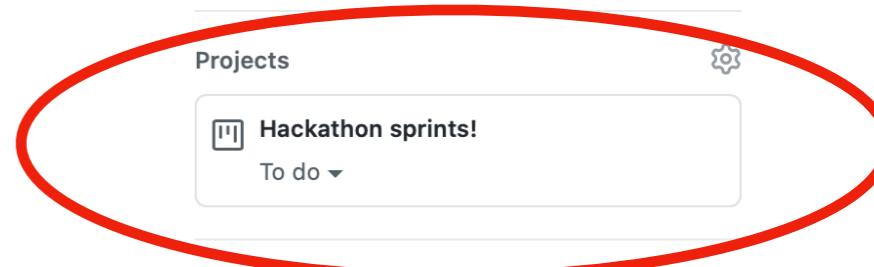


Projects



Hackathon sprints!

To do ▾



Milestone

No milestone



Linked pull requests

Successfully merging a pull request may close this issue.



None yet

Notifications

Customize

[Unsubscribe](#)

You're receiving notifications because you're



jambler24 commented now

[Author](#)[Owner](#)[...](#)

On it!



jambler24 commented now

[Author](#)[Owner](#)[...](#)

Create win98 VM using Docker



Search or jump to...

Pull requests Issues Marketplace Explore



jambl24 / git\_lecture

Unwatch 1

Star 0

Fork 0

Code Issues 1 Pull requests

Actions

Projects 1

Wiki

Security

Insights

Settings

### Hackathon sprints!

Updated 43 seconds ago

Filter cards

Add cards Fullscreen More

3 To do + ...

Make more slides on Kanban ...

Added by jambl24

More documentation on new lowercase ...

function

Added by jambl24

Does not run on windows 98 ...

#2 opened by jambl24

bug



2 In progress + ...

Break things! ...

Added by jambl24

Bug: No AI used in project. Find a use, it ...

is on trend.

Added by jambl24

1 Done + ...

Implement lowercase function ...

Added by jambl24

Add column

# Continuous Integration and Continuous Deployment

- Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project, and includes tools that assert the new code's correctness before integration
- Continuous Deployment (CD) uses automated testing to validate if changes to a codebase are correct and stable for immediate autonomous deployment to a production environment.

