

# Introduction to Git & GitHub

Jennifer Graham

Tiago Silva

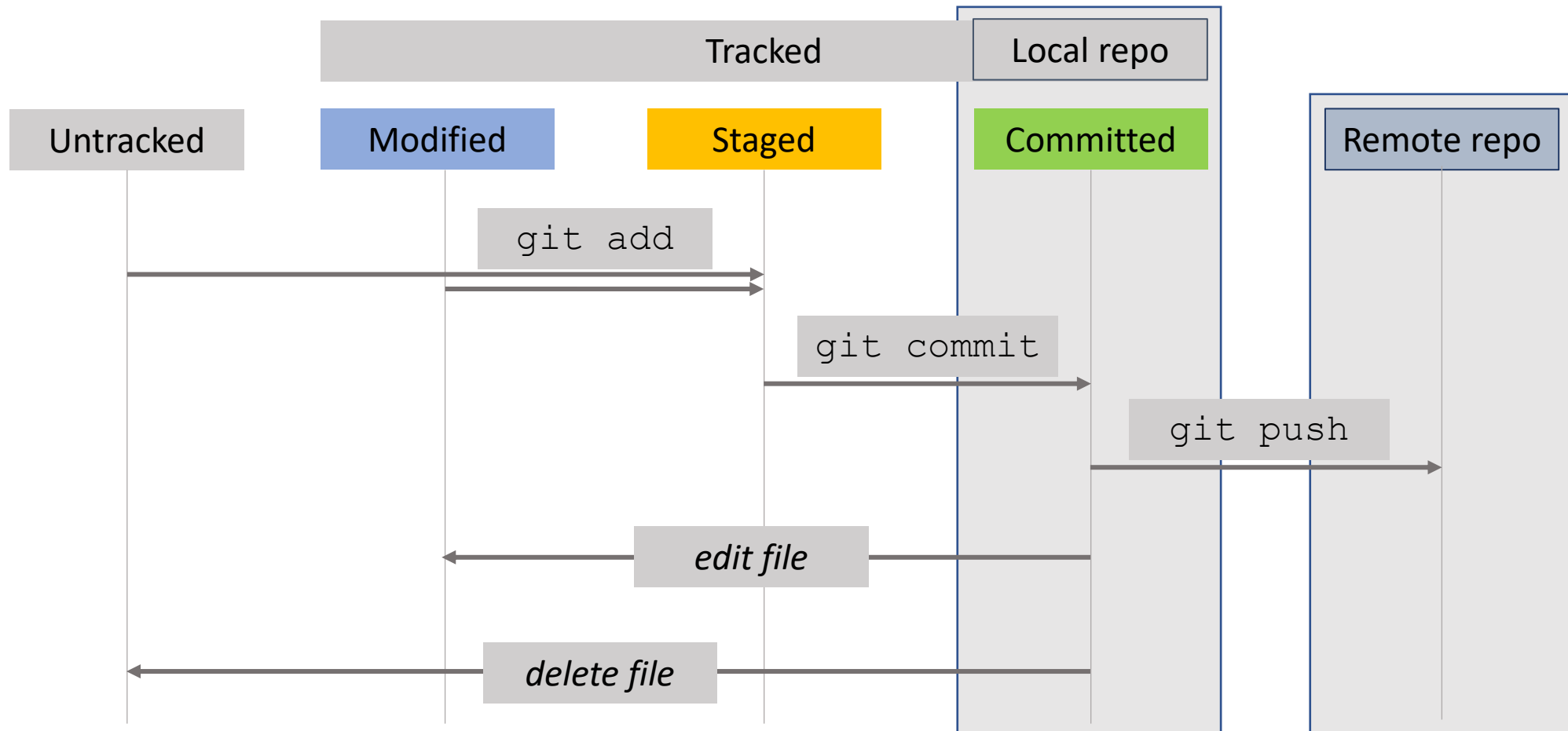
David Ryder

Stephen Gregory

Session 3

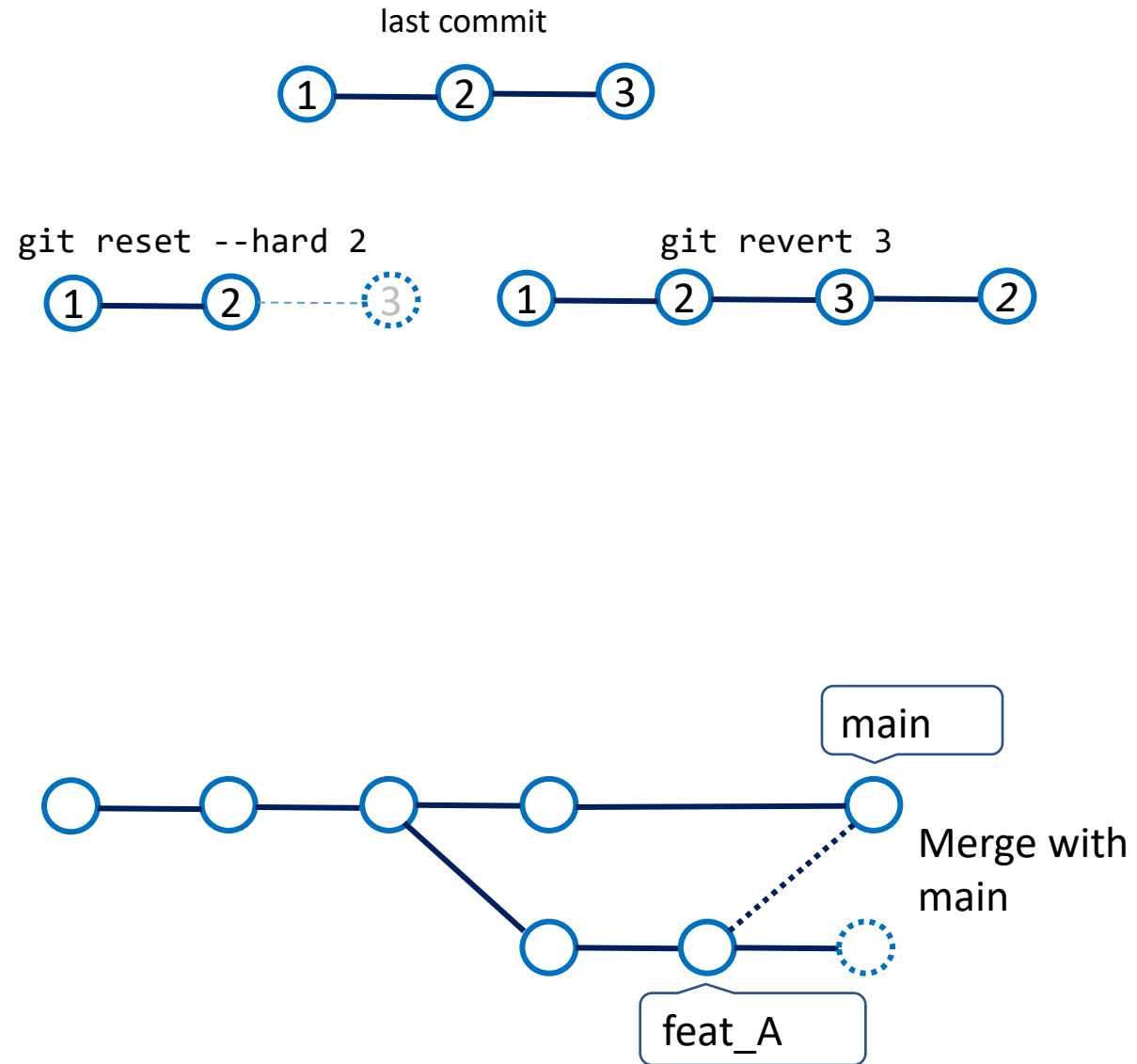
6th June 2024

Recap from Sessions 1 & 2 ...



```
git init                :: turn current directory into git repository.
git add <file>          :: stage the file for commit i.e. track changes.
git commit              :: confirm changes with message/explanation.
git push                :: publish/back-up changes on GitHub (remote server).

git status              :: show which files have been tracked or modified.
git diff <file>         :: show difference between current file and last commit.
git log <file>          :: show commit history [for file]
```



## Clearing up mistakes

Restore previous version of file

```
git checkout [<commit>] -- <path>
```

Go back to last commit, discarding all changes in working directory  
Discard all commits after [<commit>]

```
git reset --hard HEAD  
git reset --hard [<commit>]
```

Undo git add (undo staging or tracking file)

```
git reset [HEAD] <path>
```

Undo changes introduced by single commit, recording this in history

```
git revert <commit>
```

<code>git branch</code>	:: List local branches ( * = you are here ) .
<code>git branch &lt;branch-name&gt;</code>	:: Create new branch
<code>git checkout &lt;branch-name&gt;</code>	:: Change to <branch-name> (updates local directory)
<code>git diff &lt;branch1&gt;..&lt;branch2&gt; &lt;path&gt;</code>	:: Check difference between two branches (in optional <path>)
<code>git merge &lt;branch-name&gt;</code>	:: Merge changes from <branch-name> into current branch

# Aside: main vs master

- GitHub now uses “main” as the default central branch name (rather than master)
  - In 2020, other software companies announced similar changes.
- If you’re working with older software, or creating repositories locally with an older version of git, you may find that “master” is still the default branch name.
  - Your default will depend on your local git settings.
- There are a few different ways you can update this...



# How to change the “central” branch name?

**If you have just setup a new repository on GitHub, but don't want to rename anything locally:**

```
git remote add origin <GitHub_Path>
git push -u origin master
```

**If you have just setup a new repository on GitHub, and want to rename your current local branch to main:**

```
git remote add origin <GitHub_Path>
# Rename the current branch to main
git branch -M main
git push -u origin main
```

**If you rename an existing repository on GitHub, and want to update your local repository**

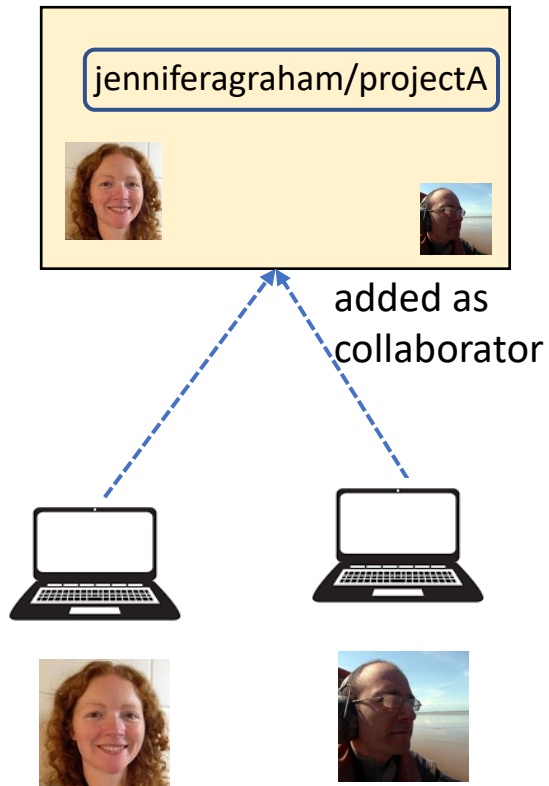
```
# Rename the master branch to main
git branch -m master main
git fetch origin
git branch -u origin/main main
git remote set-head origin -a
```

**(NB. GitHub will give tips for the above when creating a new repository)**

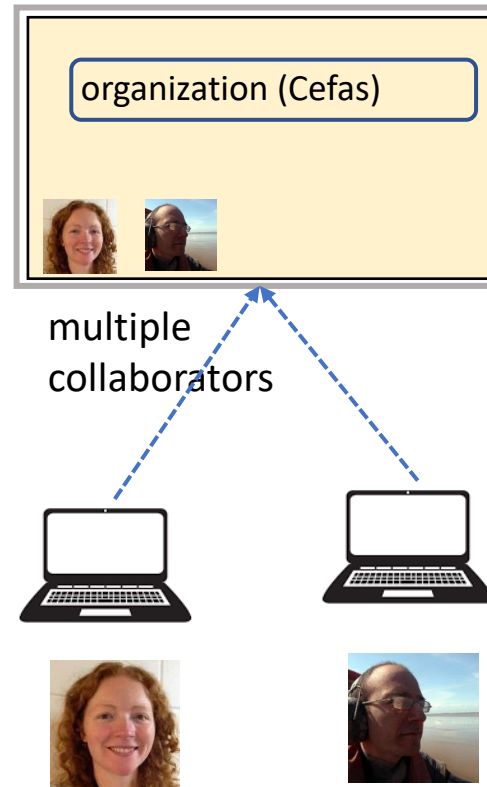
How to collaborate?

