

Session 2 - Collaboration with GitHub



Emma Karoune and Celine Kerfant

On behalf of the ICOPS Committee

Adapted from: Esther Plomp's Github Intro workshop

References: [Mozilla Science Lab's Study Group Orientation](#),

[Friendly GitHub Intro by Kirstie Whitaker](#) &

Visual description: <https://learngitbranching.js.org/>

The Turing Way's ['Getting started with Github'](#)

Other resources:

Developing Collaborative Documents [Click here for the paired online materials](#)

by Malvika Sharan

License: CC-BY 4.0

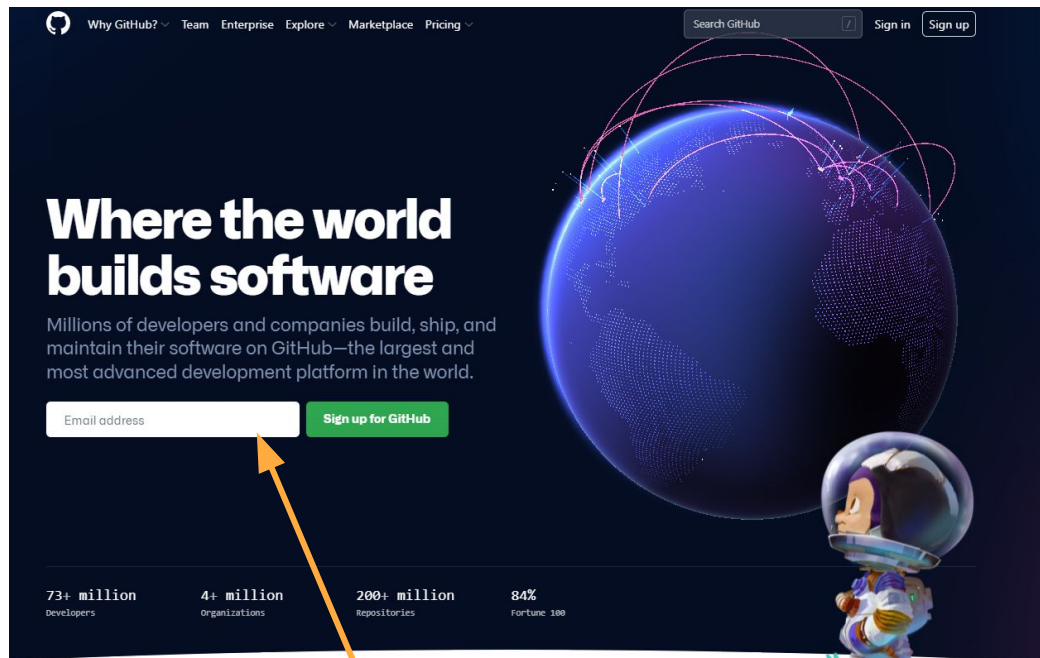




Go to github.com

Step 1: Sign-up

Step 2: Log in



Enter your email address here
and click green button

Learning outcomes - Session 2

Collaboration with Github

- To know what Version control is.
- To start using a collaborative workflow in Github.

What we learnt in session 1:

- To list some examples of use of Github.
- To be able to navigate the Github interface.
- To set up a personal repository.
- To start using markdown script.

Do you have any questions
from session 1?



Learning outcomes - Further sessions

Session 3 - *Further uses with Github*

- To know how to set up project management boards.
- To set up a simple Github webpage.

We won't be covering:

- Git through the command line - see for example [Version Control with Git](#) by the Carpentries
- Using a programming language with Git - see [slide 105](#) for R examples



Schedule

0:00 - 0:15 Log in and intro to session

0:15 - 0:30 Intro to version control

0:30 - 0:45 Forking

0:45 - 1:00 Forking exercise

1:00 - 1:20 Collaboration and branching

1:20 - 1:45 Branching exercise

1:45 - 2:00 Questions and wrap up



Code of Conduct

Yes / Encouraged

- Show empathy and kindness toward others
- Be respectful of differing opinions, viewpoints, experiences and **technological choices**
- Give and gracefully accepting constructive feedback
- Take responsibility for mistakes and any impact on others, learn from the experience
- Taking breaks to recharge!



Code of Conduct

Please contact about any violations:

- Emma Karoune (ekaroune@googlemail.com)

No harassment

- Verbal and text comments that reinforce social structures of domination related to gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, age, religion or work experience.
- Use of sexual or discriminatory imagery, comments, or jokes
- Deliberate intimidation, disruption
- Unwelcome sexual attention
- Advocating for, or encouraging, any of the above behaviour

If the Code of Conduct is violated you will be asked to stop or leave the space.



Collaboration and version control





Collaborative Documents

What are the **challenges** in developing them?



When we work with many people to collaboratively develop a document.

Challenges: asynchronous and conflicting edits, location, time zones, VERSIONS

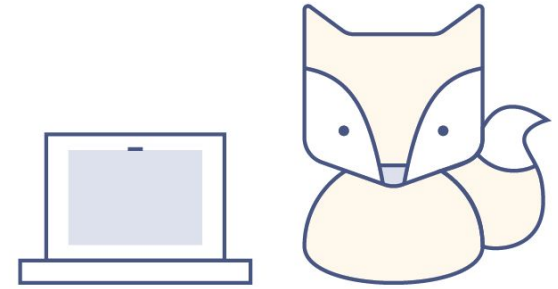
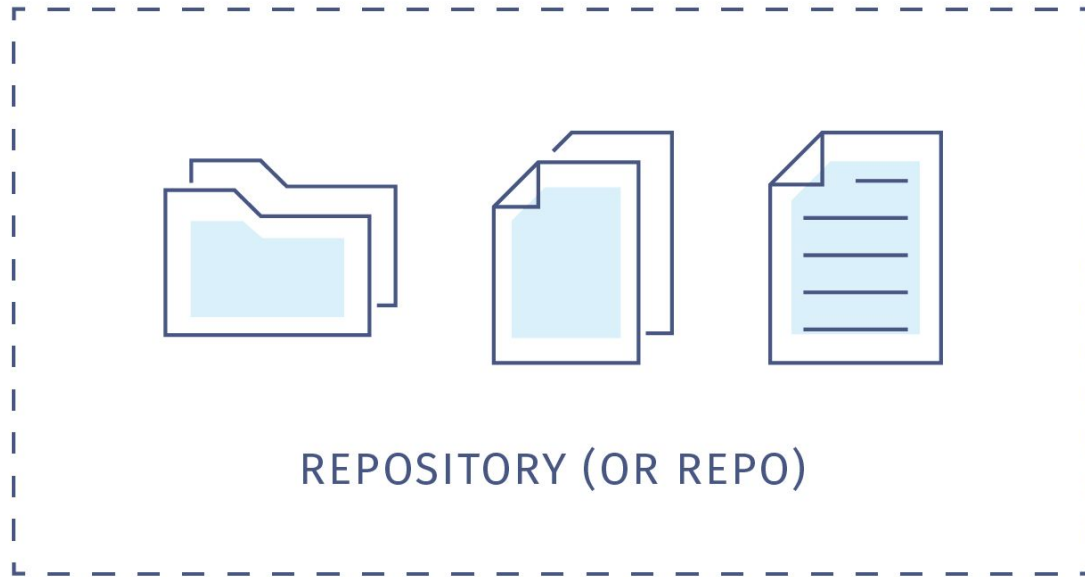




Image from Mozilla Science Lab:

<https://mozillascience.github.io/study-group-orientation/3.1-collab-vers-github.html>



Revisions and Versions

Image from Mozilla Science Lab:

<https://mozillascience.github.io/study-group-orientation/3.1-collab-vers-github.html>

Version Control

Management of **changes**, called **revisions**, to a file

- Simple file versioning with v1.0, v1.1, ...
- Simple tools like Google Drive, Dropbox...
- Advanced tools: git (local version controls system)



Revision: *Change associated with a **timestamp** and details like what changes were made, who changed them, why etc.*