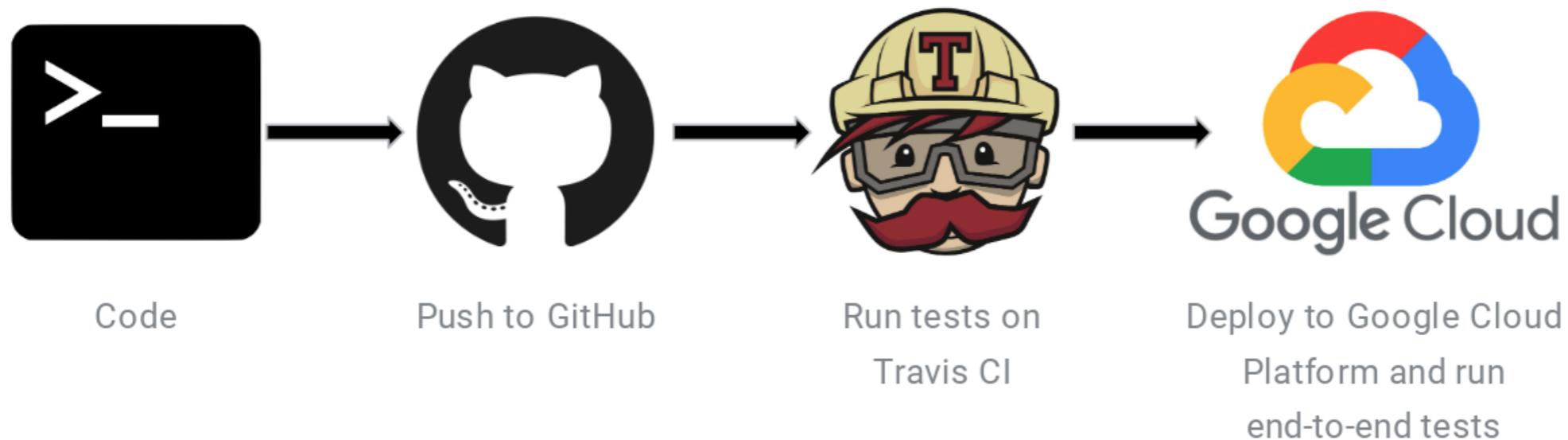


GitLab



Travis CI

# Continuous Integration and Continuous Deployment



README.md edit

build  passing  docker build  manual

## GenGraph

---

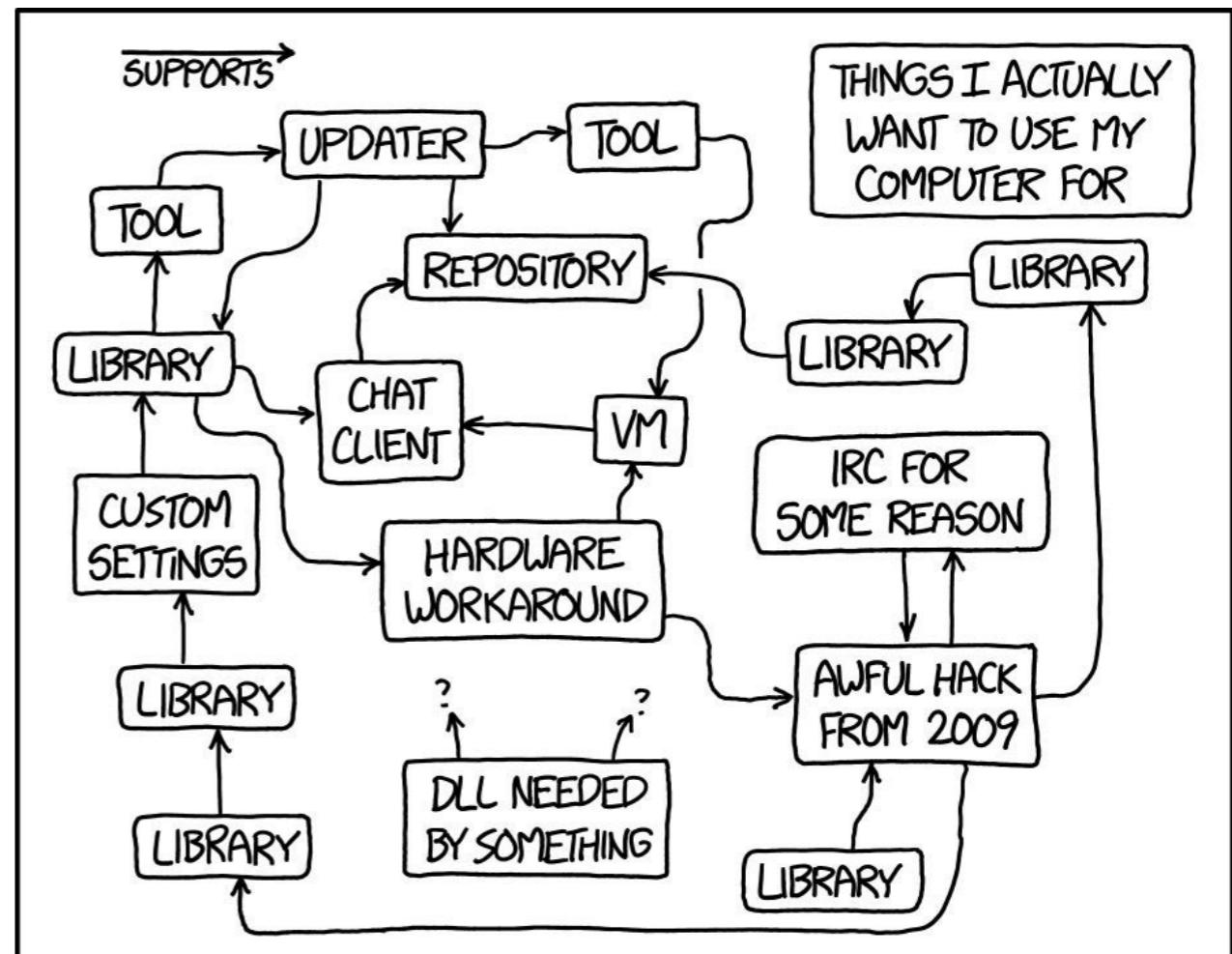
A repository for the GenGraph toolkit for the creation and manipulation of graph genomes

This project aims to make genome graphs simple to create, and provide tools for their manipulation and integration into common workflows.

Please refer to the [wiki](#) for installation and usage instructions.