# Collaboration models

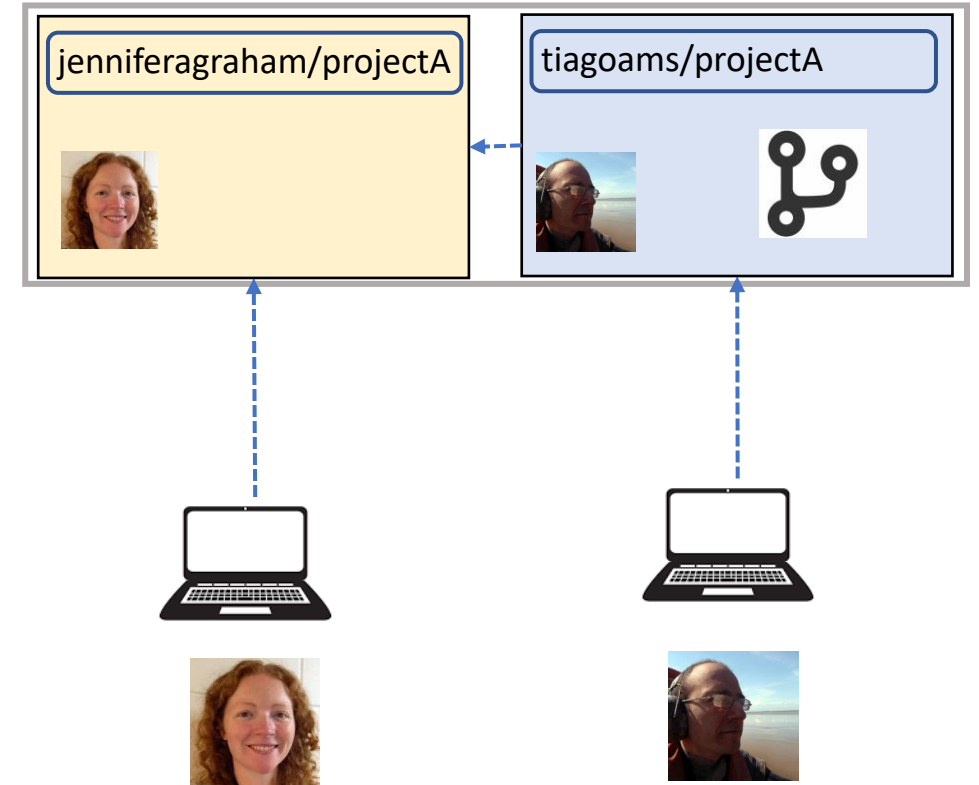
Personal account +  
add collaborator



GitHub Organization  
(CefasRepRes, UEABIO, UeaPy)

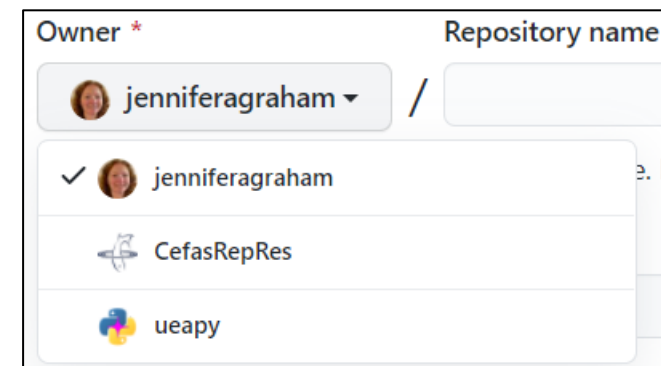


Forked + pull request  
original repo      forked repo



# Organisations

- Organisations can add further functionality on GitHub.
  - For those at Cefas, we have [CefasRepRes](#).
    - Further guidance here: <https://www.github.com/CefasRepRes/cefas-guides/>
  - For those elsewhere, if you don't have one already, you can always [create your own...](#)
- Develop private repositories that are still visible to all members.
- Easily share and organise work within the organisation.
- Repositories are technically “owned” by the organisation.
  - NB. Select the correct “owner” on creation.



The screenshot shows the GitHub repository creation interface. At the top, there are two labels: "Owner \*" and "Repository name". Below "Owner \*" is a dropdown menu currently showing "jenniferagraham". Below the dropdown is a list of suggestions: "jenniferagraham" (with a checkmark icon), "CefasRepRes" (with a repository icon), and "ueapy" (with a package icon). The "Repository name" field is empty and to the right of the owner dropdown.

Centre for Environment Fisheries

github.com/CefasRepRes?view\_as=public

CefasRepRes

Type to search

+>

Overview


Repositories406

Projects26

Packages

Teams16

People91



Centre for Environment Fisheries and Aquaculture Science (Cefas)

Cefas is a world leader in marine science and technology, providing innovative solutions for the aquatic environment, biodiversity and food security

42 followersUKhttps://www.cefas.co.ukgithub@cefas.gov.uk

Follow

Popular repositories

agriforwards-summer-school-2022

Public

Planning for the UEA / Cefas AgriFoRwArdS summer school 2022

Jupyter Notebook78

LeMaRns

Public

R41

Git\_Training

Public

HTML3

FishRAM-Shiny-App

Public

A Shiny app to run the Fisheries Resource Allocation Model.

R3

Git\_Training\_2019

Public

Introduction to Git & GitHub: How to manage and share your code

21

QGIS\_Training\_Cefas

Public

This repository contains QGIS training modules developed by Cefas

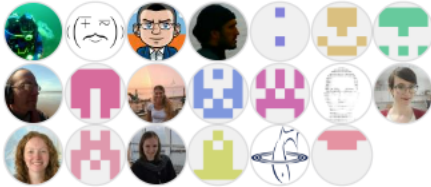
22

View as: Public

You are viewing the README and pinned repositories as a public user.

You can create a README file visible to anyone.

People



View all

Top languages

R

Python

Jupyter Notebook

HTML

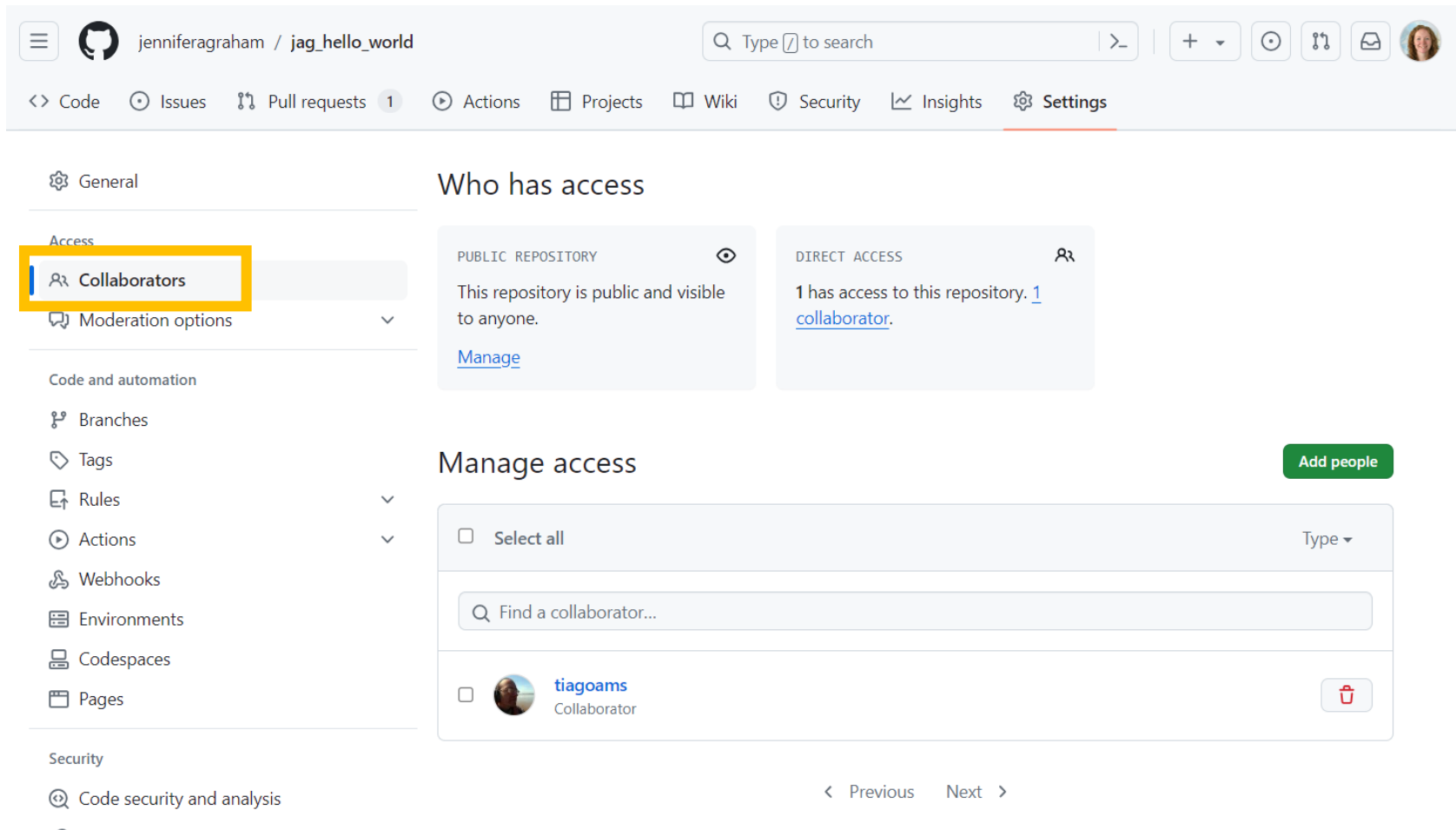
Fortran

Repositories

13

# Controlling repository access...

- Collaborators can be added to each repository.



The screenshot shows the GitHub repository settings page for 'jenniferagraham / jag\_hello\_world'. The 'Settings' tab is selected in the top navigation bar. On the left sidebar, the 'Access' section is expanded, and the 'Collaborators' option is highlighted with a yellow box. The main content area is titled 'Who has access' and shows two access types: 'PUBLIC REPOSITORY' (which is public and visible to anyone) and 'DIRECT ACCESS' (which shows that 1 collaborator has access to the repository, with a link to the collaborator). Below this, the 'Manage access' section is visible, featuring a search bar to find collaborators and a list of current collaborators. One collaborator, 'tiagoams', is listed with a red trash icon to remove them. A green 'Add people' button is located at the top right of the 'Manage access' section. The bottom of the page shows navigation links for 'Previous' and 'Next'.

jenniferagraham / jag\_hello\_world

Type to search

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

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

### Who has access

**PUBLIC REPOSITORY**

This repository is public and visible to anyone.

[Manage](#)

**DIRECT ACCESS**

1 has access to this repository. [1 collaborator](#).


### Manage access

[Add people](#)

☐ Select all

Type

Find a collaborator...

☐  **tiagoams**  
Collaborator

[Previous](#) [Next](#)

The screenshot shows the GitHub repository settings for 'jag\_hello\_world' by user 'jenniferagraham'. The 'Settings' tab is active, and the 'Collaborators' section is selected in the left sidebar. The 'Who has access' section shows 'PUBLIC REPOSITORY' status and 'DIRECT ACCESS' for one collaborator. The 'Manage access' section lists the collaborator 'tiagoams' with a yellow highlight around the entry. The interface includes a top navigation bar with repository navigation links and a sidebar with settings categories like General, Access, and Security.

- For personal repositories, there are two levels of permission:
  - **Repository owners** (Admin permissions)
  - **Collaborators** (aka Write access etc.)
- Only the owner can invite new collaborators.

- For organisation repositories, there are many levels available:


Role	Access
Admin	Full access i.e. repository owners
Maintain	Manage the repository with access to all but most sensitive settings.
Write	Can push changes to repo
Triage	Can manage issues but no write access.
Read	View code, make comments, but not change

- Only those with **Admin** access can invite new collaborators.
- Access can be grouped via “teams” as well as individuals.
- NB. Outside collaborators can be invited (don’t need to be a member to collaborate)

## Manage access

[Create team](#)
[Invite teams or people](#)


☐ Select all
 Type ▾ Role ▾

☐


**David Ryder**  
 davidryderuk2


Role: Write ▾

✕

☐


**Jennifer Graham**  
 jenniferagraham

✕

☐


**tiagoams**

✕

Choose role

Read

Can read and clone this repository. Can also open and comment on issues and pull requests.

Triage

Can read and clone this repository. Can also manage issues and pull requests.

✓ Write


Can read, clone, and push to this repository. Can also manage issues and pull requests.

Maintain

Can read, clone, and push to this repository. They can also manage issues, pull requests, and some repository settings.

Admin

Can read, clone, and push to this repository. Can also manage issues, pull requests, and repository settings, including adding collaborators.

[Security](#)
[Status](#)
[Docs](#)

[Contact GitHub](#)
[About](#)



# Restricted admin access

- Within an organisation, repositories are technically owned by the organisation, not individual.
  - Some settings can only be changed/managed by the **organisation owner** e.g.

## Danger Zone

<b>Change repository visibility</b> Organization members can't change repo visibility.	
<b>Transfer ownership</b> Organization members cannot transfer repositories	
<b>Archive this repository</b> Mark this repository as archived and read-only.	<a href="#">Archive this repository</a>
<b>Delete this repository</b> Organization members cannot delete repositories.	