Version Control

Benefits: collaborate in real time, store history, go back to previous versions



Add information on changes

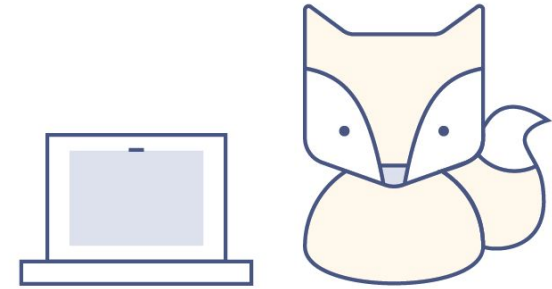
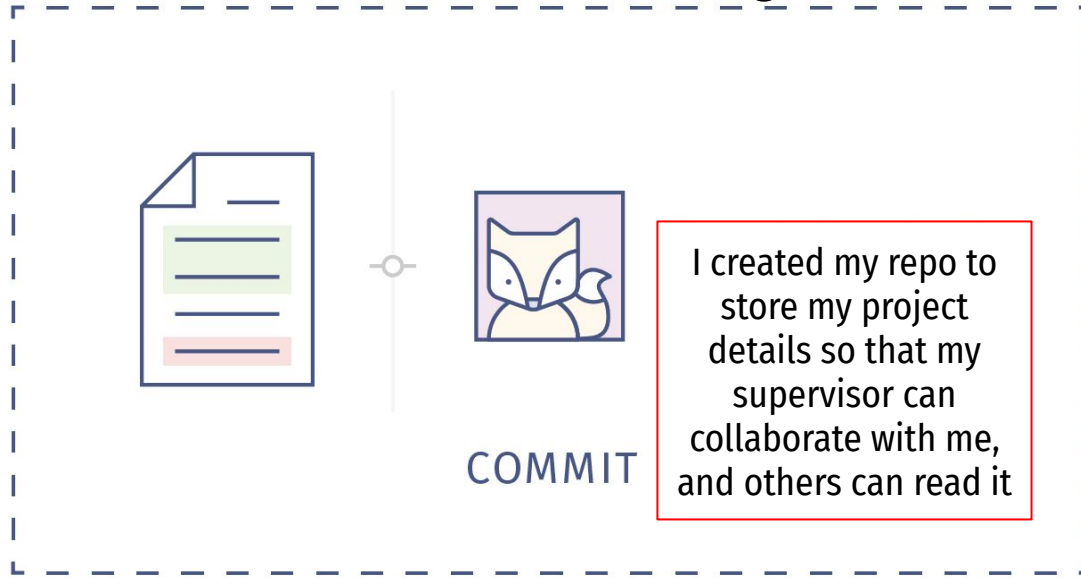


Image from Mozilla Science Lab:

<https://mozillascience.github.io/study-group-orientation/3.1-collab-vers-github.html>

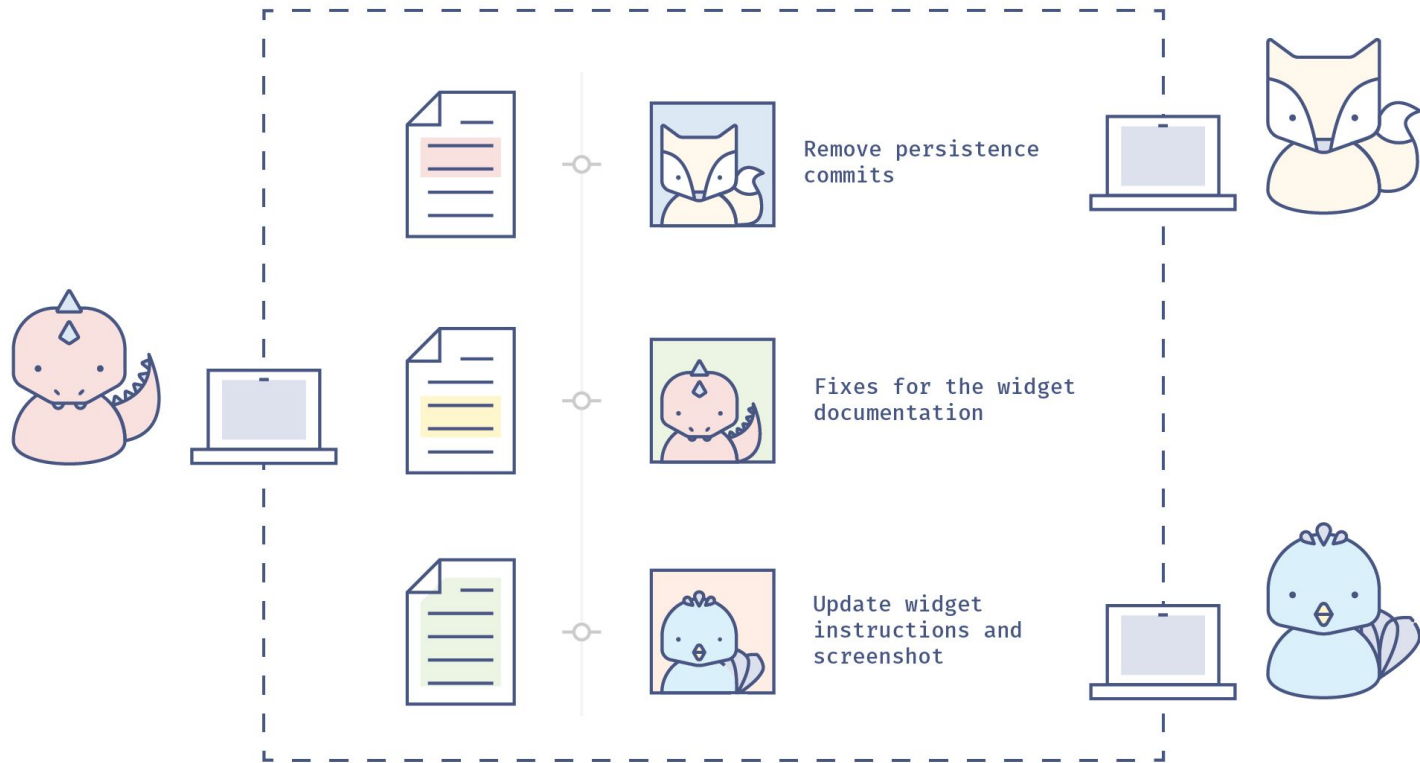


Image from Mozilla Science Lab:





<https://mozillascience.github.io/study-group-orientation/3.1-collab-vers-github.html>



GitHub

- **Hosts your repositories** (projects) online
- Helps you **work with contributors**/collaborators
- Provides **web interface** for version control
- It can be used for **project management** and communication
- Useful when a group of people are working together

GitHub Terms

- Commit 
- Branch & Fork 
- Pull request 
- Merge 



Create & manage
“Issues”

Pull Request to add your
changes into forked repo

Fork this repo

Clone to your
computer from the web
using GitHub Desktop
or git command

This material is used for teaching
non-researchers about versioning and
maintaining Open Access Documents
to invite collaboration.

malvikasharan.github.io/developing...

Readme

Download Zip to
download the zipped
repo onto your
computer

on 20 May 2020

Packages

The screenshot shows the GitHub repository interface for 'malvikasharan / developing_collaborative_document'. The repository has 2 watchers, 4 stars, and 11 forks. The 'Issues' tab is selected, showing 1 issue. The 'main' branch is the default, with 1 branch and 1 tag. The repository contains several files and folders, including '.github/ISSUE_TEMPLATE', 'lessons', '2021-03-10-attendees.md', 'CODE_OF_CONDUCT.md', 'CONTRIBUTING.md', 'LICENSE', 'README.md', and '_config.yml'. A 'Pull request #11' from 'msundukova/patch-2' is open. The 'Clone' button is highlighted, showing options for HTTPS, SSH, and GitHub CLI. The 'Download ZIP' button is also highlighted. The 'README.md' file is selected, showing its content. The repository is licensed under 'CC-BY' and has a DOI of 10.5281/zenodo.3835657.

malvikasharan / developing_collaborative_document

Unwatch 2 Star 4 Fork 11

<> Code Issues 1 Pull requests Actions Projects 1 Wiki Security Insights

main 1 branch 1 tag

malvikasharan Merge pull request #11 from msundukova/patch-2 ...

