



Git Good

an Introduction to GitHub for Collaboration



Cassandra Gould van Praag
Executive Director, RCM Cooperative
TOSI Open Scientist in Residence 2025



Emma Karoune
Principle Researcher, Alan Turing Institute
Co-Founder, RCM Cooperative

Contributors ✨

Thanks goes to these wonderful people ([emoji key](#)):

<u>Cassandra</u> <u>Gould van</u> <u>Praag</u> 	<u>Hari Sood</u> 	<u>Sophia</u> <u>Batchelor</u> 	<u>Esther</u> <u>Plomp</u> 	<u>Anne Lee</u> <u>Steele</u> 	<u>Emma</u> <u>Karoune</u> 	<u>GabinWK</u>
<u>Malvika</u> <u>Sharan</u> 	<u>Eirini</u> <u>Zormpa</u> 	<u>Ayesha</u> <u>Magill</u> 	<u>Alexandra</u> <u>Araujo</u> <u>Alvarez</u> 	<u>Vicky</u> <u>Hellon</u> 	<u>andrepiza</u> 	

All illustrations by Scriberia for
The Turing Way
used under CC-BY4.0 License

This project follows the [all-contributors](#) specification. Contributions of any kind welcome!

This template has been
created by [Slidesgo](#)



Housekeeping

- Slides and list of links available at:
 - <https://github.com/the-turing-way/gitgood>
- Recording: **Camera off / chat** if you don't want to be attributed
- Add questions to **zoom chat**
- Use **raise hands to speak** and **use zoom reactions** to let us know when you have finished an activity
- We are operating under the McGill Faculty of Medicine and Health Sciences **Code of Conduct**
 - "...demonstrate respect and appreciation for [your] colleagues..."
 - <https://www.mcgill.ca/medhealthsci/about/our-vision-mission-values/code-conduct>
 - See Appendix 1 for reporting procedures
- You will need a **GitHub account** to follow along with the tutorials
 - **Make one now if you haven't got one already!**
- All exercises will use the **online GitHub interface**
 - Does not require git to be installed or security set up
- Julia Pfarr (ORIGAMI Lab) Git/Hub 101 at Mtl BrainHack 2026:
 - https://julia-pfarr.github.io/git-gitlab_tutorial/
- Kendra Oudyk (ORIGAMI Lab) Intro to Git/GitHub OSHO Feb 2025
 - <https://www.youtube.com/@OpenScienceOfficeHours>



Aims

- Feel more confident in:
 - Interacting with GitHub;
 - Contributing to other peoples GitHub materials;
 - Setting up your own repositories to receive contributions;
- Know where to find help with any of this!

Agenda

- Part 1: What is Git & Github?
- Exercise 1: Finding an Issue and contributing to a project
 - You will practice: Using issues to contribute and document decisions, markdown
- Part 2: What is open source?
- Part 3: Using GitHub online and in the terminal (demo)
- Exercise 2: Editing a file and making a Pull Request
 - You will practice: Markdown; contributing, pull requests
- Part 4: What can we make with Github? (Websites and Project Management)
- Exercise 3 (take home): Make your own website



What is GitHub?



- **Sounds like it's only for programmers!**
- GitHub is also great for version control of anything with letters/numbers/characters (and images) in and even better for working collaboratively!
- GitHub enables:
 - Making and sharing of something which looks professional
 - Tracking of changes as your project grows
 - Integration of contributions from others
 - Maintenance of an open “To Do” list
 - Licensing and DOI’ing materials, making them reusable and citable



Git vs GitHub



- Software which enables for version control.
- Free and open source!
- Often pre-installed on your computer



- A popular service for hosting and sharing projects you have been tracking with git.
- Secure and flexible control of access to materials.





Collaborative docs



- Multiple people contributing to the same information from different times/places.

Hand/Chat:
What might be challenges with distributed collaboration?

Hand/Chat:
How could version control help?





Making distributed collaboration possible



Version control



Reviewing & updating



Attribution to contributors



Transparency & accountability



Project management



Share globally



Navigating

- Files:
 - README
 - CONTRIBUTING
 - CODE_OF_CONDUCT
 - LICENSE
- Issues
- Pull requests
- Projects
- Contributors
- File views:
 - Preview
 - Code
 - Blame
 - History
 - Commits
 - Diff
- Repo vs organization
- Organisation members
- Repo collaborators

Chat: Any buttons you want to see or understand?



Exercise 1: Commenting

- Go to the practice space repo **issues** and complete issue #3
- Take a look at the **markdown** cheat sheet to add some formatting to your comment
- Use zoom reaction when you are done

Collaboration
practice!