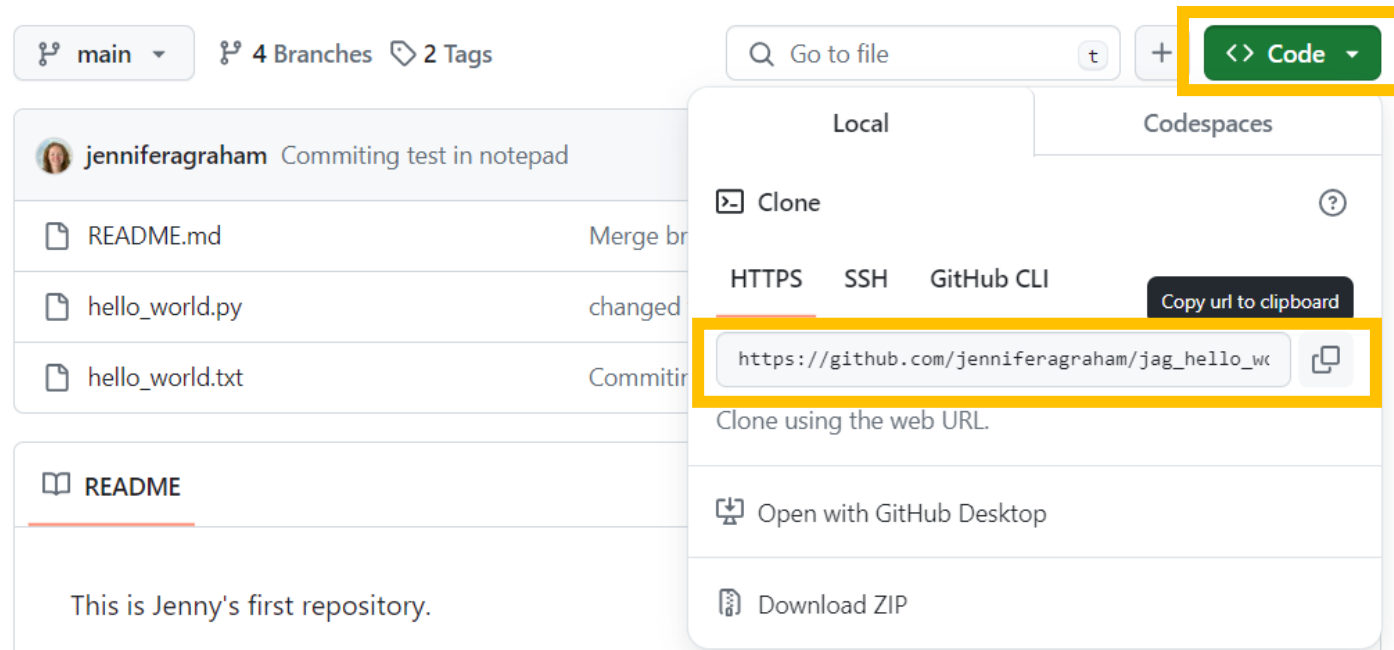
- Organisation owners can decide on the level of access for members.
- Only owners can invite/approve new members.

(How) can I modify someone  
else's repository?

It depends on your access...

# Admin, write or collaborator access...

- Use the same workflow discussed already.
  - You can push changes to main, and create branches for developments.
- Use `git clone` to make a local copy.



```
git clone <copied-path>  
download into new local repository
```

# fetch, merge (or pull)



- Once you start collaborating, you need to be aware that others may (will) commit and push changes.
  - If you try to push changes, you may see an error message...
- To bring your local repository up to date with remote changes:

```
git fetch      :: brings in information on changes from the remote
git merge     :: merge those changes into your repository
--
git pull      :: short cut to fetch and merge in one step (if confident
               that there are no conflicts)
```

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git status

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git push

To https://github.com/jenniferagraham/jag\_hello\_world.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/jenniferagraham/jag\_hello\_world.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$

git pull

:: fetch updates from remote and merge in one step

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git fetch

git fetch

:: fetch retrieves updates from remote server,  
but does not change your local repository files.

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/jenniferagraham/jag\_hello\_world

ddd7a97..c07fb9d master -> origin/master

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git status

On branch master

Your branch and 'origin/master' have diverged,  
and have 1 and 1 different commits each, respectively.  
(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$



JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/jenniferagraham/jag\_hello\_world

ddd7a97..c07fb9d master -> origin/master

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git status

On branch master

Your branch and 'origin/master' have diverged,  
and have 1 and 1 different commits each, respectively.

(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git merge

git merge

:: merge retrieved commits into local repository.

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/jenniferagraham/jag\_hello\_world

ddd7a97..c07fb9d master -> origin/master

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git status

On branch master

Your branch and 'origin/master' have diverged,  
and have 1 and 1 different commits each, respectively.

(use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

\$ git merge

Auto-merging README.md

Merge made by the 'recursive' strategy.

README.md | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)

JG10@G6W1YF2 MINGW64 ~/OneDrive - CEFAS/GitHub/Repos/jag\_hello\_world (master)

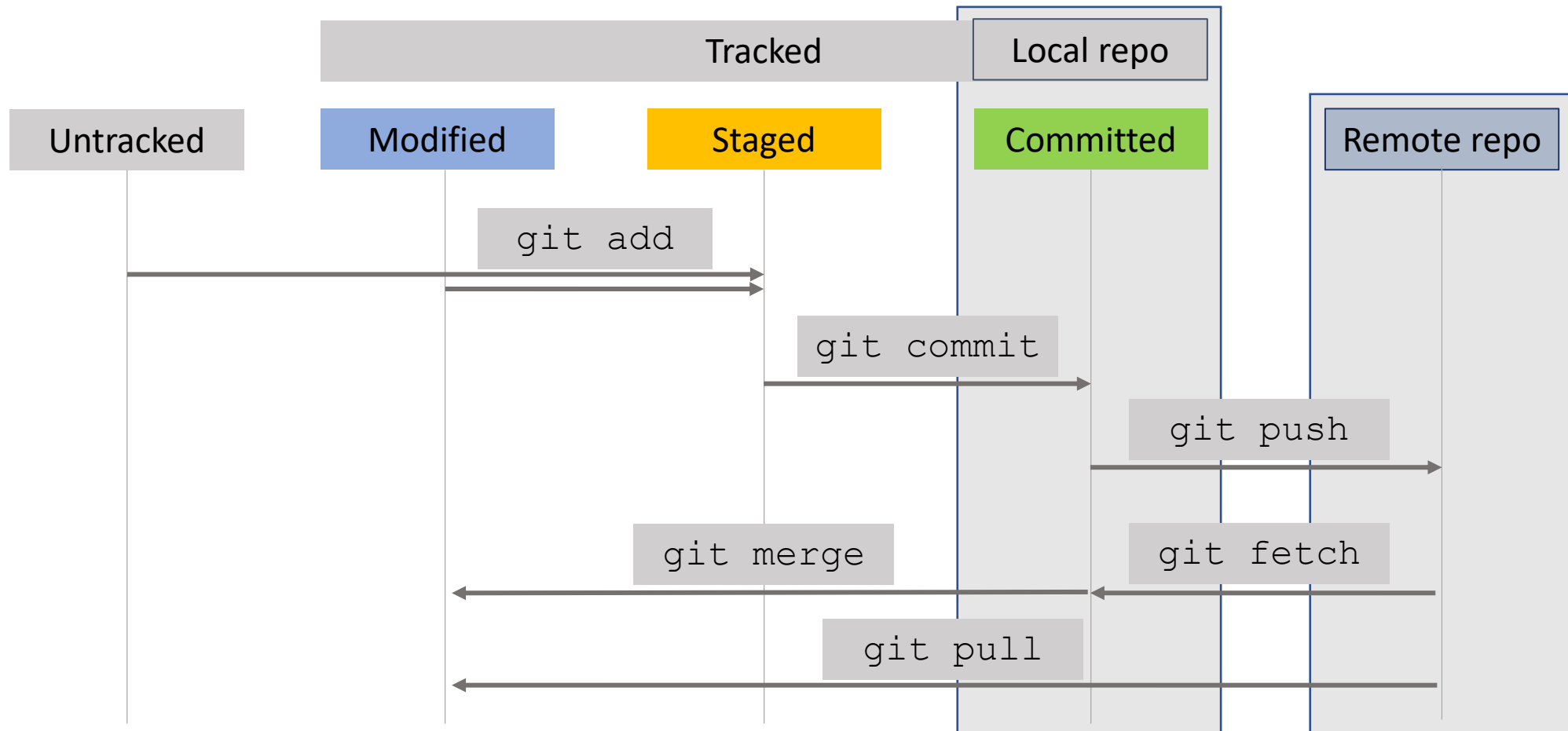
\$

```
git clone <copied-path>    :: download into new local repository
```

```
git fetch                  :: brings in information on changes from the remote
```

```
git merge                  :: merge changes into your local repository
```

```
git pull                   :: short cut to fetch and merge in one step
```



# If you have only read access...

e.g. For public repositories, or those within your organisation.

- You can still clone the repository, then track locally as normal.
- However, if you modify the repository, you won't be able to push changes back to the remote repository.
- If you want to make a copy and intend to make developments, consider forking...

# What is a fork?



- Similar to a branch, in that both can be used to develop new features, away from the main branch.
- In basic terms, the difference is:
  - If you have write access for the repository -> branch.
  - If you don't have write access -> fork.
- Forking creates a copy of the original repository in your ownership (including its history).
- Repository remains linked to the original (main) so you can see how the developments diverge.
- From a fork, you must create a “pull request” to merge changes.
  - This will notify the code owner, and allow them to review changes before deciding whether or not to accept/merge (i.e. request them to pull changes into their repository).

# How to create a fork?

Note: you can't create a fork owned by the same individual or organisation as the original.

The screenshot shows the GitHub interface for the repository 'jenniferagraham / jag\_hello\_world'. The 'Fork' button is highlighted with a yellow box, and a tooltip is visible: 'Fork your own copy of jenniferagraham/jag\_hello\_world to your account'. The repository details show 5 commits, 1 branch, and 0 releases. The commit history lists two commits: 'Testing commit with vi' (2 hours ago) and 'Fixing typo' (7 days ago). The README content is visible at the bottom.

My first repository [Manage topics](#) [Edit](#)

5 commits 1 branch 0 releases


Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)




[jenniferagraham](#) Testing commit with vi Latest commit 024d609 2 hours ago







<a href="#">README.md</a>	Testing commit with vi	2 hours ago
<a href="#">hello_world.py</a>	Fixing typo	7 days ago

[README.md](#)

This is Jenny's first repository test.

 **CefasRepRes / jag\_hello\_world** Private  
forked from jenniferagraham/jag\_hello\_world

 Watch 0  Star 0  Fork 1



 Code  Pull requests 0  Projects 0  Security  Insights  Settings


My first repository Edit

[Manage topics](#)


5 commits1 branch0 releases

Branch: masterNew pull requestCreate new fileUpload filesFind FileClone or download


This branch is even with jenniferagraham:master. Pull request Compare

 jenniferagraham


testing commit with viLatest commit 024d609 19 hours ago

 README.md

Testing commit with vi19 hours ago

 hello\_world.py

Fixing typo8 days ago

 README.md

This is Jenny's first repository test.



🔑 CefasRepRes / jag\_hello\_world Private  
forked from jenniferagraham/jag\_hello\_world

👁 Watch 0 ★ Star 0 🍴 Fork 1

🔗 Code 📄 Pull requests 0 📁 Projects 0 🛡 Security 📊 Insights ⚙ Settings

My first repository

Edit

[Manage topics](#)

📦 5 commits

🌿 1 branch

📦 0 releases

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾

This branch is 2 commits behind jenniferagraham:master.

🔗 Pull request 📄 Compare

👤 jenniferagraham testing commit with vi

Latest commit 024d609 19 hours ago

📄 README.md

Testing commit with vi

19 hours ago

📄 hello\_world.py

Fixing typo

8 days ago




📄 README.md

✎

This is Jenny's first repository test.





 [jenniferagraham](#) / [jag\\_hello\\_world](#) Private

 Watch 0  Star 0  Fork 1

[Code](#) [Issues](#) 0 [Pull requests](#) 0 [Projects](#) 0 [Security](#) [Insights](#) [Settings](#)

- Pulse
- Contributors
- Traffic
- Commits
- Code frequency
- [Dependency graph](#)
- Network
- Forks**

 [jenniferagraham](#) / [jag\\_hello\\_world](#)  
↳  [CefasRepRes](#) / [jag\\_hello\\_world](#)

Note: You can't create a private fork, it will always be visible to the original owner.