File/Folder	Description	Last Updated
.github/ISSUE_TEMPLATE	Update issue templates	
lessons	Update 7-contributing-to-other-repo.	
2021-03-10-attendees.md	Merge branch 'master' into patch-2	
CODE_OF_CONDUCT.md	add code of conduct	
CONTRIBUTING.md	Create CONTRIBUTING.md	13 months ago
LICENSE	add cc-by license	13 months ago
README.md	Update README.md	2 months ago
_config.yml	Update _config.yml	2 months ago

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI

https://github.com/malvikasharan/developing_collaborative_document

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

13 months ago

13 months ago

2 months ago

2 months ago

README.md

DOI 10.5281/zenodo.3835657

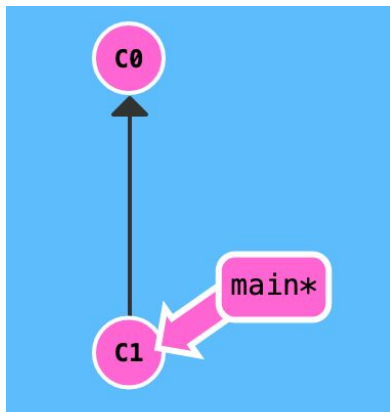
Your default **branch**
is “main” - but you
can switch or add
branches here

The “README.md”
is automatically
rendered when you
visit a repo

Commit 🔑

Saving a version of file(s)

- 'commit' changes with message (i.e. 'My first commit')



→ ↺ 🏠 <https://github.com/alan-turing-institute/the-turing-way/commits/>

🔑 Commits on Apr 26, 2021

Update contributors.md ...
👤 EstherPlomp committed 16 days ago ✓

🔑 Commits on Apr 24, 2021

Update hyperlink so it renders in Netlify build (#1823)
👤 inwaves committed 18 days ago ✓

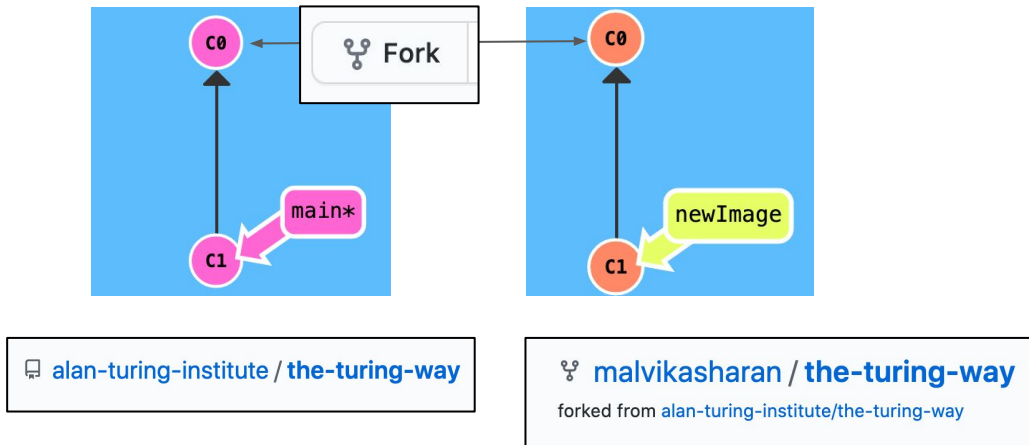
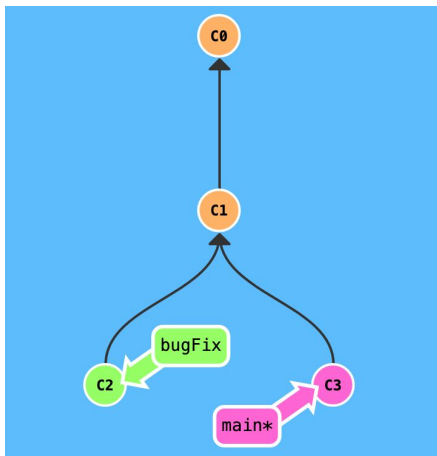
Improve grammar in compendia.md (#1822) ...
👤 inwaves and malvikasharan committed 18 days ago ✓

🔑 Commits on Apr 22, 2021

Update project_management/turing_mybinder_role.md
👤 sgibson91 committed 20 days ago ✓

Branch & Fork

- **Branch:** points to a specific commit in **your** repo, temporary places to work through a feature ← you have **write** access
- **Fork:** copy of repo that is an entirely **independent repo** ← no need of write access or intent to merge with the main



Branch & Fork

- **Branch:** points to a specific commit in **your** repo, temporary places to work through a feature ← you have **write** access

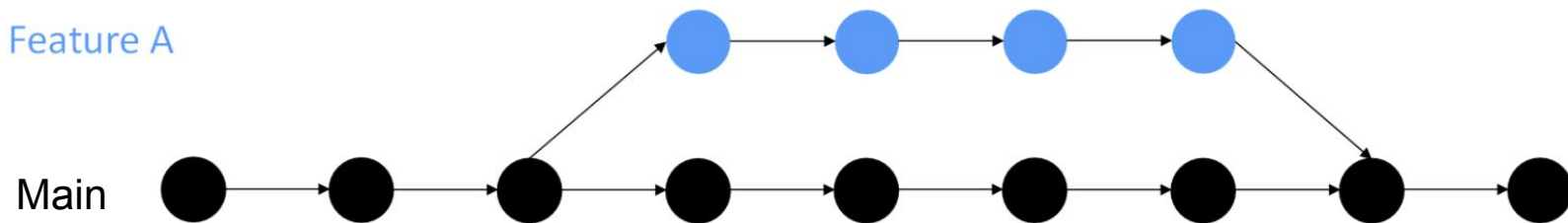
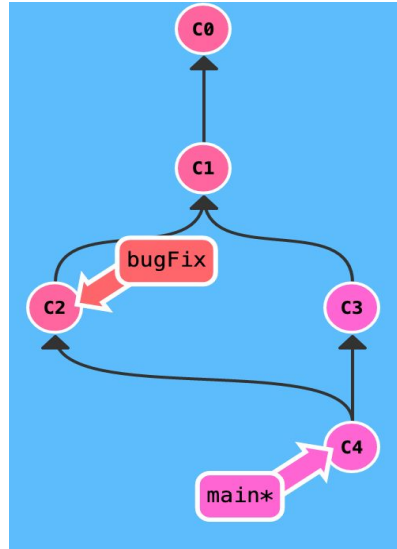
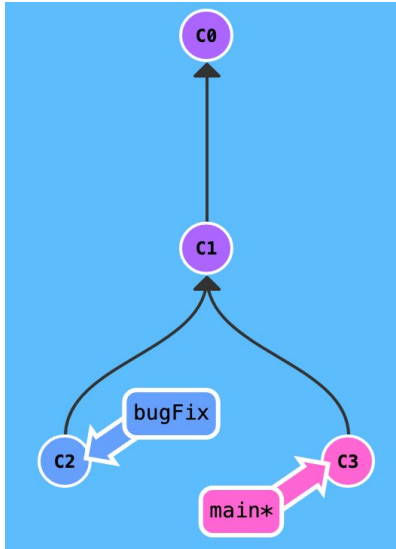


Fig. 18 An illustration of a development and main branch in git.

Pull Request (PR)

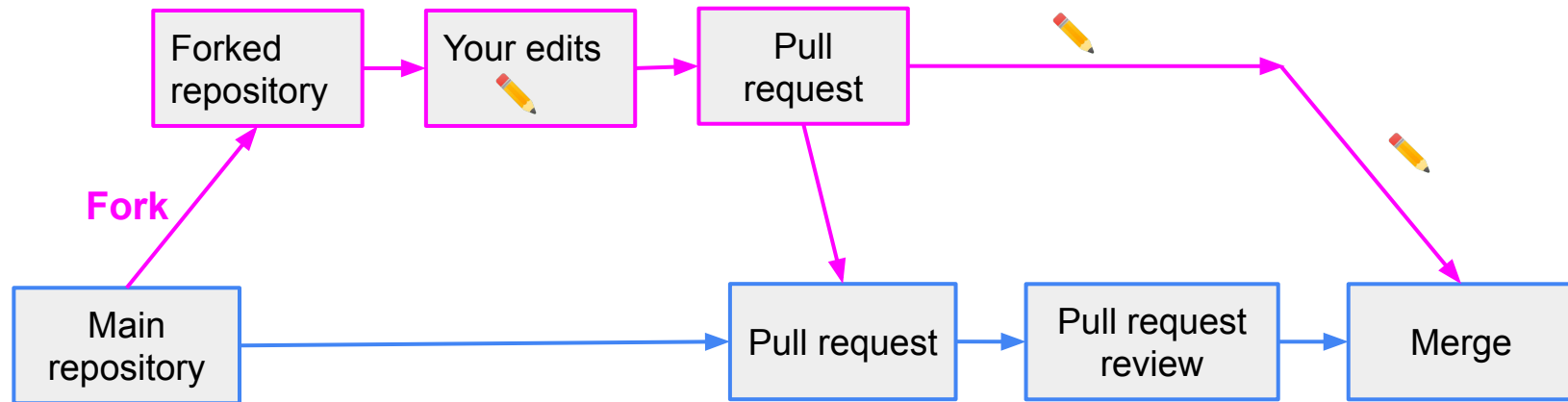
- **PR:** Requesting to combine the work from a forked repo or two branches to the main repo



Fork + Pull Request Workflow

If you don't own a repo and aren't an official collaborator:

Happening in your Github account



Happening in your the repo owners Github account



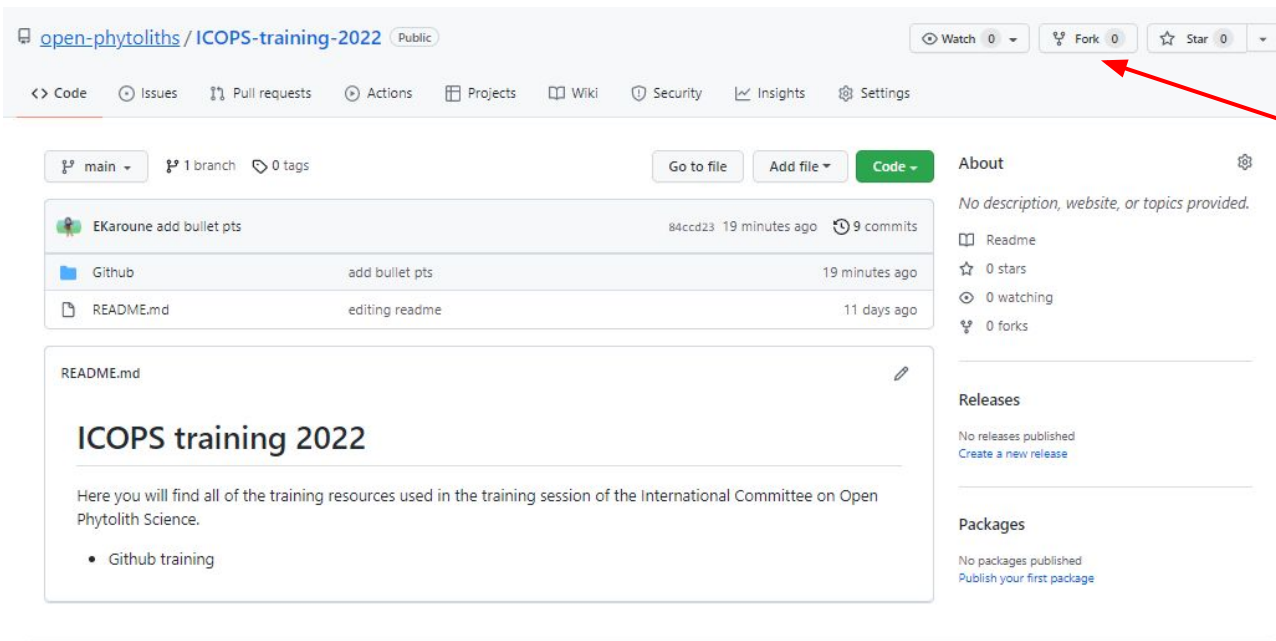
Fork + Pull Request Workflow

If you don't own a repo and aren't an official collaborator:

- a. You will **fork a repo**
- b. Work on your forked copy of the repo
- c. **Make a pull request** for the changes you've made
- d. Author/owner will decide to merge them in
- e. Make friends + party!



1. Find the repo you want to fork and fork it!



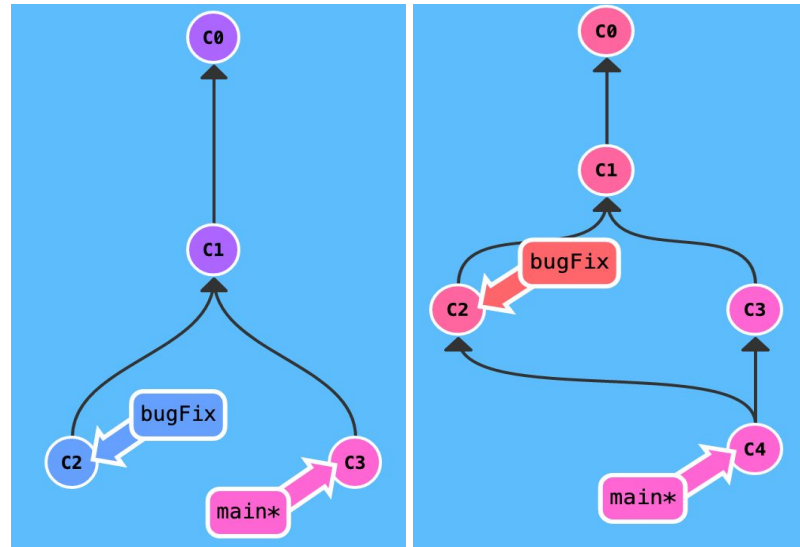
Press fork here!

Choose to fork in
your Github
account.

Link to repo for forking: <https://github.com/open-phytoliths/ICOPS-training-2022>

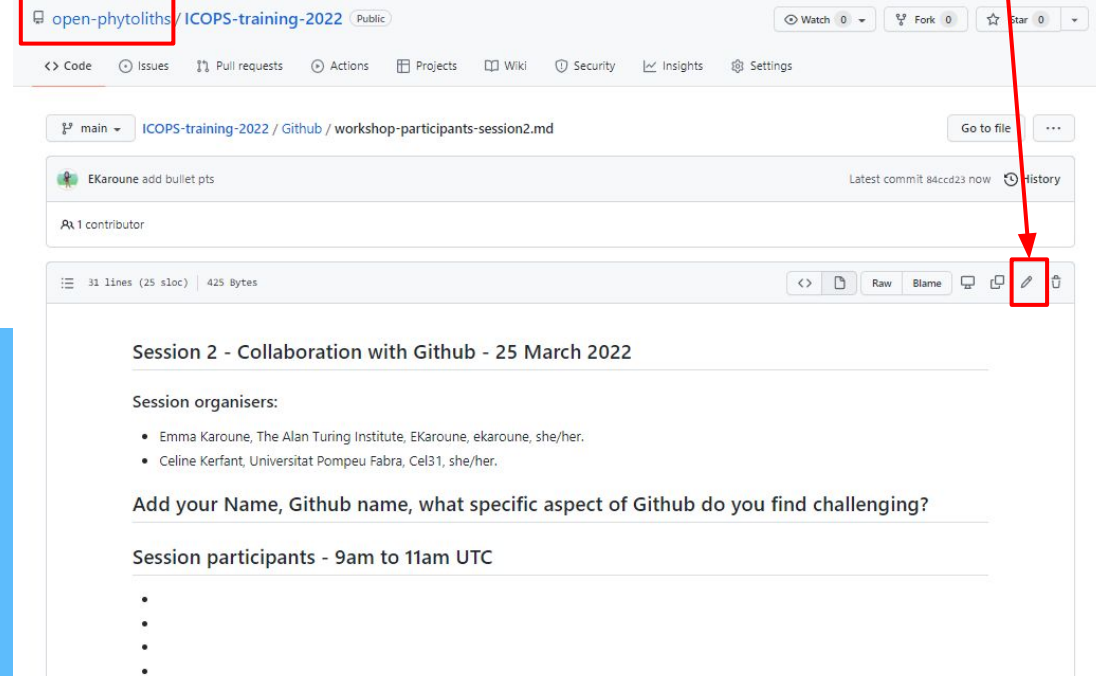
2. Work on forked repo - edit file

Edit by adding your details to
“*ICOPS-training-2022/Github/workshop-participants-session2.md*”



This will be your account here

Click pencil to edit



Remember to commit (save) once you have finished editing.

3. Start a Pull Request

- Click on pull requests at the top of the window and then green new pull request button.
- Give it a title and press create pull request.
- Now you will be brought to the pull request, now located in the owners repository.

The screenshot illustrates the GitHub interface for creating a pull request. At the top, the 'Pull requests' tab is highlighted with a red box, and a red arrow points to it from the first instruction. Below the navigation bar, the 'New pull request' button is highlighted with a red box, and a red arrow points to it from the first instruction. The 'Comparing changes' section shows the base repository as 'alan-turing-institute/the-turing-way' and the head repository as 'malvikasharan/the-turing-way'. The 'Open a pull request' section shows the title 'Create a test PR' and the 'Create pull request' button.

Code Pull requests Actions Projects Wiki Security Insights Settings

Filters is:pr is:open Labels 11 Milestones 0 New pull request

0 Open 2 Closed Author Label Projects Milestones Reviews Assignee Sort

Labels 11 Milestones 0 New pull request

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: alan-turing-institute/the-turi... base: master ← head repository: malvikasharan/the-turing-way compare: master

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](#).

base: master ← compare: master ✓ Able to merge. These branches can be automatically merged.