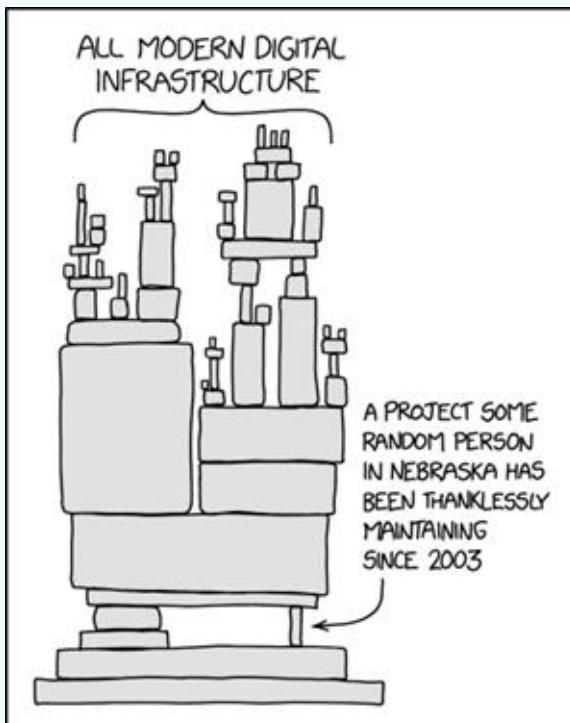
Discussion

- What are the pros and cons of using a platform like GitHub for:
 1. Your own code and documents?
 2. Shared (private) code and documents?
 3. Shared (public) code and documents?
- **What are your questions?**

Hands /
Chat



Part 2: Open Source



Chat: What open source software do you use?

xkcd, 'Dependency': <https://xkcd.com/2347/>

Open Source Initiative (OSI) Legal definition:

- The software is free to redistribute
- The source code is included
- "Derived works" (remixes) are allowed
- There is no discrimination against persons, groups, or fields of endeavour (e.g., you cannot ban a specific company or country from using it)
- Roots in free software movement
- Developed into free and open source software (F/OSS) movement
- Influenced other 'open' movements: open knowledge, open data, open access, open science

Why do we need licenses?

The Legal Default is "Closed"

- By international copyright law, any original work (code, text, or data) is **automatically protected by "All Rights Reserved"** the moment it is created.
- If you put materials on GitHub without a license, other researchers can look at it, but they cannot **legally** download, run, or build upon it.
- A license tells the world, "**I own this, but I am giving you legal permission to use it under these specific conditions.**"

Safeguarding "Safe Sharing"

- Researchers often hesitate to share because they fear their work will be used in ways they didn't intend. Licenses create a contract.
- Liability Protection: Almost all open licenses include a "Disclaimer of Warranty." which protects you from being sued if someone else uses your code and it fails or produces an error.

Credit as The Currency of Academia

- Academia runs on citation and reputation.
- In traditional publishing, we have established norms for citing papers. For code and documentation, licenses like MIT or CC-BY turn "attribution" from a polite suggestion into a legal requirement.
- By licensing your "non-paper" outputs (datasets, software, tutorials), you are forcing the academic ecosystem to treat those contributions as valuable, citable objects.



Licenses & copyright



Copyright

All Rights Reserved

Reuse requires the permission from the copyright owner

Open License

Some Rights Reserved

Re-use is permitted without permission under the specific limitations of the license

Public Domain

No Rights Reserved

May be used without permission

- **Permissive Licenses**

- "You can do whatever you want with this, as long as you give me credit (attribution) and don't hold me liable."
- Examples: **MIT, Apache 2.0, BSD, CC-BY-4.0**
- Good: Encourage maximum adoption. No issues with IP or license stacking
- Bad: Changes can be relicensed as closed

- **Copyleft (reciprocal) Licenses**

- "If you use my open materials, you must make your derived materials open too"
- Examples: **GNU GPL (v2 or v3), AGPL, CC-BY-SA**
- Good: Prevents enclosure of the commons
- Bad: Hesitancy to force openness