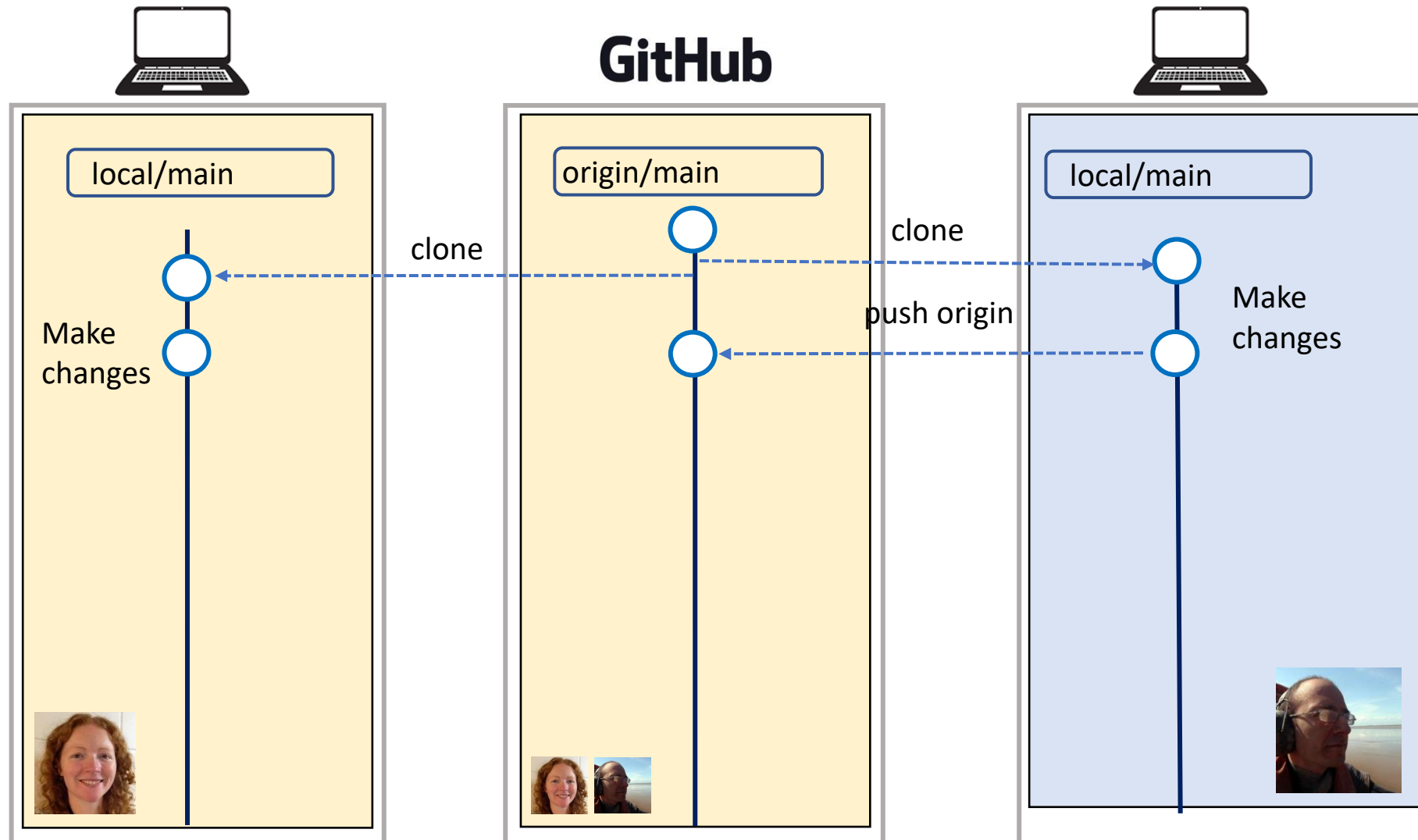
Branch	Fork
<ul style="list-style-type: none"><li>● Created in your repository</li><li>● You have total control over what you merge</li><li>● Pull requests are sent to your own repository</li><li>● Always intended to merge successful branches into master</li></ul>	<ul style="list-style-type: none"><li>● Creates an independent copy of another person's repository on your account so you can make changes</li><li>● You can suggest changes to the original project but they don't have to merge them</li><li>● Pull requests automatically go back to the original project</li><li>● Allows you to take a GitHub project in a different direction - may never intend to merge with original project</li></ul>

Collaboration workflows...

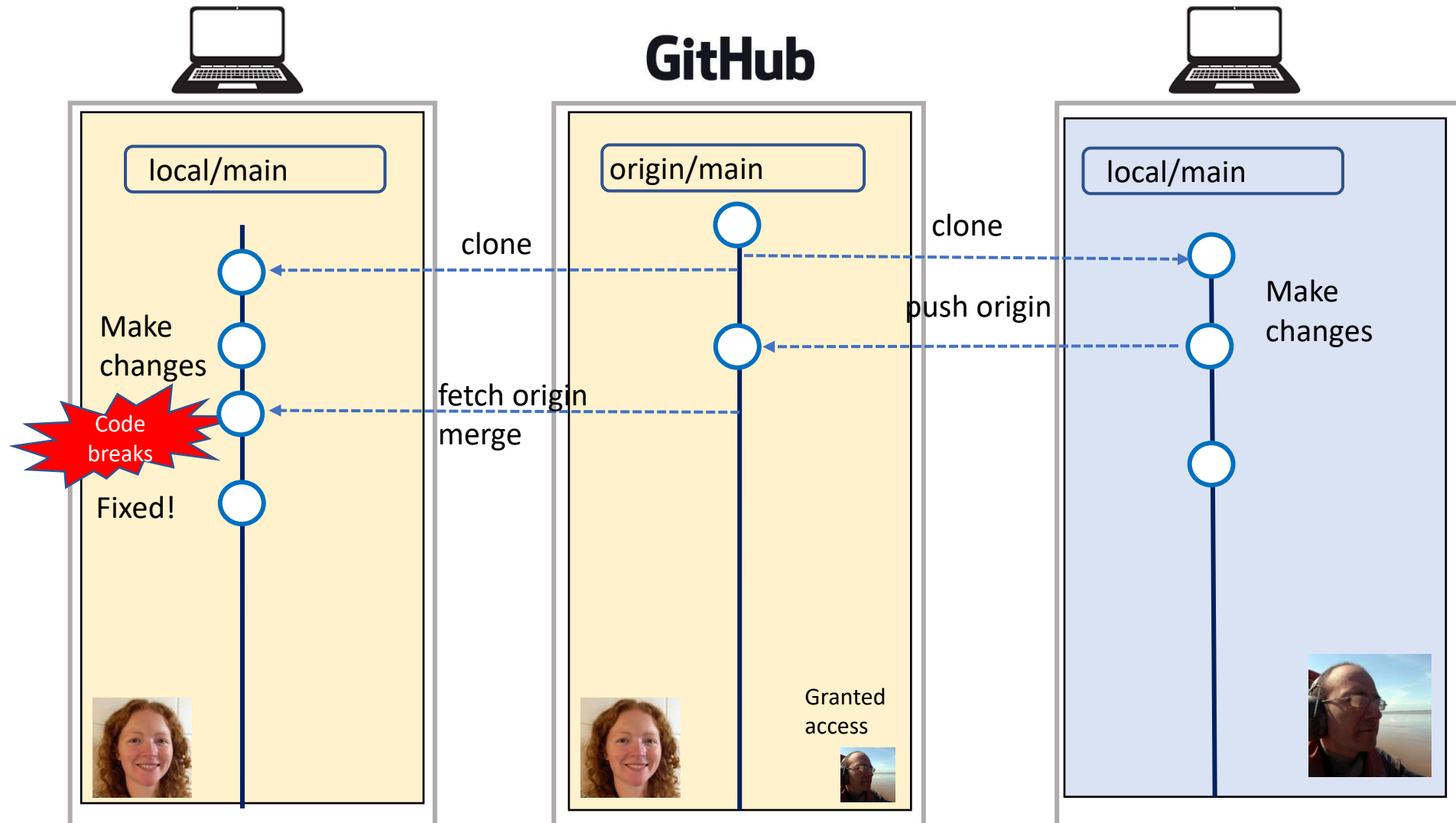
# Collaboration workflows...

- git is just the tool
- Many workflows can be used, making use of various features available, and will depend on:
  - number of collaborators
  - frequency code is written
  - type of organisation (community, corporation)
  - criticality of the code

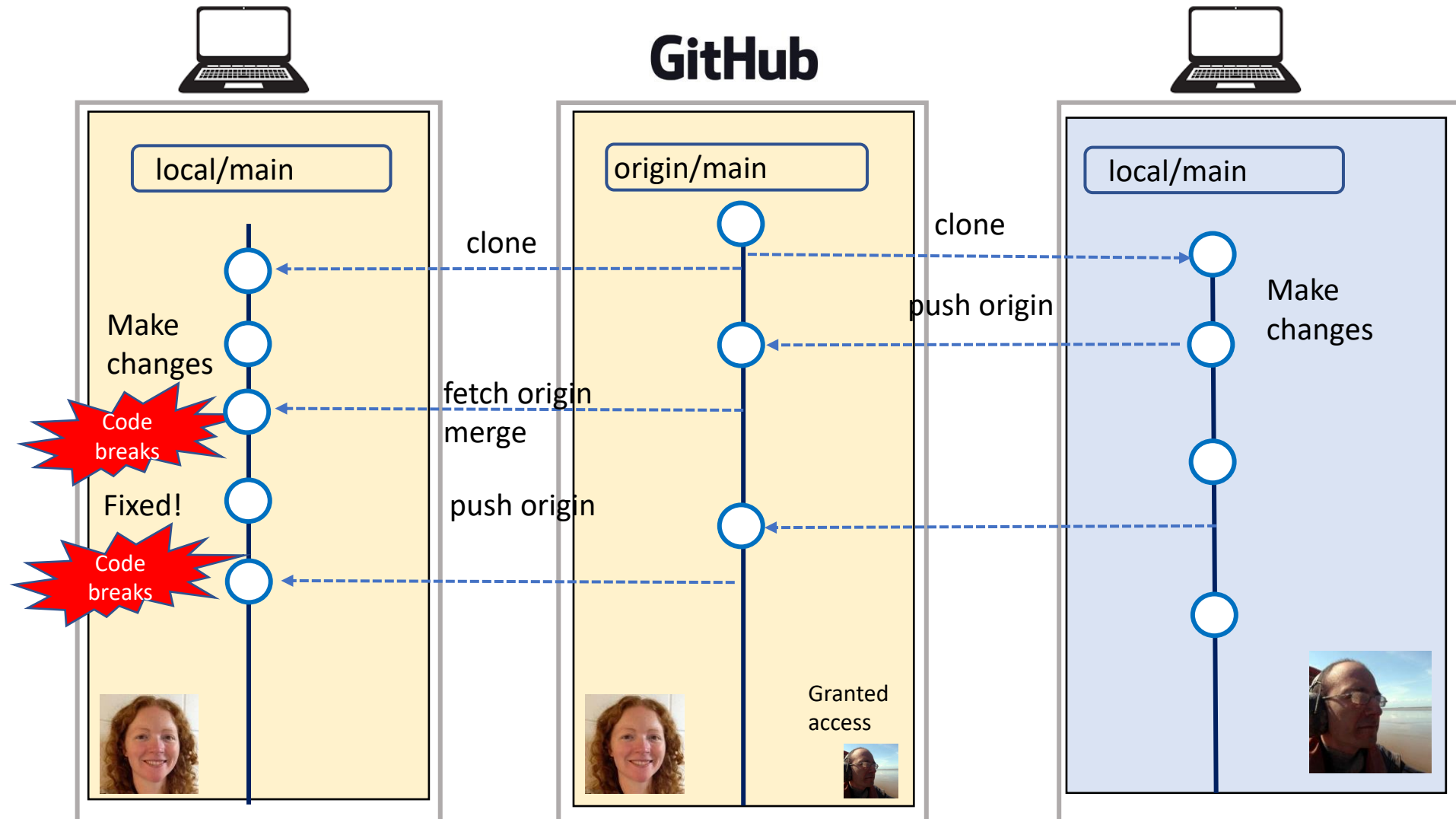
# Bad workflow: competitors vs. collaborators