Create a test PR

Write Preview H B I ≡ < > @ ↶ ↷

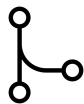
Summary

* Test PR for the workshop

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

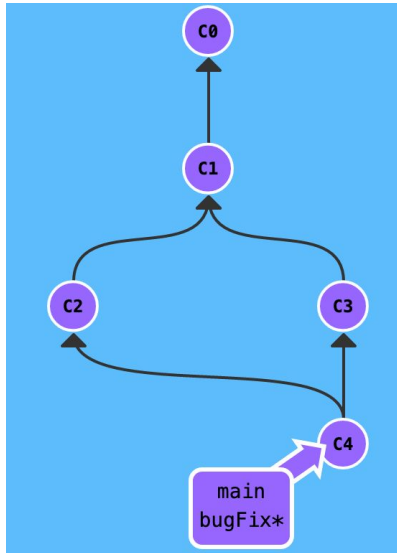
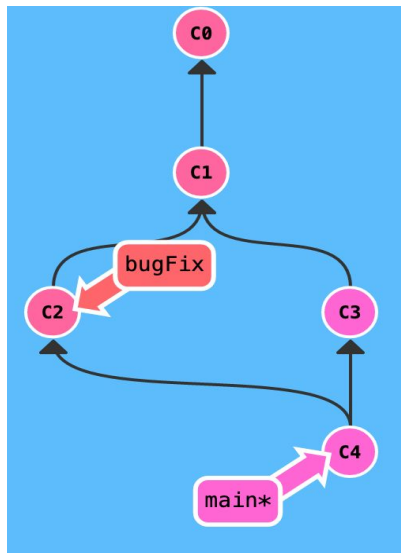
Allow edits and access to secrets by maintainers Create pull request

Merge



- **Merge:** combining the work from two different branches together after review

- Now wait!
- Only the owner of the repo or those that have write access can review and merge the pull request.



Create a test PR #1868

[Open](#) malvikasharan wants to merge 2 commits into `alan-turing-institute:master` from `malvikasharan:master`

Conversation 0 Commits 2 Checks 0 Files changed 1

malvikasharan commented 2 minutes ago

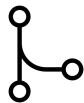
Member

Summary

- Test PR for the workshop

Conversation 1 Commits 2 Checks 13 Files changed 1

Merge



- **Merge:** combining the work from two different branches together after review

In a pull request (PR) you can see:

- who created it
- who has done the edits - could be more than one person.
- the changes made.
- who approves it.

Create a test PR #1868

malvikasharan wants to merge 2 commits into `alan-turing-institute:master` from `malvikasharan:master`

Conversation Commits Checks Files changed

malvikasharan commented 2 minutes ago Member

Summary

- Test PR for the workshop

Conversation Commits Checks Files changed

workshops/github-workshop/2021-05-13-intro-workshop.md

```
@@ -37,7 +37,7 @@ Name / Institute or affiliation / **What specific aspect of GitHub do you find c
37      * Kendra Phelps / EcoHealth Alliance /
38      * Mabrouk Boutagoug / University of Batna 1 /
39      * Maria Nanton / Buenos Aires' Ministry of Health /
40      * Malvika / The Alan Turing Institute / Testing this PR
41      *
42      *
43      *
```

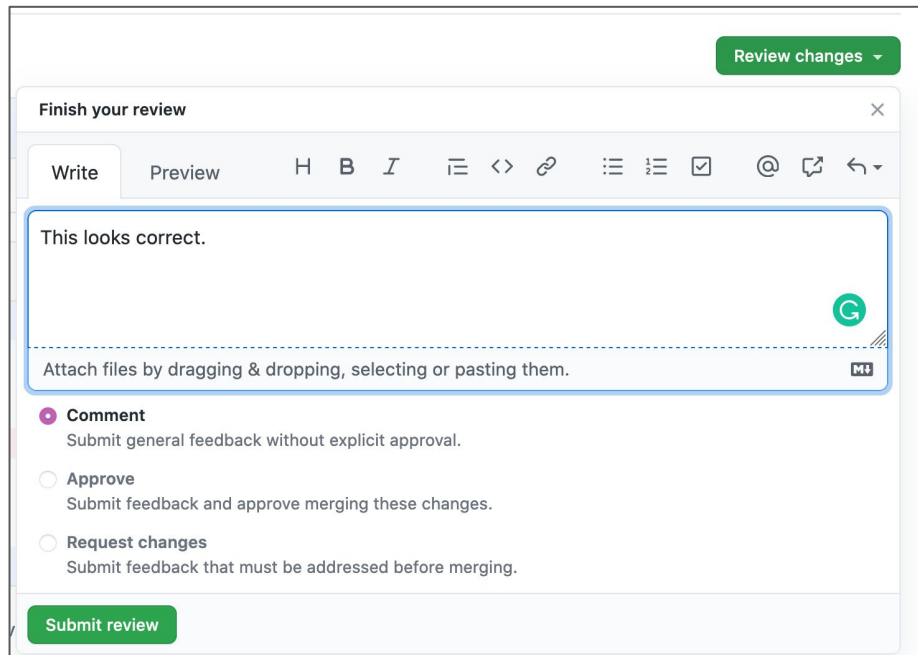
Review changes before merge

Users with **write access** can **approve** and merge PRs

A reviewer:

- Looks at the files changed.
- Makes suggested changes or edits the files themselves.
- They write comments in review changes and approve the merge.

- Comment - for general comments but not allowing merging.
- Approve - gives the go ahead to merge.
- Request changes - asks for changes to be made before approval will be given.



The screenshot shows a GitHub pull request review interface. At the top right is a green button labeled "Review changes". Below it is a "Finish your review" section with a close button (X). This section contains a "Write" tab and a "Preview" tab. The "Write" tab is active and shows a rich text editor with a toolbar containing icons for bold, italic, link, code, list, and other formatting options. The text area contains the comment "This looks correct." and a green circular icon with a white 'G' (GitHub logo) in the bottom right corner. Below the text area is a dashed line and the text "Attach files by dragging & dropping, selecting or pasting them." with a small icon. Below this is a list of three options: "Comment" (selected with a purple dot), "Approve" (radio button), and "Request changes" (radio button). Each option has a description: "Submit general feedback without explicit approval.", "Submit feedback and approve merging these changes.", and "Submit feedback that must be addressed before merging." respectively. At the bottom of the "Finish your review" section is a green button labeled "Submit review".

Now it is your go! Fork away!

1. Find the repository you want to fork and fork it!
 - a. Fork this repository: <https://github.com/open-phytoliths/ICOPS-training-2022>
2. Work on the forked repository
 - a. Edit the file, press commit.
3. Start a pull request
 - a. Click pull request at the top and then press the green new pull request button.
 - b. Write a title and press green create pull request button.
4. Merge
 - a. Now wait! The owner of the repo, or those that have write access, can do this - NOT you!
 - b. When merged you should be able to see your edits in the original repository.

What are issues for?


Issues are like posts/messages

- Report bugs and errors, propose a new idea, save notes or invite discussions
- Give a descriptive header, provide details
- Great for working as part of an asynchronous team












Issue: ✨ General

Have a different idea? If this doesn't look right, [choose a different type](#).



Write


Preview


H B I         

<!-- Please complete the following sections when you open an issue. You are encouraged to keep this top level comment box updated as you develop and respond to reviews. If you have write access to the repository please also assign the appropriate label (or labels) to your issue. Note that text within html comment tags will not be rendered. -->


Summary

<!-- Please provide a detailed description of the change or addition you are proposing, or the question you're asking. Please provide as much context as possible and link to related issues and/or pull requests. -->

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

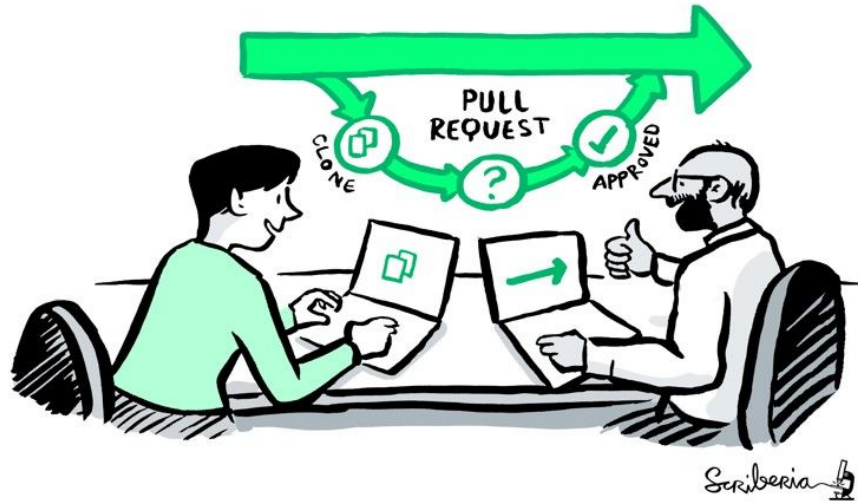
 Styling with Markdown is supported

Submit new issue

 Remember, contributions to this repository should follow its [contributing guidelines](#) and [code of conduct](#).

Issue vs Pull Request (PR)

- With a **issues**, you can **propose ideas for possible changes**, and invite discussions before you **make a PR** (or do something else).
- With a **pull request**, you can **propose changes**, discuss, and iterate on before you **merge** the changes into the project.