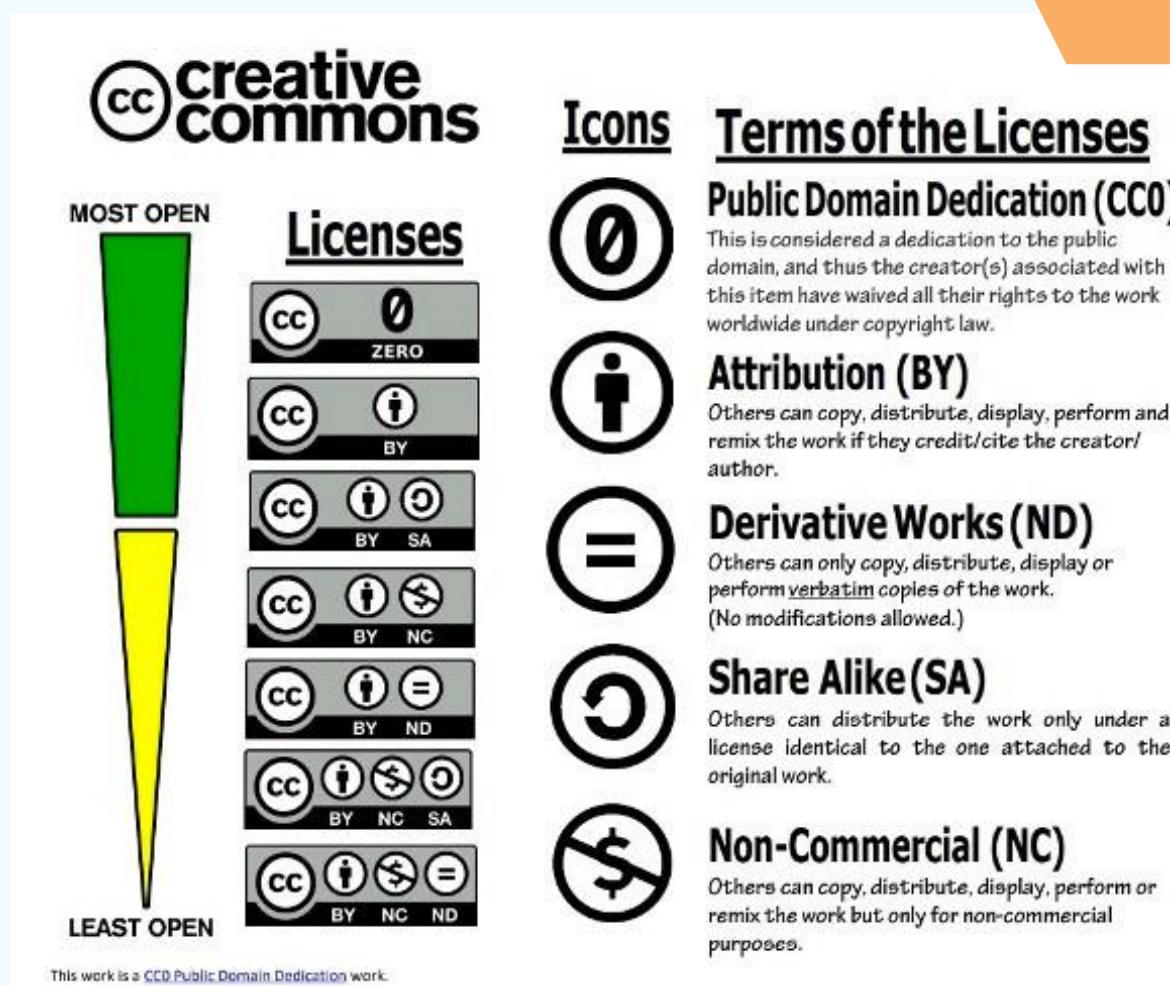
Go look for a software license



Licenses for docs

- When you publish in Neuroimage, you are asked to choose from one of Elsevier's approved "open access user licenses":
 - Creative Commons Attribution (**CC BY**)
 - Creative Commons Attribution-NonCommercial (**CC BY-NC**)
 - Creative Commons Attribution-NonCommercial-NoDerivatives (**CC BY-NC-ND**)
 - Copyright Elsevier or a Society
 - Elsevier user license

Which is
most / least
permissive?



Ohio State University, CC0



Licenses for docs

Why do docs have different licenses than code?

"Software" vs. "Literary Works"

- Code Licenses (MIT, GPL): are designed for functional works. They deal with things like "compilation," "object code," and "execution."
- Creative Commons (CC): Are designed for literary and artistic works. They cover "reproduction," "adaptation," and "display."
- Most code licenses don't explicitly address the rights needed for documentation, such as the right to translate a book into another language or the right to print and sell physical copies.

The "Attribution" Problem

- In the MIT license, the requirement is to keep the "copyright notice and permission notice" in all copies of the software (LICENSE.md)
- In documentation, CC-BY gives you more flexibility on how you give credit (e.g., a "Credits" page or a footer).
- Using MIT for a book might legally require you to print the entire MIT license text on every single page or diagram.

How would
you like your
research
outputs to be
licensed?



Licenses for hardware

- Follow similar permissive <> copyleft spectrum

The "Right to Manufacture"

- "Copying" involves buying components, soldering, and assembly.
- A hardware license gives you the legal right to manufacture a physical version of the design.
- Without a hardware-specific license, someone might have the right to look at your circuit diagram (protected by copyright), but they might not technically have the right to build and sell the physical item if the "utility" isn't covered by the license.

The Patent "Safety Net"

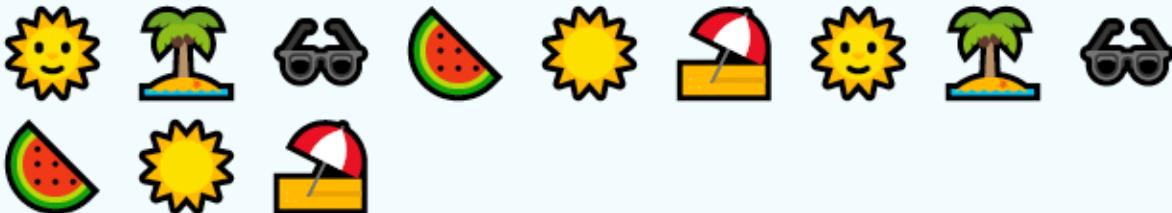
- Copyright protects how ideas are written down. Patents protect how things work.
- If you share a new sensor design under a documentation license, a company could file a patent on the "invention," and then sue you for making your own device.
- Licenses like CERN-OHL or Solderpad include "Patent Grants." This means anyone who contributes to the project effectively says: "I promise not to sue anyone for using any patents I have that are part of this design."