# Bad workflow: competitors vs. collaborators

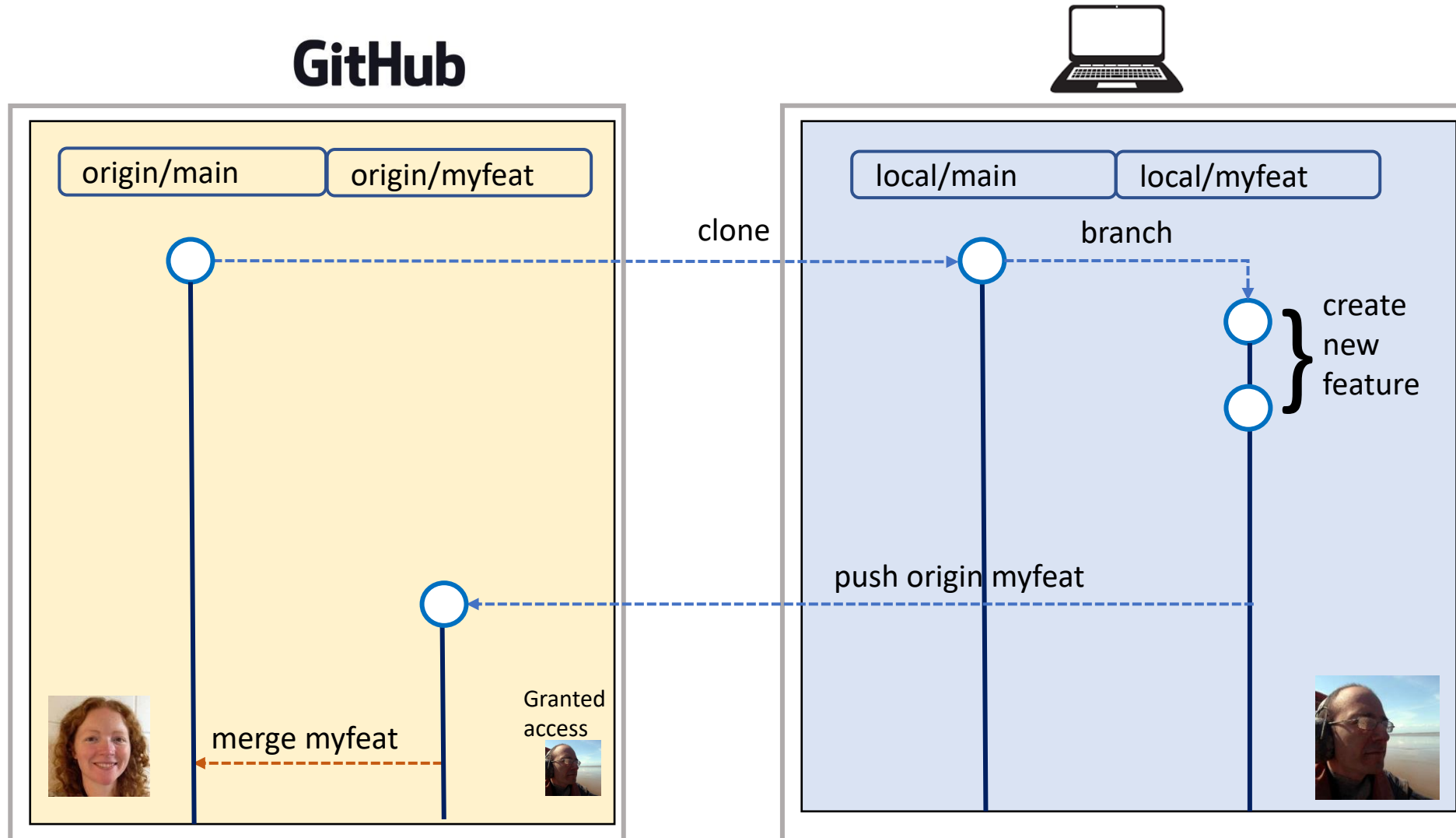


# Bad workflow: competitors vs. collaborators





# Shared access workflow: feature branches



# Shared access workflow: creating and pushing a branch

## 1. Make a local repo

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$
```

# Shared access workflow: creating and pushing a branch

1. Make a local repo

2. Create branch

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$ git branch fixreadme
$ git checkout fixreadme
$ git branch
* fixreadme
  main
$
```

# Shared access workflow: creating and pushing a branch

1. Make a local repo

2. Create branch

3. Modify

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$ git branch fixreadme
$ git checkout fixreadme
$ git branch
* fixreadme
  main
$ notepad README.md
$ git commit -am "Update name of UEA cluster"
[main 1ba2db8] Update name of UEA cluster
1 file changed, 1 insertion(+), 1 deletion(-)
$
```

# Shared access workflow: creating and pushing a branch

1. Make a local repo

2. Create branch

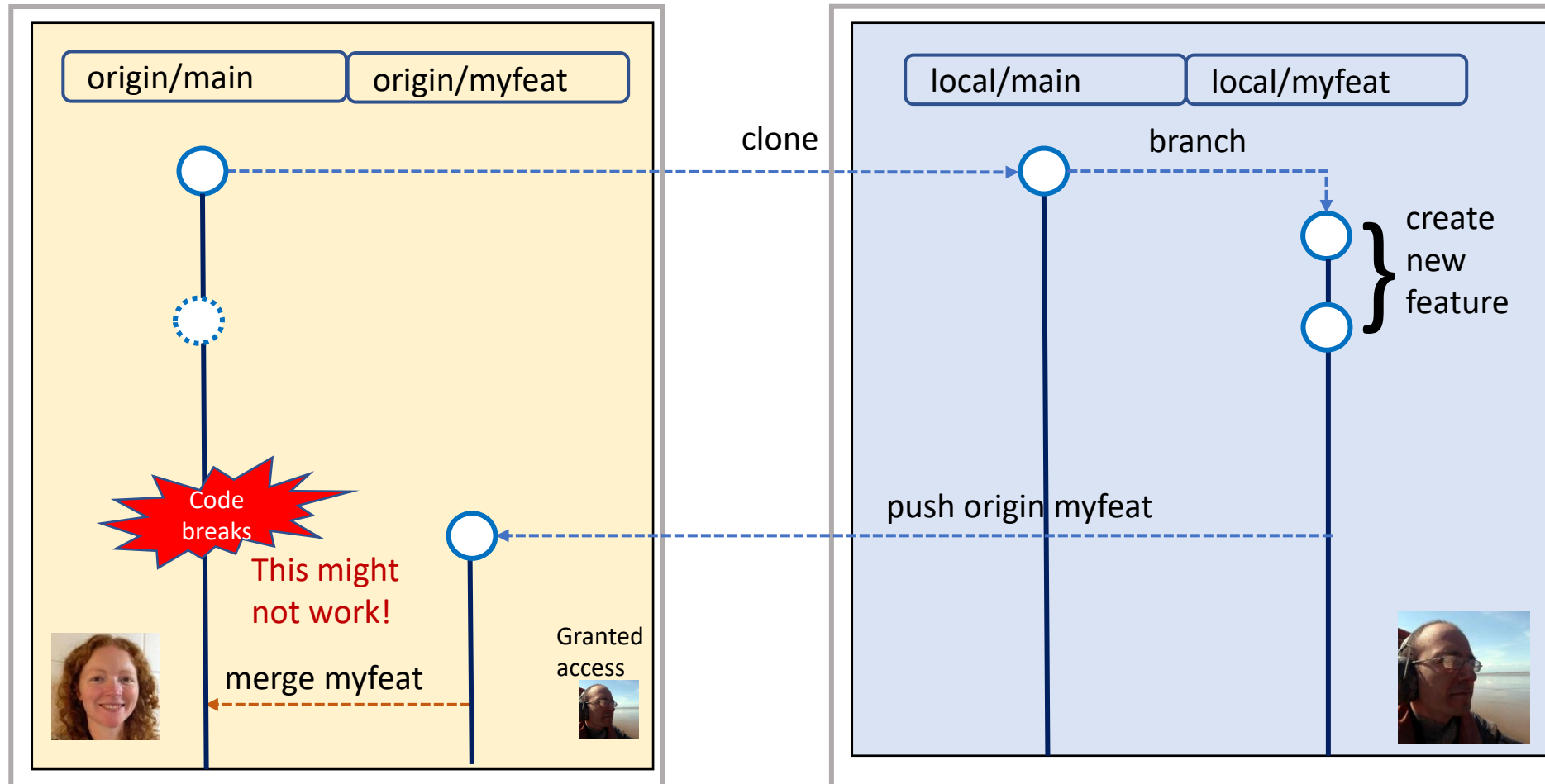
3. Modify

4. Share branch

```
$ git clone https://github.com/jenniferagraham/jag_hello_world.git
Cloning into 'jag_hello_world'...
$ cd jag_hello_world/
$ git branch fixreadme
$ git checkout fixreadme
$ git branch
* fixreadme
  main
$ notepad README.md
$ git commit -am "Update name of UEA cluster"
[main 1ba2db8] Update name of UEA cluster
1 file changed, 1 insertion(+), 1 deletion(-)
$ git push origin
Enumerating objects: 5, done.
(...)
* [new branch]      fixreadme -> fixreadme
```

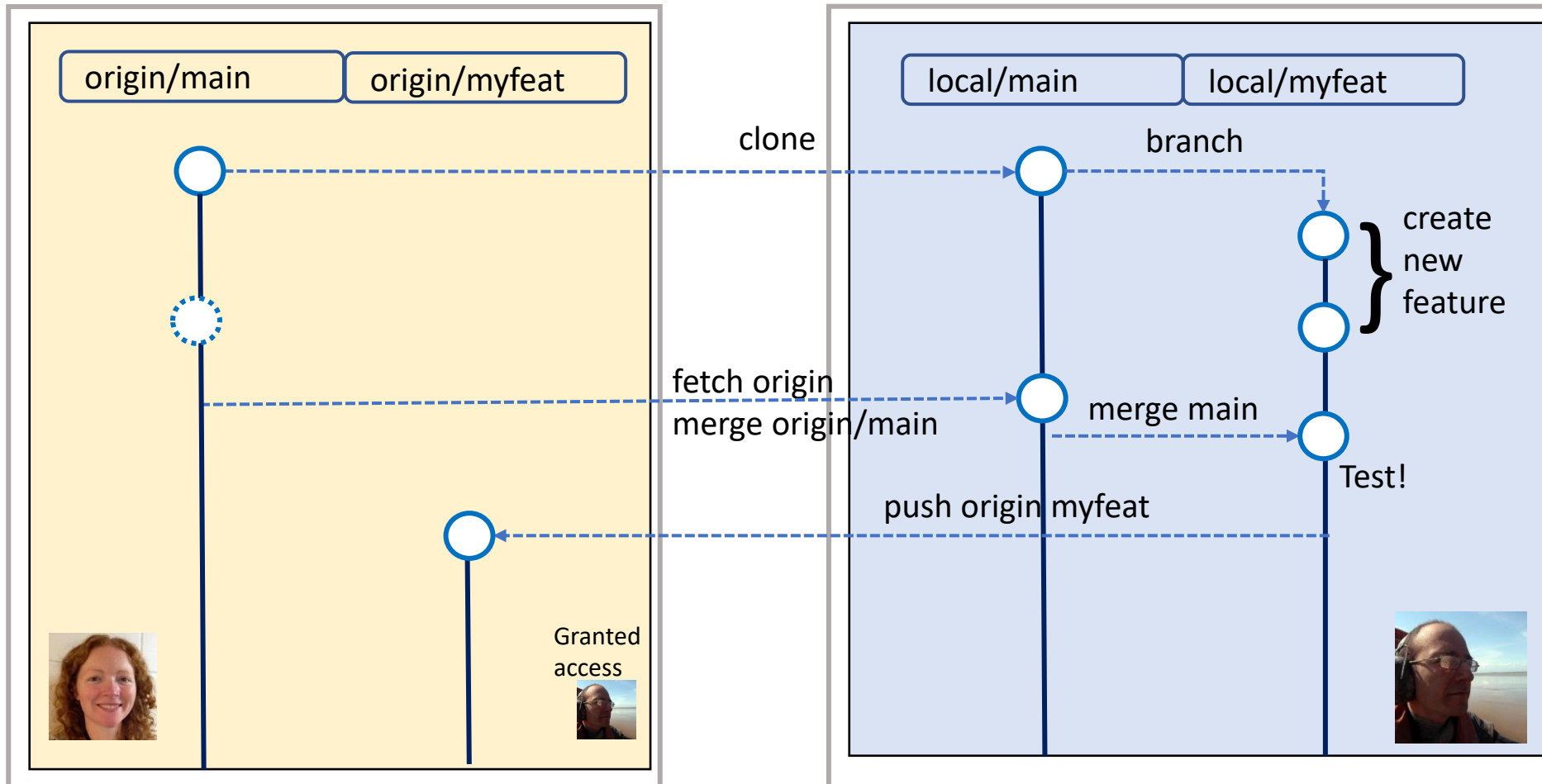
# What if main branch has changed?

**GitHub**



# Shared access workflow: what if origin/main has changed?

**GitHub**



# Shared access workflow: merging with changes on origin

## 3. Modify

## 4. update local main

## 5. merge to new branch

## 6. Share branch

```
$ git commit -am "Update name of UEA cluster"
[main 1ba2db8] Update name of UEA cluster
1 file changed, 1 insertion(+), 1 deletion(-)

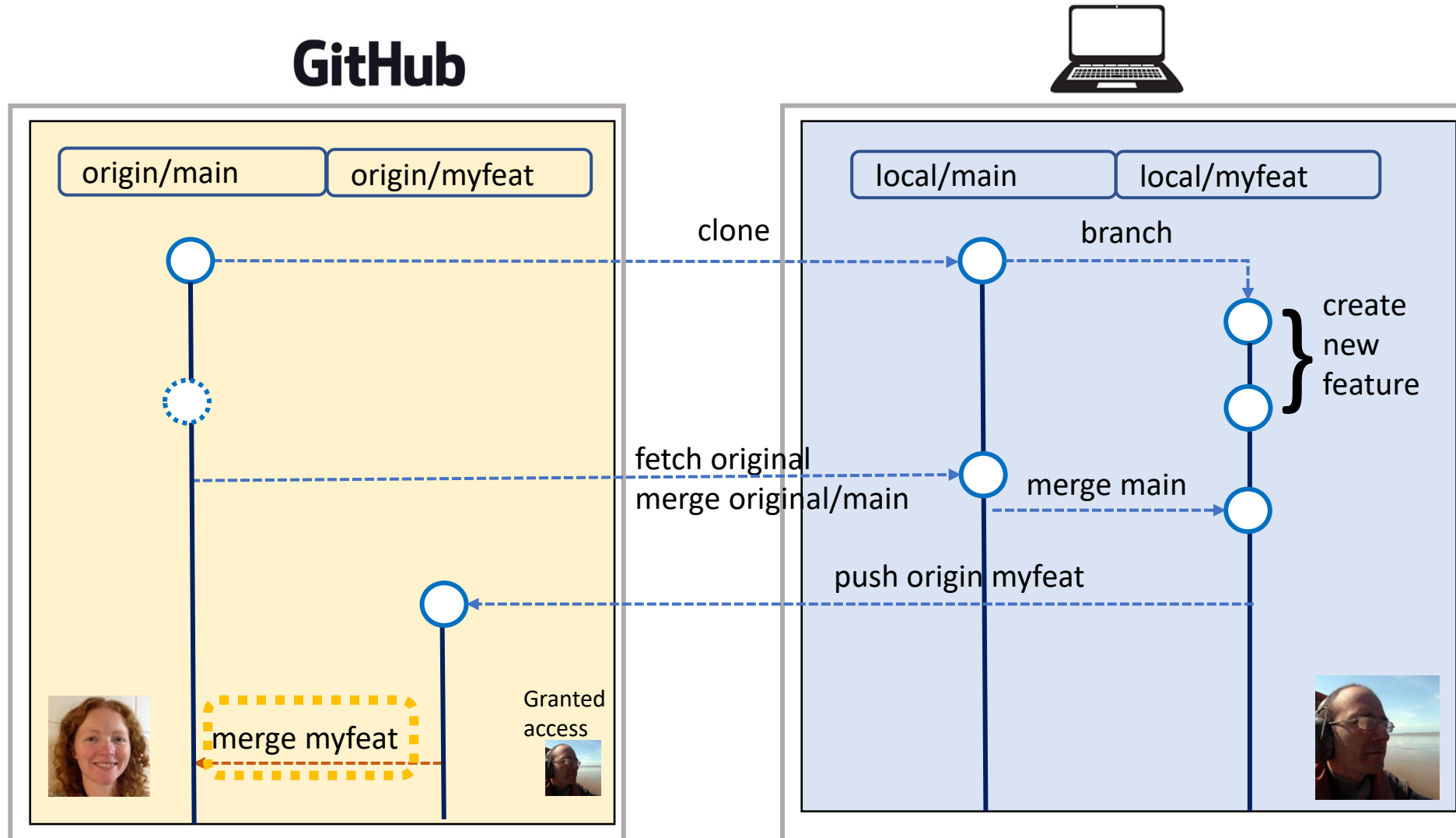
$ git pull
$ git pull
(...)