Using Issue & PR Features

- Different **contributors can see** what you are working on and offer help, share thoughts or learn from your work.
- Provide sufficient details, respond patiently, accept constructive feedback and **invite collaboration**.




Using Issue & PR Features

- **Advance features:** Use labels to sort or highlight relevant PRs and issues (“good first issue”) and tag people for comments/review

Onboarding Callum as Turing cluster admin #1856

[Open](#) sgibson91 opened this issue 9 days ago · 1 comment




sgibson91 commented 9 days ago · edited ▾

Member


⋮

Actions

- ✓ Open issue in <https://github.com/jupyterhub/team-compass> introducing Callum
 - Issue here [jupyterhub/team-compass#407](#)
- ✓ Add item to May's meeting agenda to introduce Callum
- ✓ Invite Callum to the May JupyterHub & Binder team meeting
- ✓ Callum to request access to the `turingmybinder` Azure subscription through TopDesk
- ✓ Sarah to schedule meeting with Callum to give a tour of the GitHub repositories and various tools/procedures
- ✓ Sarah to schedule sync-up meeting with Callum, Kirstie and Malvika re synergies between TPS/Turing Way/Binder (Jupyter more widely)





sgibson91 added community project management labels 9 days ago



sgibson91 assigned callummole and sgibson91 9 days ago

Assignees

 callummole

 sgibson91

Labels

community project management

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

Congratulations!



You now know:

Commit



Issue



Pull request

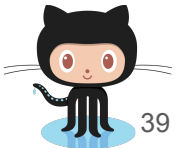


Merge



Branches and Pull Requests in your repository

How to work on
collaborative projects
with your team



Branch & Pull Requests



- **Branch**

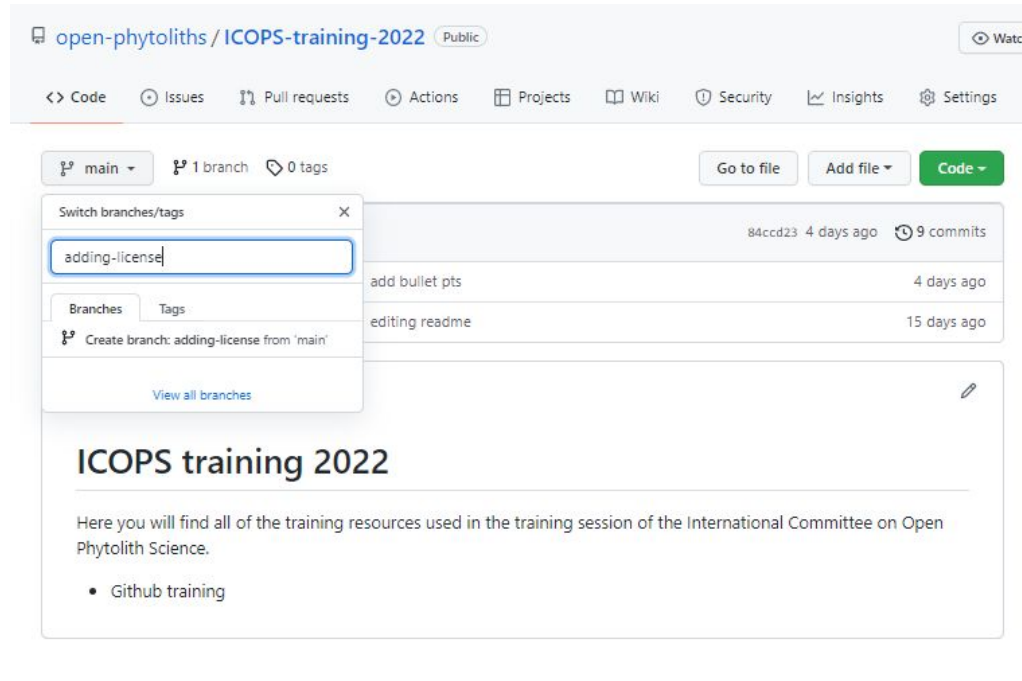
- pointers to a specific commit in **your** repo, temporary places to work through a feature ← you have **write** access

- **Pull request**

- request to add your changes from a branch back into master

- **Merge**

- act of incorporating new changes (commits) from one branch to another



Why work on a branch?

The screenshot displays a GitHub commit history for a branch. It is organized into three sections based on the date of the commits:

- Commits on Oct 6, 2021:**
 - A merge pull request #1 from EstherPlomp/EstherPlomp-patch-1, committed 4 days ago. The commit hash is 38cdaec.
 - An update to index.html, committed 4 days ago. The commit hash is 2d23ff3.
- Commits on Oct 5, 2021:**
 - An update to index.html, committed 5 days ago. The commit hash is e8056fb.
 - Another update to index.html, committed 5 days ago. The commit hash is e38210f.
- Commits on Oct 3, 2021:**
 - Update index.html, committed 7 days ago. The commit hash is 380ed9b.
 - Update index.html, committed 7 days ago. The commit hash is c3ca55f.
 - Update index.html, committed 7 days ago. The commit hash is 76ce4bc.
 - Update index.html, committed 7 days ago. The commit hash is 11c8779.
 - Update index.html, committed 7 days ago. The commit hash is fd437f9.
 - Update index.html, committed 7 days ago. The commit hash is d8a4842.

Each commit entry includes a green 'Verified' badge, a copy icon, the commit hash, and a diff icon. The commit messages are 'Merge pull request #1 from EstherPlomp/EstherPlomp-patch-1' and 'Update index.html'.

- This is your work in progress.
- Allows you to work with your team members on a task so you can comment and edit before adding to main.
- When you have a website, branch keep it off the live page.

Collaborative workflow for your collaborative project

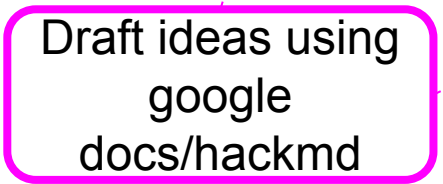
Main repo branch



Your branch

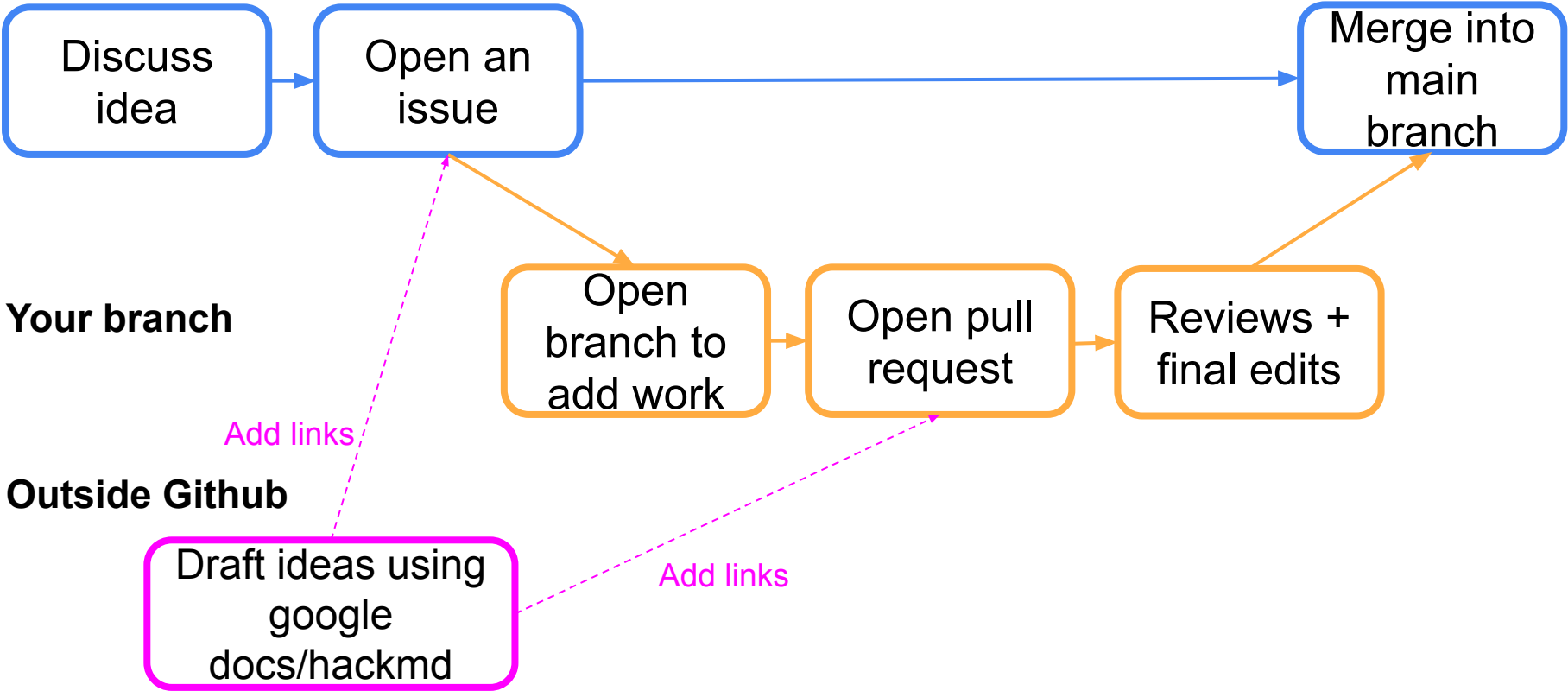


Outside Github



Add links

Add links



Branching exercise

1. Person 1 creates a repository named test-repo and adds your team as collaborators.
2. Person 2 starts a new branch called Readme and adds the README.md template.
3. Person 3 starts another branch called license and adds license file.
4. Person 1 starts another branch to called code-of-conduct to add a code of conduct file.
5. All do a pull request.
6. All practice reviewing a pull request and then merge into the main repository.