# Scientific workflows

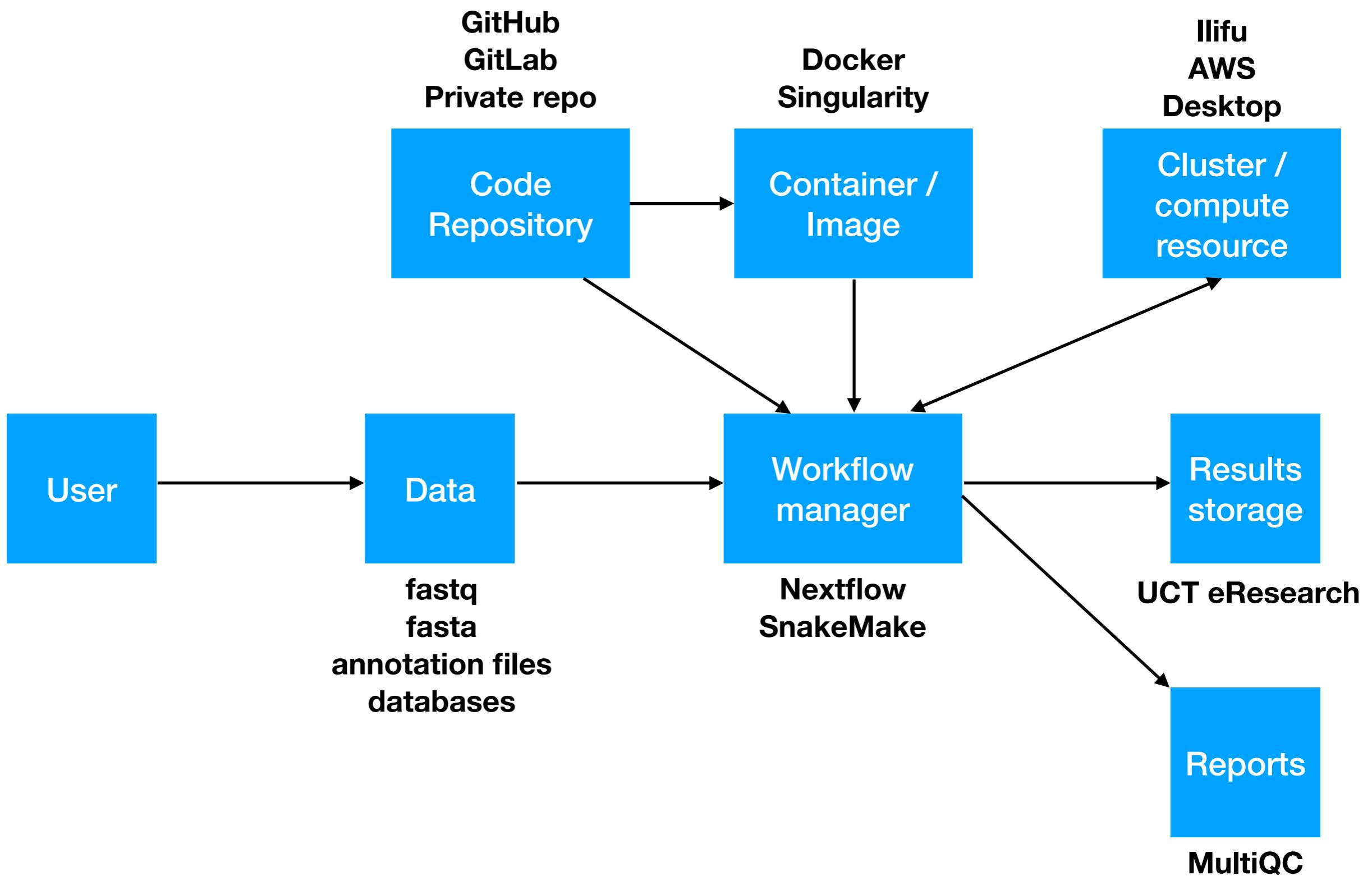
- Create reproducible research
- Scalable pipelines
- Platform agnostic
- Simplify installation process
- Write it once, write it well



EVERY NOW AND THEN I REALIZE I'M MAINTAINING A  
HUGE CHAIN OF TECHNOLOGY SOLELY TO SUPPORT ITSELF.

<https://xkcd.com>

# Workflow template



# Practical