$ git merge main
Merge made by the 'recursive' strategy.
README.md | 4 +++-
fileC.txt | 1 +
2 files changed, 4 insertions(+), 1 deletion(-)
create mode 100644 fileC.txt

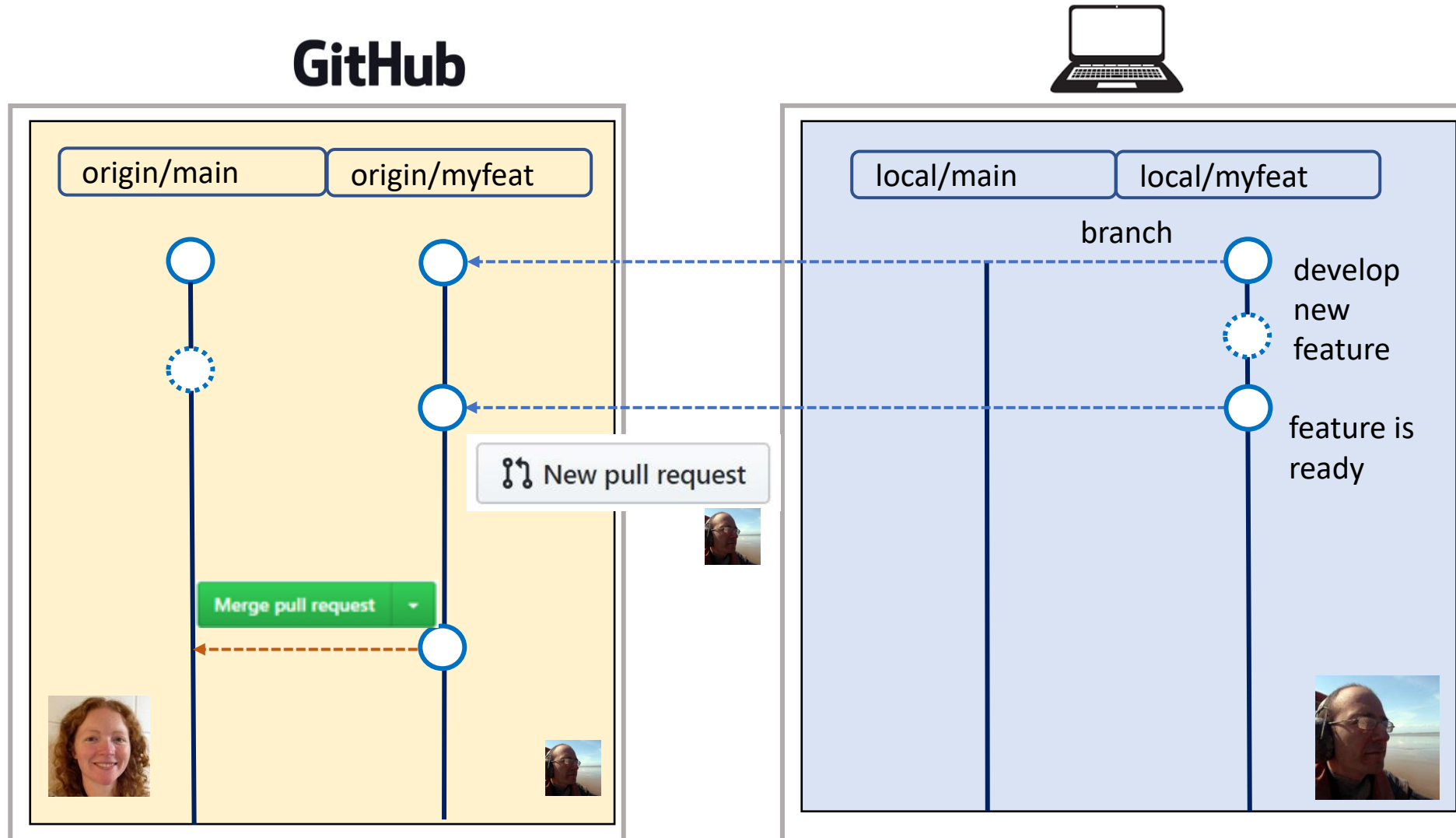
$ git push origin
Enumerating objects: 5, done.
(...)
* [new branch]      fixreadme -> fixreadme
```



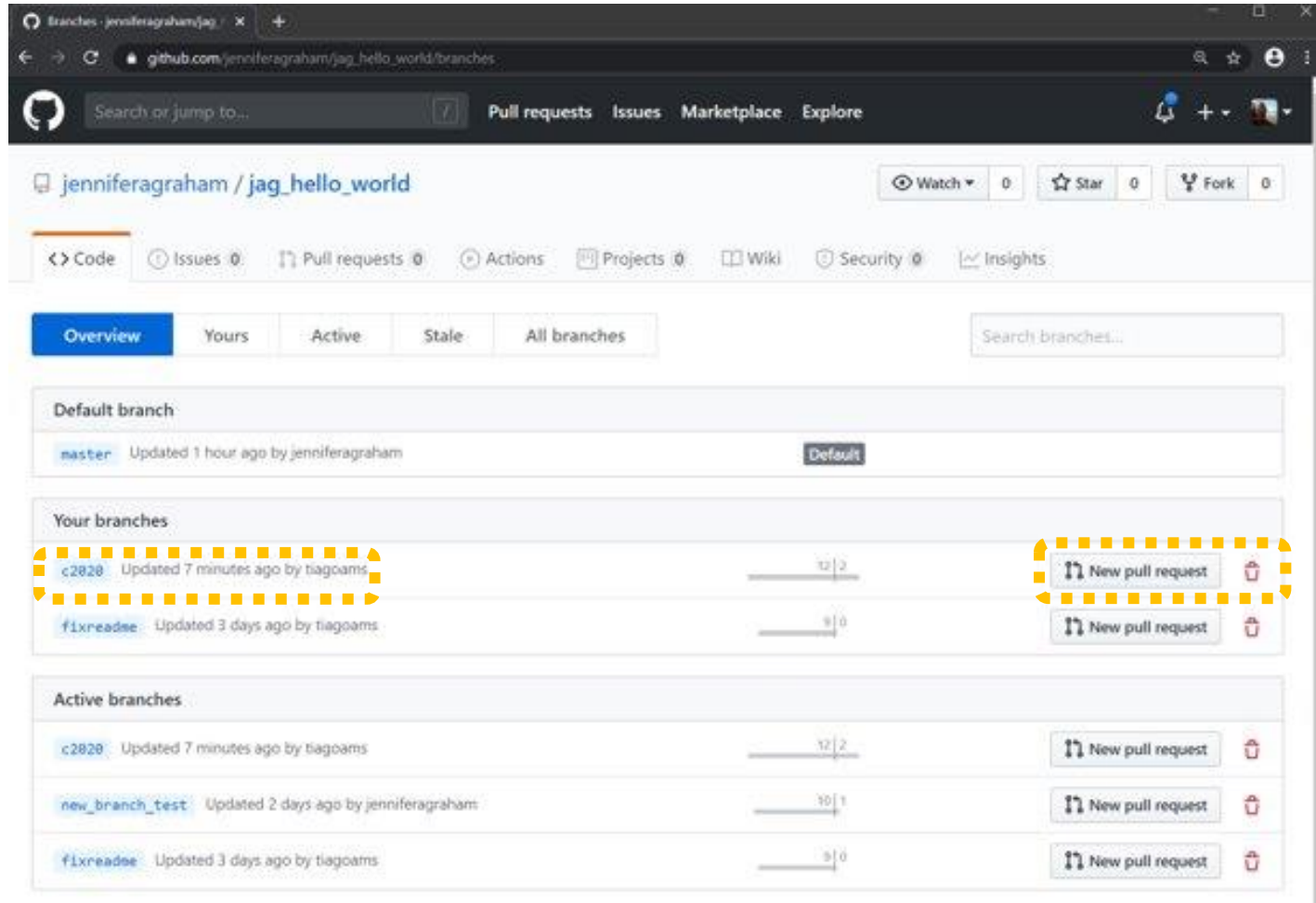
# Shared access workflow: how is the merging done?



# Recommended option: create a pull request



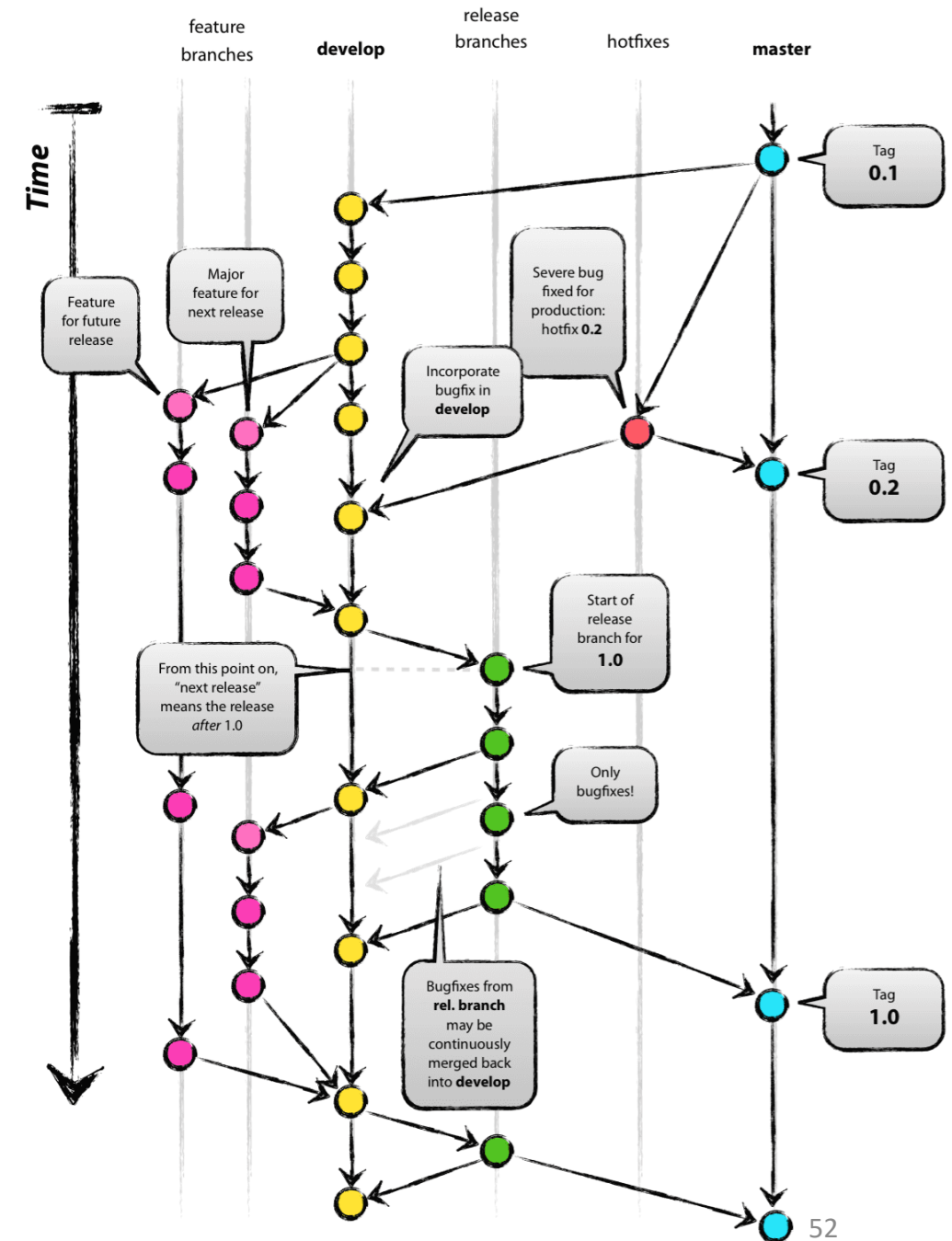
# Shared access workflow: the pull request can be inspected and accepted/rejected on GitHub



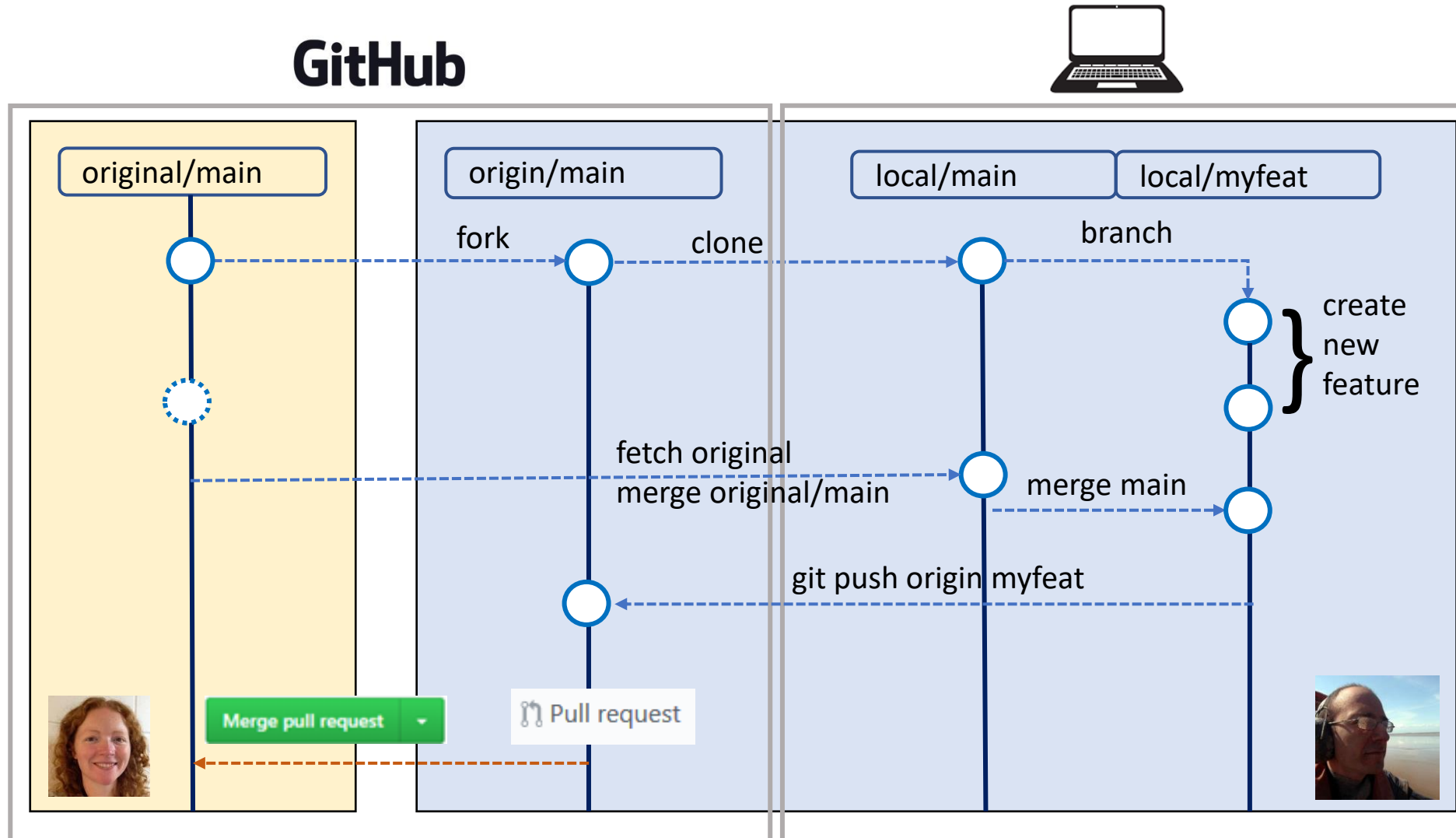
# Shared access workflow: Gitflow

<https://nvie.com/posts/a-successful-git-branching-model/>

- Many software developers use strict branching systems to ensure “the main is always shippable”
  - i.e. changes always tested on development branch, so they don’t break the code!



# Fork workflow: no privileged access required



# Pull requests: the contributor vs owners view

# No write access: create a pull request

Branch: myfeat ▾

New pull request


Create new file


Upload files


Find File

Clone or download ▾


This branch is 1 commit ahead of jenniferagraham:master.

 Pull request

 Compare


 **tiagoams** Added time dependent salutations

Latest commit 9536440 11 minutes ago

 [README.md](#)

Update README remote

2 days ago

 [hello\\_world.py](#)


Added time dependent salutations

11 minutes ago


# No write access: create a pull request

## 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 repository: jenniferagraham/jag\_hello\_world base: master head repository: tiagoams/jag\_hello\_world compare: myfeat

✓ **Able to merge.** These branches can be automatically merged.




Added time dependent salutations

Write

Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ 📌 ↶

I added a different salutation for different times of the day. It requires the datetime module.


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

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

Reviewers

Suggestions

 jenniferagraham [Request](#)

Assignees

No one—assign yourself

Labels

None yet


Projects


None yet

Milestone

No milestone

file Upload files Find File Clone or download

 Pull request

 Compare

Latest commit 9536440 11 minutes ago

2 days ago

11 minutes ago

56



# No write access: create a pull request

## 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 repository:  base:  head repository:  compare:

✓ **Able to merge.** These branches can be automatically merged.



Added time dependent salutations

Write Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ 🚩 ↶

I added a different salutation for different times of the day. It requires the datetime module.

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

☒ Allow edits from maintainers. [Learn more](#)

Reviewers

Suggestions

jenniferagraham

[Request](#)

Assignees

file

Upload files

Find File

Clone or download

Pull request

Compare

Latest commit 9536440 11 minutes ago

2 days ago

11 minutes ago

Showing 1 changed file with 9 additions and 2 deletions.

Unified Split