The "Physical" Attribution

- Hardware licenses often include clauses about "Notice Retention" on the physical object.
- Hardware licenses specify where/how the license is identified on the physical item.



Summary slide



- We want to **share our work** so our research can have the greatest impact!
- Sharing on GitHub enables the use of a well tested mechanism to receive the input of others
 - More contributions been **more robust** and potentially more impactful
 - Contributions are **attributed and version controlled**
- When we share, we should always apply a license so we can control **how our outputs are used on by whom**.
 - Licenses can be **more or less restrictive**
 - Software documentation and hardware all have specific license types which **legal terms appropriate** to different outputs





Part 3: Using GitHub

- For some repos you might have a ‘local’ copy on your computer and a ‘remote’ copy on GitHub
- Personally, I have a local copy where I want to do more complicated editing

Pushing your local changes to your remote

- Directory must first be initialised for git (ready to be tracked)

```
> git status
```

```
> git add [filename]
```

```
> git commit -m "[describe the change]"
```

```
> git push
```

```
gitgood -- zsh -- 111x38
(base) cassandragouldvanpraag@Mac gitgood % git status
On branch main
Your branch is up to date with 'origin/main'.

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:   CODE_OF_CONDUCT.md
    modified:   README.md
    modified:   workshops/202602_OSHO/list-of-links.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    tmp.md
    workshops/202602_OSHO/202603_GitGood_OSHO.pptx
    workshops/202602_OSHO/gitgood-new-new.pptx
    workshops/202602_OSHO/gitgood-new.pptx
    workshops/202602_OSHO/gitgood.pptx

no changes added to commit (use "git add" and/or "git commit -a")
(base) cassandragouldvanpraag@Mac gitgood % git add README.md
(base) cassandragouldvanpraag@Mac gitgood % git commit -m "update repo checklist"
[main 48f9fa9] update repo checklist
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) cassandragouldvanpraag@Mac gitgood % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 10 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:the-turing-way/gitgood.git
  20aed3f..48f9fa9  main -> main
(base) cassandragouldvanpraag@Mac gitgood %
```



Editing online

Making online changes to your remote

1. Find the file
2. <Edit this file>
3. Make the change
4. <Commit changes>

The screenshot shows a GitHub commit dialog. At the top, it displays a commit hash: 8d415dc · 13 hours ago. Below this is a "History" button. A large orange circle highlights the "Edit this file" button, which is located in the toolbar below the commit hash. The toolbar also includes "Raw", "Copy", "Download", and "Edit" (circled in orange). Further down, there are "Cancel changes" and a green "Commit changes..." button. The main area of the dialog is titled "Commit changes" and contains fields for "Commit message" (with "demo remote edit" typed in) and "Extended description" (with a placeholder "Add an optional extended description..."). At the bottom, there are two radio buttons: one selected for "Commit directly to the main branch" and one for "Create a new branch for this commit and start a pull request". A link "Learn more about pull requests" is provided. Finally, there are "Cancel" and "Commit changes" buttons at the bottom right.