Work in groups of 3

**Main room
(English):**

**Room 1
(English):**

**Room 2
(Spanish):**

**Room 3
(Spanish):**

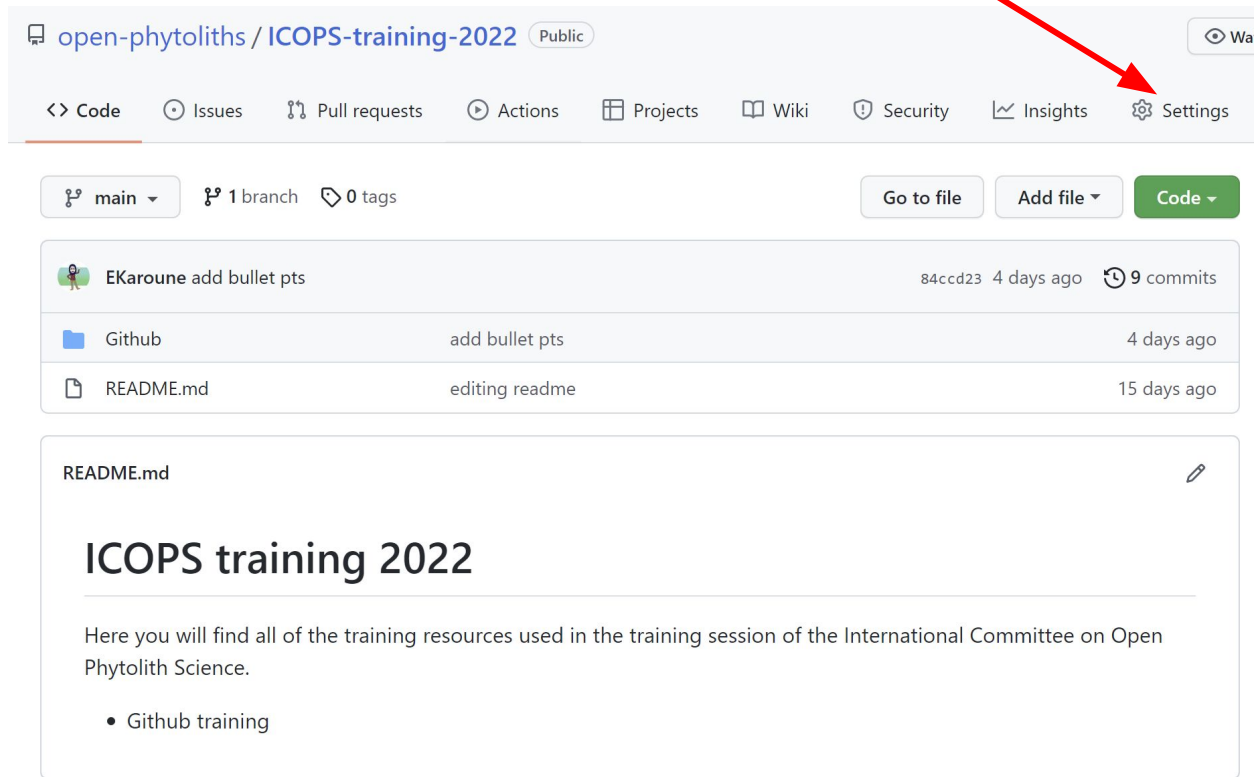


***Create your repository and add your
collaborators***

Add collaborators

1. Only the **owner** of a repo can:
 - a. **add** people as **collaborators**
 - b. each collaborator can read/write files in the repo
 - c. each collaborator is adding files and other content → making branches → and either directly merging changes in or via pull requests

Settings - add collaborators here



open-phytoliths / ICOPS-training-2022 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

EKaroune add bullet pts 84ccd23 4 days ago 9 commits

Folder	Github	add bullet pts	4 days ago
File	README.md	editing readme	15 days ago

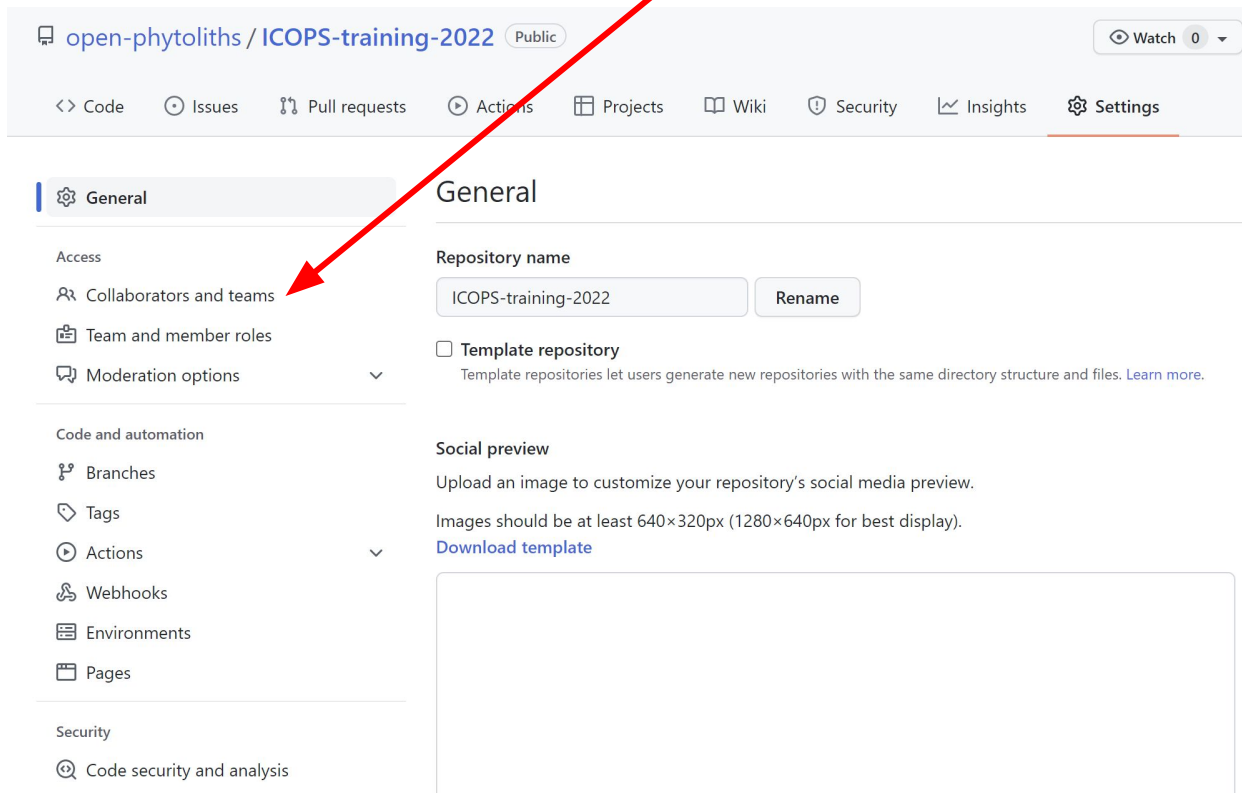
README.md

ICOPS training 2022

Here you will find all of the training resources used in the training session of the International Committee on Open Phytolith Science.