```
11 hello_world.py
@@ -4,6 +4,13 @@
4      @author JGraham
5      created 25/06/19
6      ...
7      + import datetime
8      + current_time = datetime.datetime.now()
7      9
8      - hello = 'hello world'
9      - print(hello)
10     +
11     + if current_time.hour < 12:
12     +     print("Good morning")
13     + elif current_time.hour < 18:
14     +     print("Good afternoon")
15     + else:
16     +     print("Good evening")
```

# Owner view:

jenniferagraham / jag\_hello\_world Private


Watch 0 Star 0 Fork 2

Code Issues 0 Pull requests 1 Projects 0 Security Insights Settings


## Added time dependent salutations #1 Edit

Open tiagoams wants to merge 1 commit into jenniferagraham:master from tiagoams:myfeat



Conversation 0 Commits 1 Checks 0 Files changed 1 +9 -2

 tiagoams commented 1 hour ago


I added a different salutation for different times of the day. It requires the datetime module.

 Added time dependent salutations 9536440

Add more commits by pushing to the **myfeat** branch on tiagoams/jag\_hello\_world.


  **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 AA B i “ < > @ 1 participant


Leave a comment


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


 Close pull request **Comment**


Reviewers   
Suggestions  
 jenniferagraham [Request](#)

Assignees   
No one—assign yourself

Labels   
None yet

Projects   
None yet

Milestone   
No milestone

Notifications [Customize](#)  
 **Subscribe**  
You're not receiving notifications from this thread.

←

→

↺

https://github.com/jenniferagraham/jag\_hello\_world/pull/1/commits/9536440b9aa74386a5fa9fbf7814ed1363ba24de

📁 jenniferagraham / jag\_hello\_world

Private

👁 Watch 0

★ Star 0

🍴 Fork 2

<> Code

! Issues 0

🔗 Pull requests 1

📁 Projects 0

🛡 Security

📊 Insights

⚙ Settings

Added time dependent salutations #1

Edit

🔗 Open

tiagoams wants to merge 1 commit into jenniferagraham:master from tiagoams:myfeat

💬 Conversation 0

🔗 Commits 1

📋 Checks 0

📄 Files changed 1

+9 -2

Changes from all commits

File filter...

Jump to...

⚙

Review changes

Added time dependent salutations

tiagoams committed 1 hour ago

11 hello\_world.py

@@ -4,6 +4,13 @@

4 @author JGraham

5 created 25/06/19

6 ...

7

8 -hello = 'hello world'

9 -print(hello)

4 @author JGraham

5 created 25/06/19

6 ...

7 +import datetime

8 +current\_time = datetime.datetime.now()

9

10 +

11 +if current\_time.hour < 12:

12 + print("Good morning")

13 +elif current\_time.hour < 18:

14 + print("Good afternoon")

15 +else:

16 + print("Good evening")

Write

Preview

AA B i “ <> 🔗 ☰ ☷ ☹ @ 📌 ↶

This is a nice change!

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

Comment

Submit general feedback without explicit approval.

Approve

Submit feedback and approve merging these changes.

Request changes

Submit feedback that must be addressed before merging.



Submit review





💡 ProTip! Use `n` and `p` to navigate between commits in a pull request.

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

59


 **Open** Added time dependent salutations #1  
tiagoams wants to merge 1 commit into `jenniferagraham:master` from `tiagoams:myfeat` 


  Added time dependent salutations 9536440  
  jenniferagraham approved these changes now [View changes](#)

jenniferagraham left a comment


This is a nice change!


Add more commits by pushing to the `myfeat` branch on `tiagoams/jag_hello_world`.



 **Changes approved** [Show all reviews](#)


1 approving review [Learn more](#).

 1 approval

 **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


AA B i “ <> 🔗 ⋮ ⋮ ✓ @ 🚩 ↶

Leave a comment


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

 Close pull request


Comment

**Assignees** 


No one—assign yourself

**Labels** 

None yet


**Projects** 

None yet

**Milestone** 



No milestone


**Notifications** [Customize](#)

 Unsubscribe

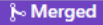

You're receiving notifications because you commented.


2 participants


 


 Lock conversation


💡 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

 **Merged** jenniferagraham merged 1 commit into `jenniferagraham:master` from `tiagoams:myfeat`  now

 Conversation **1**


 Commits **1**

 Checks **0**



 Files changed **1**

+9


-2




tiagoams commented 1 hour ago


+  


I added a different salutation for different times of the day. It requires the datetime module.



 Added time dependent salutations



9536440




 jenniferagraham approved these changes 1 minute ago


[View changes](#)

jenniferagraham left a comment


+  

This is a nice change!



 jenniferagraham merged commit 2883dfa into `jenniferagraham:master` now


[Revert](#)



**Pull request successfully merged and closed**

You're all set—the `tiagoams:myfeat` branch can be safely deleted.

[Delete branch](#)



Write

Preview


AA


B


i


“

<>












@







Leave a comment

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



[Comment](#)

Reviewers

 jenniferagraham 

Assignees

No one—assign yourself

Labels

None yet

Projects


None yet

Milestone

No milestone



Notifications


Customize


 [Unsubscribe](#)


You're receiving notifications because you modified the open/close state.

2 participants

 Lock conversation

 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

61

# Your turn:

## Option 1 – Shared repository

- Pair up and find each other's test repositories
- Add write access to colleague on test repository
- Clone, create branch, make a commit and push changes
- Check if changes were made in the original repo whilst you were editing (there might!) 😈
  - Pull changes and merge to new branch
  - Push to shared repo

## Option 2 – Fork Jenny's jag\_hello\_world

[https://github.com/jenniferagraham/jag\\_hello\\_world.git](https://github.com/jenniferagraham/jag_hello_world.git)

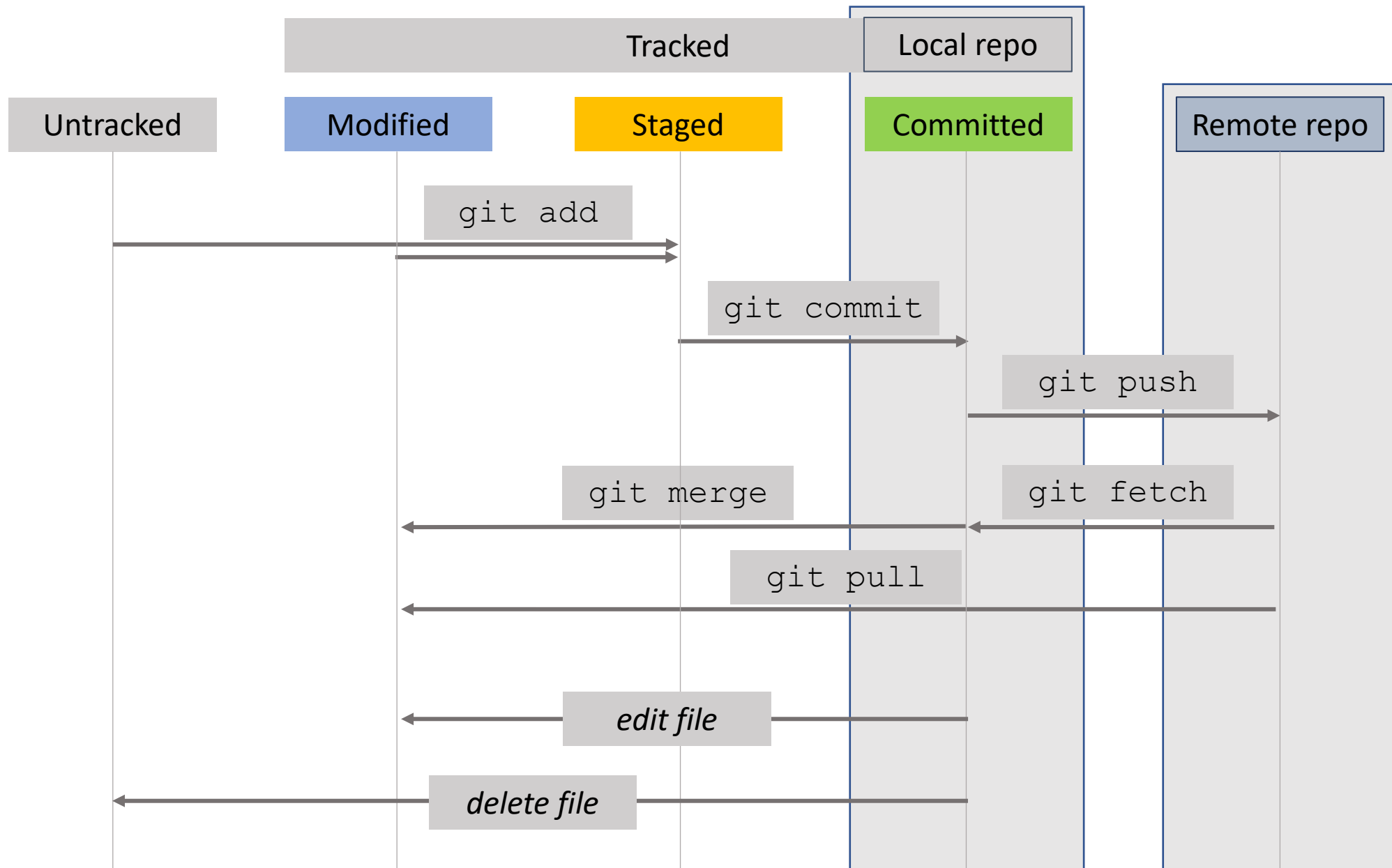
- Clone your own forked copy to your PC
- Create branch, make a commit and push to your remote
- Make a pull request on GitHub



# Please provide feedback!

Anonymous comments can be provided here:

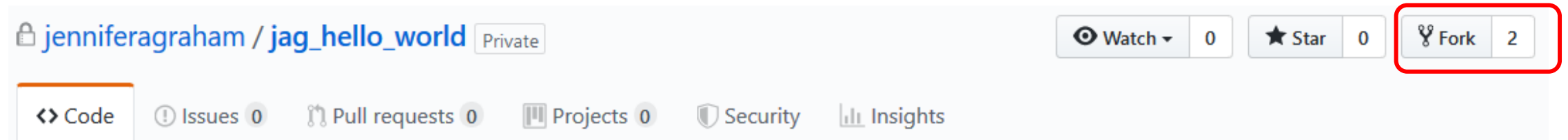
<https://forms.gle/uaCPkrVZHngmHXQp9>



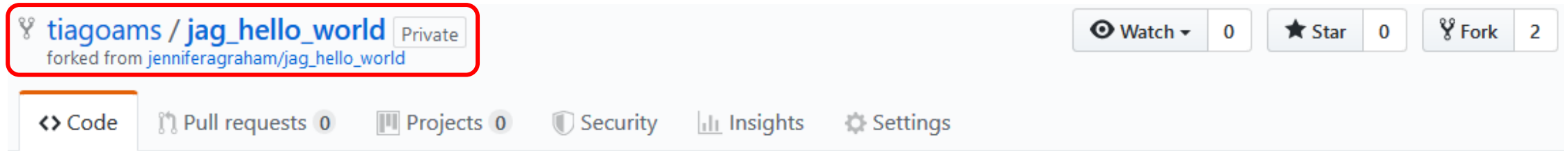


# Fork workflow: step by step...

## 1. Fork other user's repo on GitHub

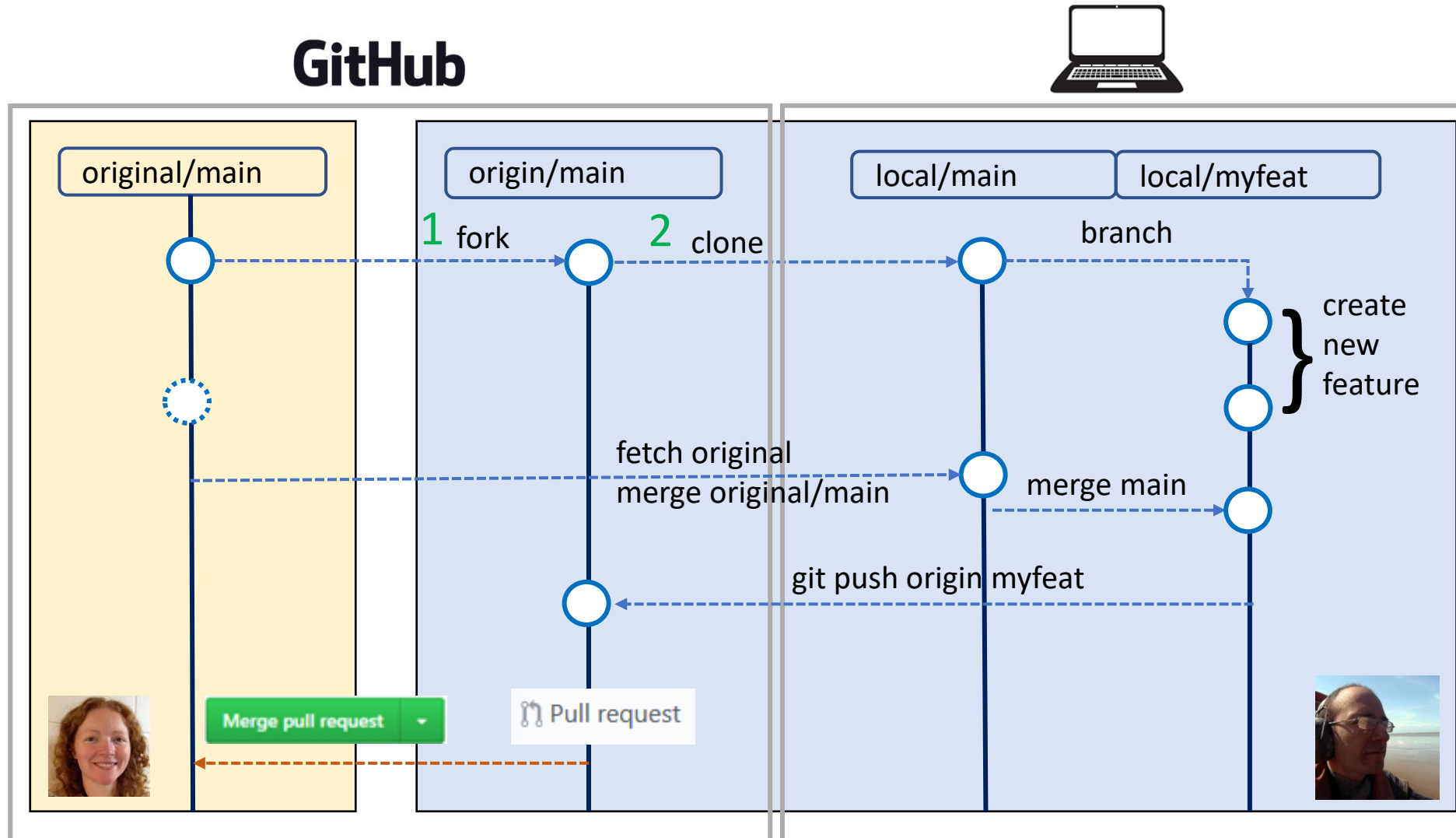


This is how it looks in your account