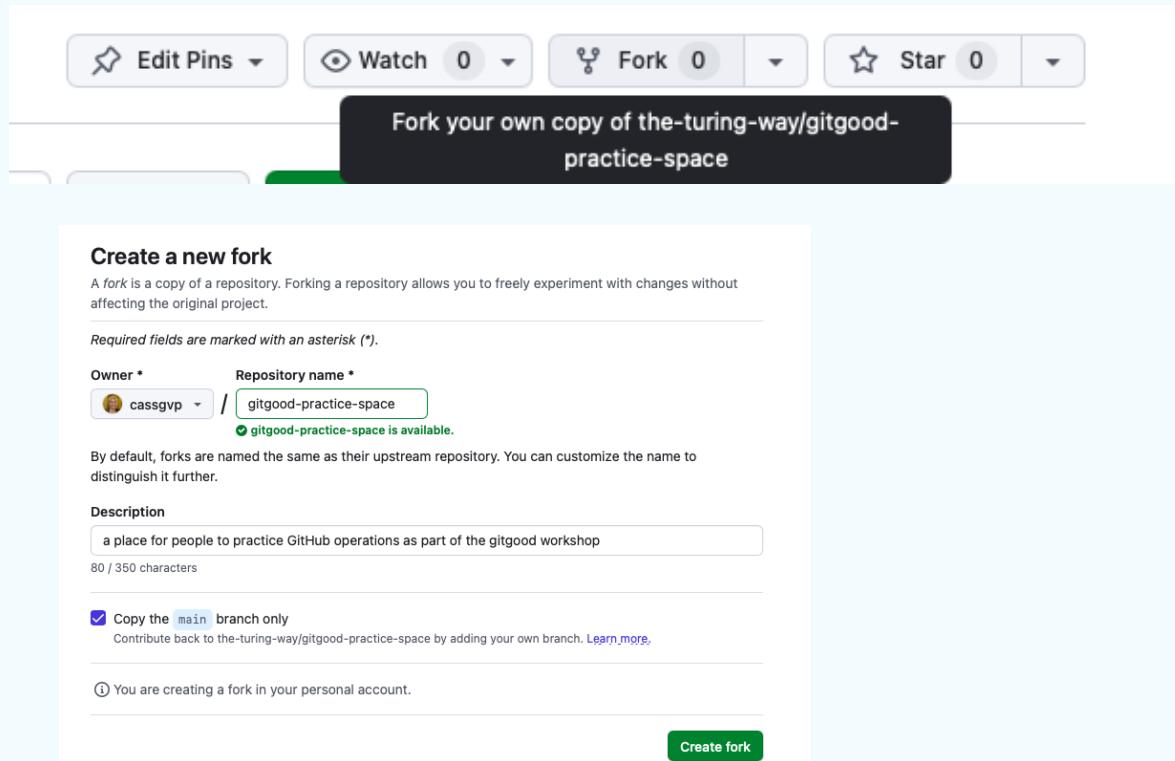
Collaborating on someone else's repo

1. Make a copy of their repo (**fork**)
 2. Make your changes in *your* online copy (**commit**)
 3. Share your changes with the repo owner (**pull request**)
 4. Changes are incorporated in the original repo (**merge**)
-
- Additional step of making your own copy of the repo preserves the fidelity of the original and give you space to play!
 - We're going to practice this online only!



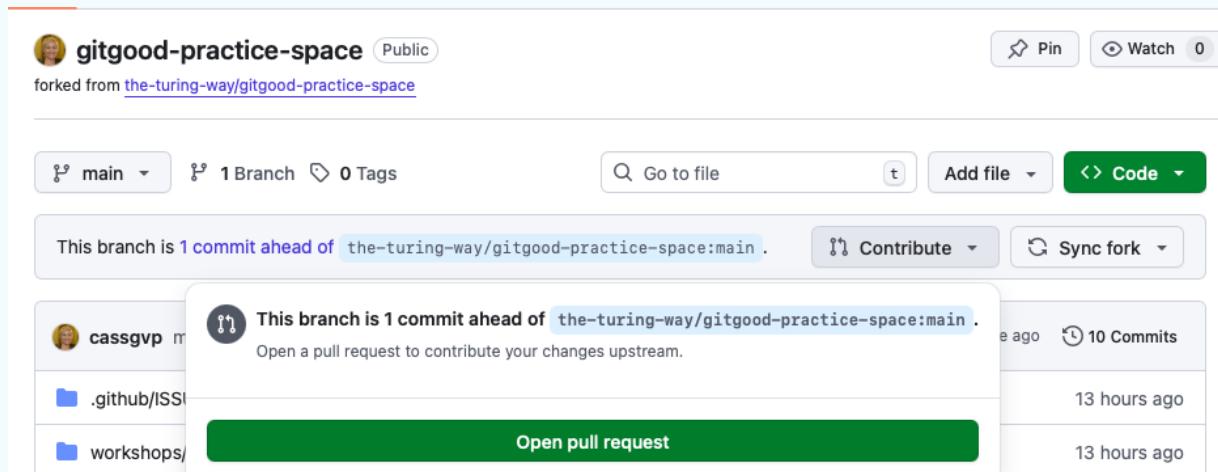
Collaborating on someone else's repo

1. Make a copy of their repo (**fork**)



A screenshot of a GitHub repository page showing the 'Fork' button highlighted. A modal window titled 'Create a new fork' is open, prompting the user to enter a repository name ('gitgood-practice-space') which is available. Other fields include 'Owner' (cassgvp), 'Repository name' (gitgood-practice-space), 'Description' (a place for people to practice GitHub operations as part of the gitgood workshop), and a checked 'Copy the main branch only' option. A note indicates that forks are named the same as upstream. A success message at the bottom says 'You are creating a fork in your personal account.' A green 'Create fork' button is at the bottom right.

2. Make your changes in *your* online copy (**commit**)
3. Share your changes with the repo owner (**pull request**)



A screenshot of a GitHub repository page for 'gitgood-practice-space'. It shows a commit from 'cassgvp' that is 1 commit ahead of the upstream 'main' branch. A callout box highlights this commit with the message: 'This branch is 1 commit ahead of the-turing-way/gitgood-practice-space:main. Open a pull request to contribute your changes upstream.' A large green 'Open pull request' button is visible. The repository has 1 branch and 0 tags. The commit was made 13 hours ago.