- Github training

Click here



open-phytoliths / ICOPS-training-2022 Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

- Collaborators and teams
- Team and member roles
- Moderation options

Code and automation

- Branches
- Tags
- Actions
- Webhooks
- Environments
- Pages

Security

- Code security and analysis

General

Repository name

ICOPS-training-2022 [Rename](#)

☐ **Template repository**
Template repositories let users generate new repositories with the same directory structure and files. [Learn more.](#)

Social preview
Upload an image to customize your repository's social media preview.
Images should be at least 640×320px (1280×640px for best display).
[Download template](#)

General

Access

Collaborators and teams

Team and member roles

Moderation options

Code and automation

Branches

Tags

Actions

Webhooks

Environments

Pages

Security

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

BASE ROLE

All 4 members can access this repository.

Read

DIRECT ACCESS

1 has access to this repository. 1 member.

[Create team](#)

Add people

Add teams

Manage access

☐ Select all

Type Role

Find a team, organization member or outside collaborator...

☐ Emma Karoune
EKaroune

Role: Admin

Click add people

Type or copy/paste your team members github account names

- they will receive an email invite to the repository

Add people to ICOPS-training-2022

Search by username, full name, or email

Select a member above

Now your team member can work on the repository

Look at the instructions on the shared doc, each person needs to:

- Start by making a new branch from the landing page
- Add a file
- Start a new pull request
- Then review each others pull request and merge into main.

Share out!

Main room: Room 1: Room 2: Room 3:



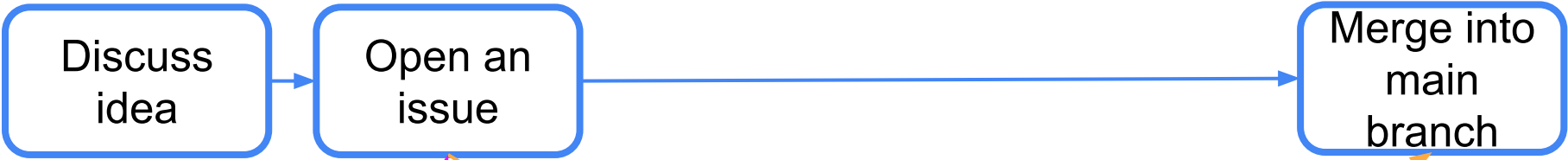
What we have learnt

- What version control is
- Collaborative workflow
 - fork
 - branch
 - pull request
 - merge
 - issue

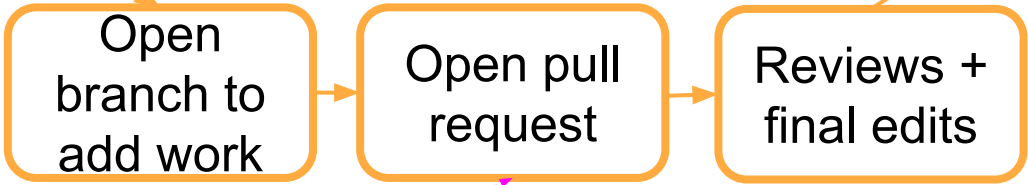
Any questions?

Collaborative workflow for your collaborative project

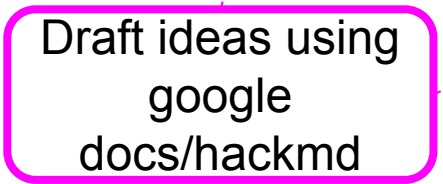
Main repo branch



Your branch

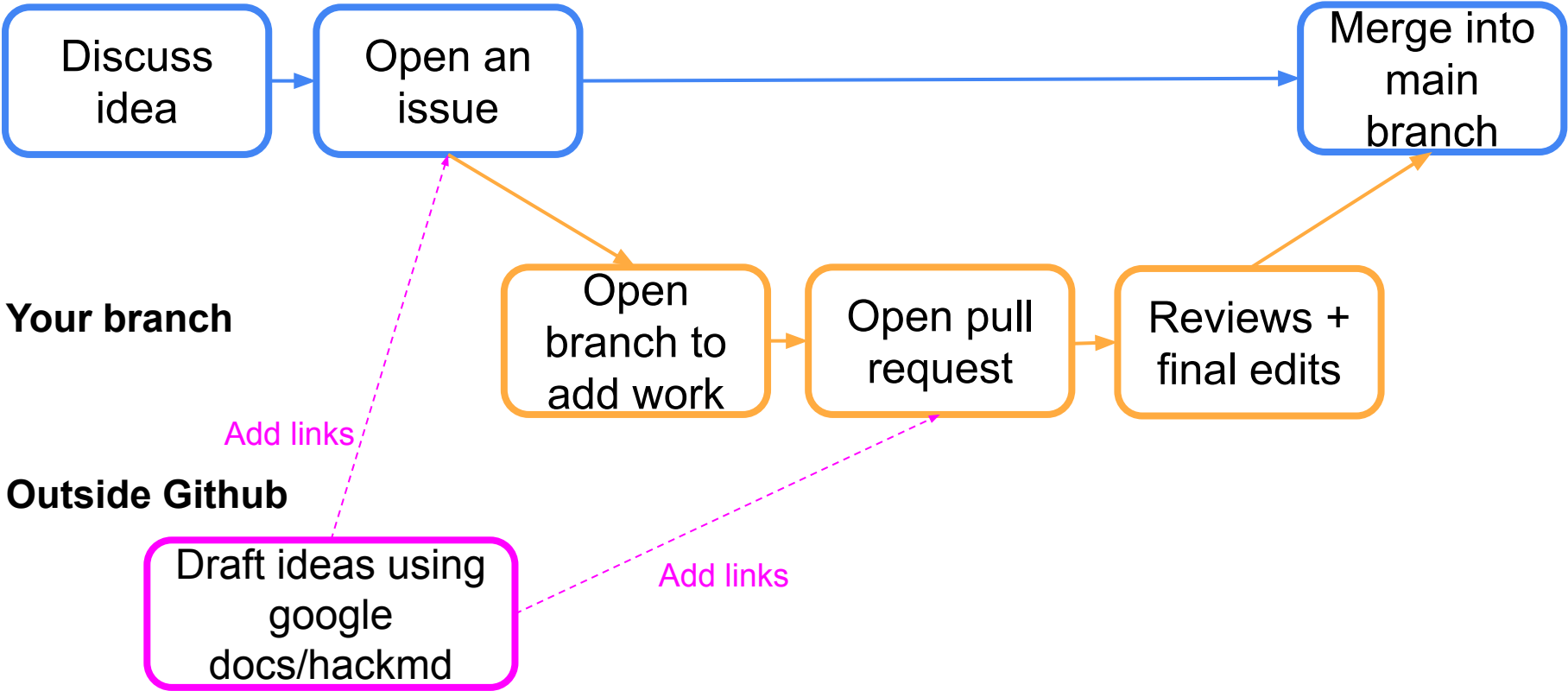


Outside Github



Add links

Add links



Useful GitHub resources

Useful GitHub resources (based on

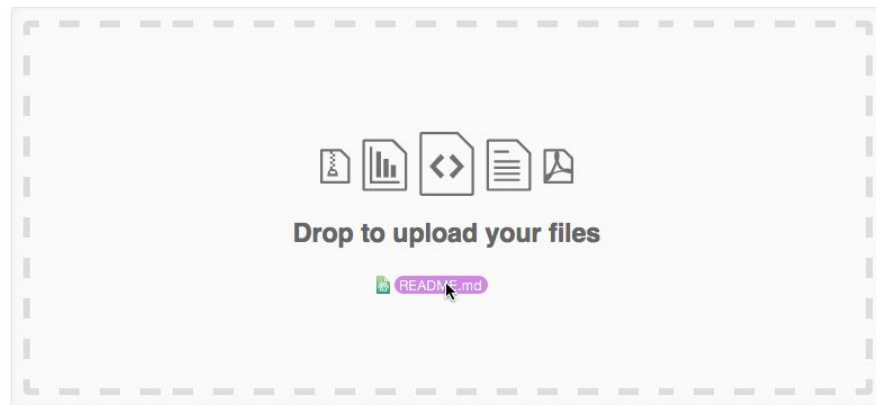
<https://github.com/alan-turing-institute/the-turing-way/blob/master/workshops/github-workshop/2021-05-13-intro-workshop.md>)

- [GitHub for Collaboration \(from Mozilla\)](#)
- [Understanding the GitHub workflow](#)
- [A Friendly Github Intro Workshop \(from Danielle Robinson\)](#)
- [GitHub help](#)
- [Oh Shit, Git!?!](#)
- [Try Git](#)
- [Git book](#)
- [GitHub glossary](#)
- [Writing on Github](#)
- [Markdown Cheatsheet](#)
- [Git workflow](#)

How to add files?

- Web interface
- [GitHub desktop](#)
- [Terminal/Git](#)
- [RStudio](#)

octo-repo /



<https://docs.github.com/en/repositories/working-with-files/managing-files/adding-a-file-to-a-repository>