Collaborating on someone else's repo

4. Changes are reviewed by repo original repo owner (**merge**)

Commit message

Merge pull request #4 from cassgvp/main

Extended description

make a small change in my copy

This commit will be authored by 43407869+cassgvp@users.noreply.github.com.

Confirm merge **Cancel**



Exercise 2: Pull request

We're all going to submit a pull request to the practice repo!

1. Choose a line to edit from the framapad
2. Make a copy of the practice repo (**fork**)
3. Add your name; date today; a link to your research profile or some of your work to exercise-2.md (**commit**)
4. Share your changes with the repo owner (**pull request**)
5. Emma will review and incorporate your changes (**merge**)
6. Use zoom reaction when your changes have been merged!

We'll look out for any accident merge conflicts to review together

Collaboration practice!





Discussion

- What are your fears / concerns about working on someone else's repo?
- How would you like people to interact with you / your repos? (CONTRIBUTING.md)
- **What are your questions?**

Hands /
Chat



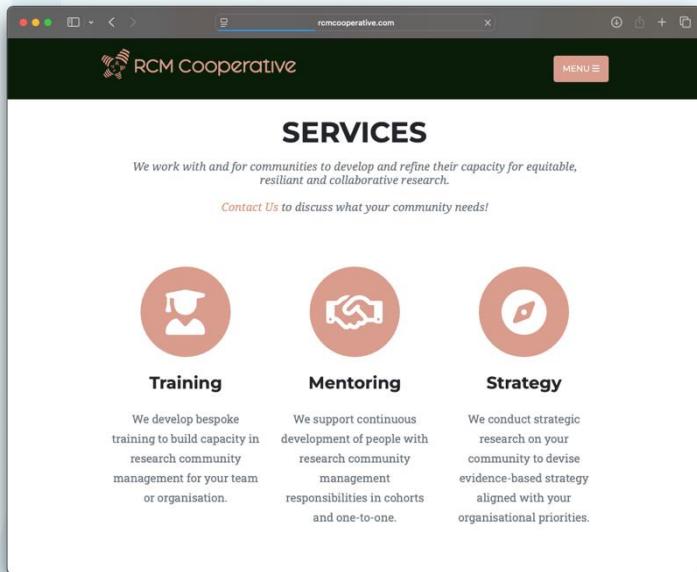
Part 4: What else can you do with GitHub

GitHub Pages (github.io)

The screenshot shows the homepage of jekyllthemes.io. At the top, there's a navigation bar with links for 'Browse Themes', 'Blog', 'Resources', and 'About'. A large banner on the left features the 'Jekyll THEMES' logo. The main heading is 'Free Jekyll themes'. Below it, a text block says: 'There are thousands of free themes for Jekyll created by the open source community. Here you'll find **only the very best ones**, hand-picked for their design and development quality.' A section titled 'Featured themes' displays several theme cards, each with a preview image, title, and a 'Get Started' button. One card for 'Minimal Mistakes' highlights it as a 'flexible two-column Jekyll theme' for personal sites, blogs, and portfolios. Another card for 'Just the Docs' focuses on documentation. Other cards include 'Hi, my name is Forty' (a responsive site template), 'Massa Libero' (a minimalist theme), and 'About All' (a theme for documentation).

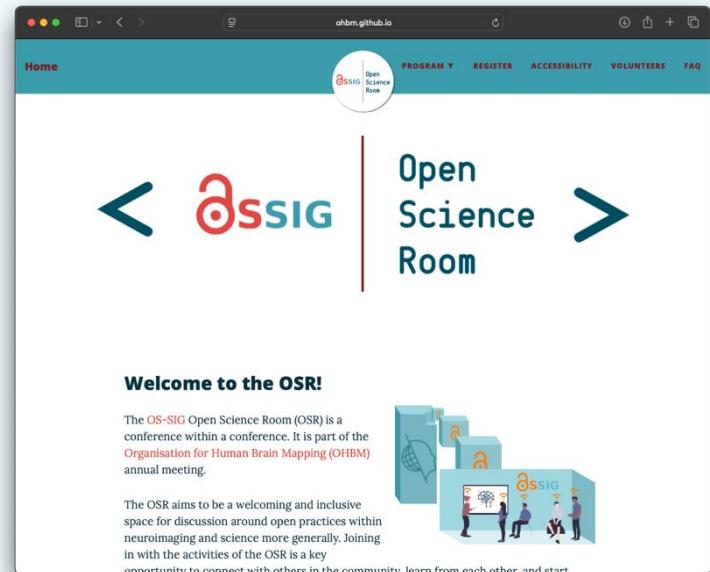


GitHub Pages

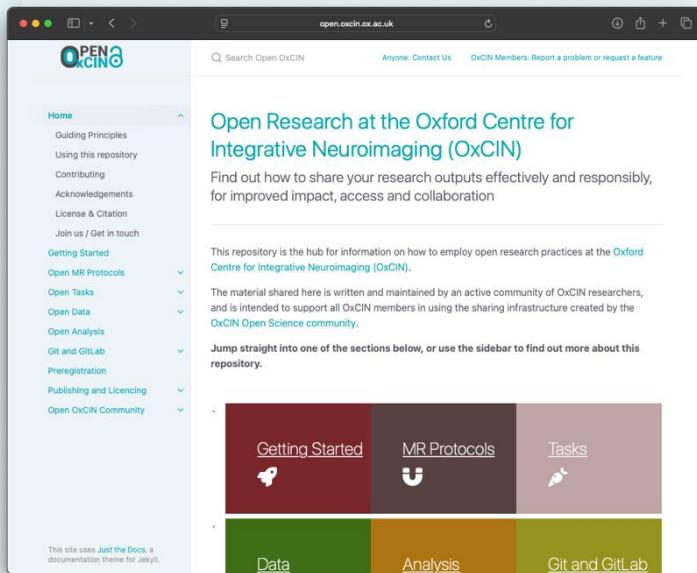


The screenshot shows the RCM Cooperative GitHub Pages site. The header includes the RCM Cooperative logo and a "MENU" button. Below the header, there's a section titled "SERVICES" with a sub-section "We work with and for communities to develop and refine their capacity for equitable, resilient and collaborative research." A call-to-action "Contact Us to discuss what your community needs!" is present. Three circular icons represent "Training" (person icon), "Mentoring" (handshake icon), and "Strategy" (compass icon). Below each icon is a brief description:

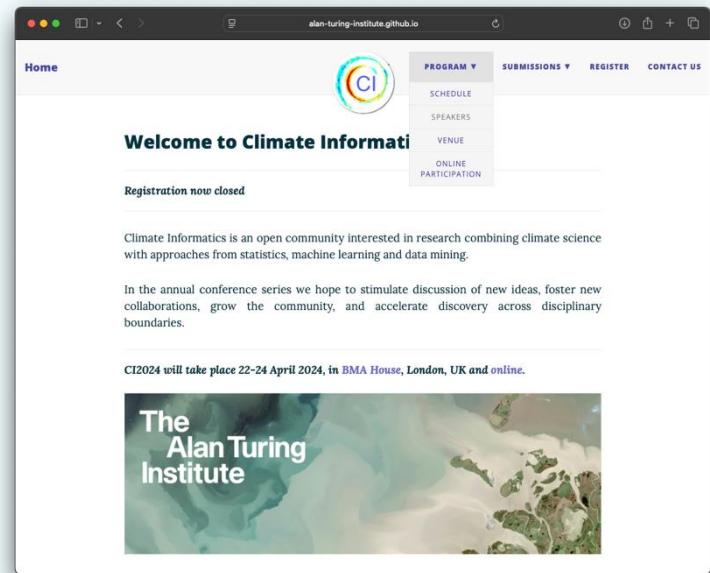
- Training**: We develop bespoke training to build capacity in research community management for your team or organisation.
- Mentoring**: We support continuous development of people with research community management responsibilities in cohorts and one-to-one.
- Strategy**: We conduct strategic research on your community to devise evidence-based strategy aligned with your organisational priorities.



The screenshot shows the OSSIG Open Science Room GitHub Pages site. The header includes the OSSIG logo and links for "PROGRAM", "REGISTER", "ACCESSIBILITY", "VOLUNTEERS", and "FAQ". The main title is "Open Science Room" flanked by '<' and '>'. Below the title, there's a "Welcome to the OSR!" section. It describes the OS-SIG Open Science Room (OSR) as a conference within the OHBMR annual meeting, aiming to be a welcoming and inclusive space for discussion around open practices within neuroimaging and science more generally. It highlights the opportunity to connect with others in the community, learn from each other, and start with the activities of the OSR.



The screenshot shows the Open Research at the Oxford Centre for Integrative Neuroimaging (OxCIN) GitHub Pages site. The header includes the OxCIN logo and a search bar. The main title is "Open Research at the Oxford Centre for Integrative Neuroimaging (OxCIN)". Below the title, there's a sub-section "Find out how to share your research outputs effectively and responsibly, for improved impact, access and collaboration". The sidebar contains links for "Home", "Guiding Principles", "Using this repository", "Contributing", "Acknowledgements", "License & Citation", "Join us / Get in touch", "Getting Started", "Open MR Protocols", "Open Tasks", "Open Data", "Open Analysis", "Git and GitLab", "Preregistration", "Publishing and Licensing", and "Open OxCIN Community". The main content area includes sections for "Getting Started", "MR Protocols", "Tasks", "Data", "Analysis", and "Git and GitLab".



The screenshot shows the Climate Informatics GitHub Pages site. The header includes the CI logo and links for "PROGRAM", "SUBMISSIONS", "REGISTER", and "CONTACT US". The main title is "Welcome to Climate Informatics". Below the title, it says "Registration now closed". The page describes Climate Informatics as an open community interested in research combining climate science with approaches from statistics, machine learning and data mining. It states that in the annual conference series we hope to stimulate discussion of new ideas, foster new collaborations, grow the community, and accelerate discovery across disciplinary boundaries. It mentions that CI2024 will take place 22-24 April 2024, in BMA House, London, UK and online. The background features a green and blue abstract design with the text "The Alan Turing Institute".

- Collaborative, version controlled, reproducible!



GitHub Pages

1. Enable Pages

The screenshot shows the GitHub repository settings page for 'gitgood-practice-space'. The 'Settings' tab is highlighted with an orange circle. On the left, a sidebar lists repository options like General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Requirements, Codespaces, and Pages. The 'Codespaces' and 'Pages' items are also highlighted with an orange circle. The main content area shows the 'General' settings, including the repository name 'gitgood-practice-space' and a preview of the 'main' branch.

2. Choose a branch

Build and deployment

Source

Deploy from a branch ▾

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more about configuring the publishing source for your site.](#)

None ▾

Save

Select branch



main

t access to your GitHub Pages site by publishing it privately. You can I documentation or knowledge base with members of your enterprise. s. [Learn more about the visibility of your GitHub Pages site.](#)

3. Apply a (remote) theme

Branch

Your GitHub Pages site is currently being built from the `main` branch. [Learn more about configuring the publishing source for your site.](#)

main ▾

/ (root) ▾

Save

Learn how to [add a Jekyll theme](#) to your site.



Project Management

This screenshot shows a GitHub project board for 'Book Dash Planning'. The board has a timeline view from February 2024 to August 2024. There are several columns representing different planning cycles. A specific task, 'Proposed First Planning Committee Meeting', is highlighted in yellow, indicating it is open or due soon. Other tasks include 'Second Planning Committee Meeting', 'Third Planning Committee Meeting', 'Fourth Planning Committee Meeting', 'Fifth Planning Committee Meeting', and 'Planning Committee Feedback & Wash Up'.

This screenshot shows a GitHub project board for 'RCM Coop Surplus'. It lists various funding proposals and their status. A task, 'Awaiting confirmation', is highlighted in yellow, indicating it is currently awaiting submission or review. Other tasks include '[FUNDING] DisCourse Flexible Fund #23', '[CONSULTING PROPOSAL] FORTAL BioFAIR Pathfinder #33', '[FUNDING] SSI Fellowship (Case) #24', '[CONTRACT] Open Science Residency Program (OSRP) #19', '[FUNDING] UKTRE DARE UK Community Groups #11', '[CONTRACT] ADR UK Community Leaders Workshops #20', '[FUNDING APPLICATION] SSI Further development fund #2', and '[CONSULTING PROPOSAL] Understanding Patient Data: health data PIPE commun... #35'.

This screenshot shows a GitHub project board for 'Infrastructure Working Group'. The board uses a Kanban-style interface with four columns: 'Ready to Work', 'In progress', 'In review', and 'Blocked'. Tasks are color-coded by priority: P0 (red), P1 (blue), and P2 (yellow). The 'Ready to Work' column contains tasks like 'Convert implicit references to explicit references' and 'Jupyter Book 2 migration'. The 'In progress' column contains tasks like 'Deploy Portuguese book' and 'Decalrative team management'. The 'In review' column contains tasks like 'Add dependabot config for pip packages'. The 'Blocked' column contains tasks like '[BUG]: Bibliography not working' and 'Finding out if there is missing alt text in TTW book'.



Discussion

- What are you going to try next with GitHub?
- What are your questions?
- Please fill in the feedback form now 😊

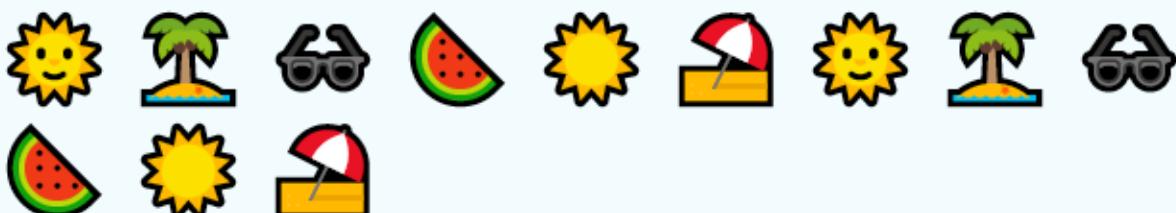
Hands /
Chat



Summary slide



- We want to **share our work** so our research can have the greatest impact!
- Sharing on GitHub enables the use of a well tested **mechanism to collaborate with attribution**
- We know why and how to **license our materials**
- We know how to **submit and receive contributions**
- We know that we can **publish collaborative webpages** with GitHub
- We know that GitHub can be used to **manage collaborative projects**





Closing

- Thanks to TTW GitGood contributors and TTW infrastructure team!
- Thanks to OSHO for bringing us together!
- Thank you for engaging!
- Book time with Cass to discuss anything related to this workshop, open science / reproducibility, collaboration / research community management!