```
$ git clone git@github.com:tiagoams/jag_hello_world.git
```

# Fork workflow: no privileged access required



# Fork workflow: working locally

## hello\_world.py

```
'''
My first code.

@author JGraham
created 25/06/19
'''

hello = 'hello world'
print(hello)
```

## 3. Create new local branch

```
checkout -b <branch> = branch <branch>
                     checkout <branch>
```

## 4. Modify and commit

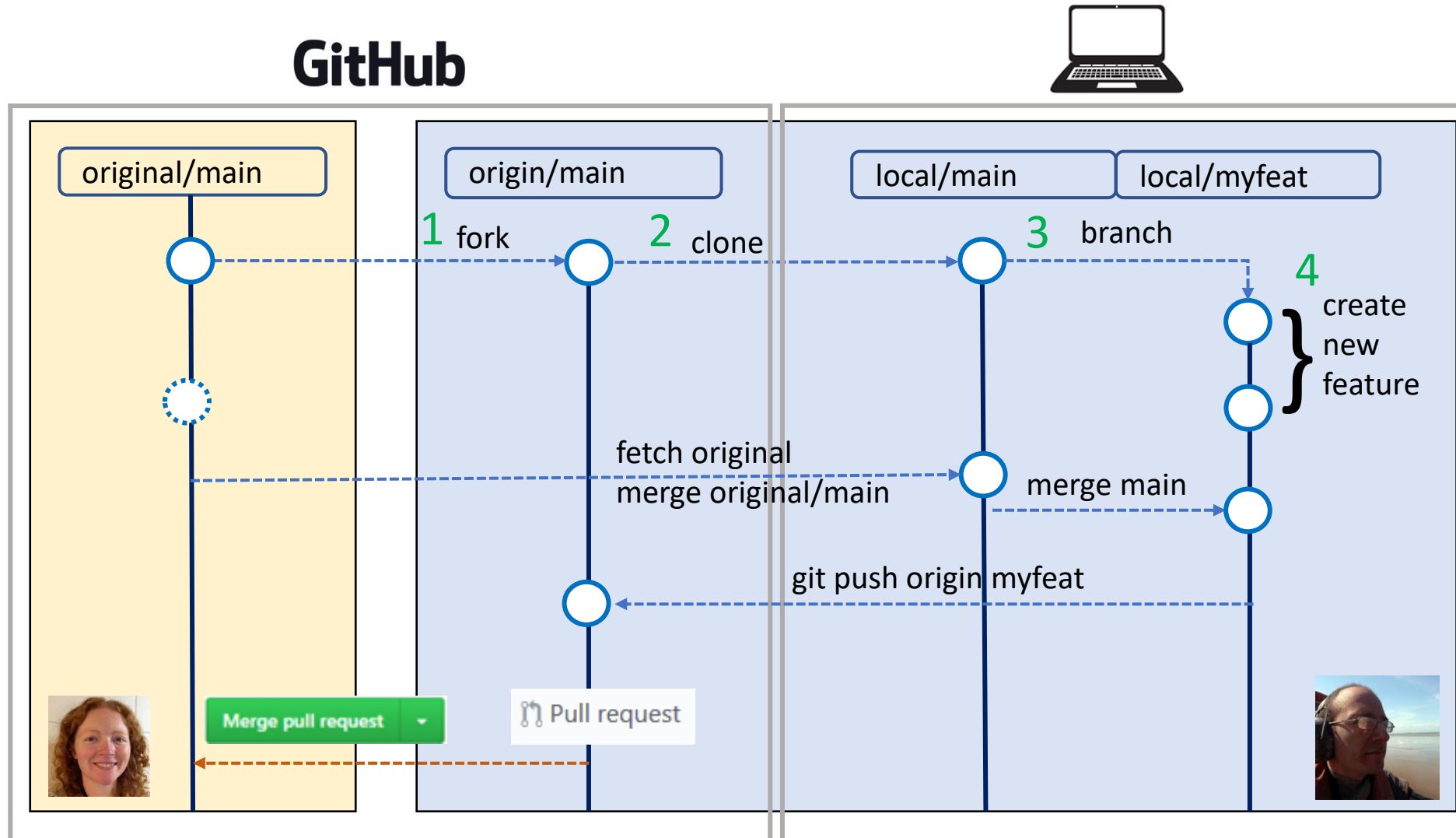
```
commit -a -m <msg> = add *
                  commit -m <msg>
```

```
$ git checkout -b myfeat
Switched to a new branch 'myfeat'
$ cat hello_world.py
(...)
import datetime
current_time = datetime.datetime.now()

if current_time.hour < 12:
    print("Good morning")
elif current_time.hour < 18:
    print("Good afternoon")
else:
    print("Good evening")

$ git commit -am "Added time dependent salutations"
[myfeat 9536440] Added time dependent salutations
1 file changed, 9 insertions(+), 2 deletions(-)
```

# Fork workflow: no privileged access required



# Fork workflow: check for changes since fork

5. Add a new remote *original*

6. Fetch and merge to local main branch

7. Merge/rebase to *myfeat* branch

```
$ git remote add original git@github.com:jenniferagraham/jag_hello_world.git
$ git remote -v
origin    git@github.com:tiagoams/jag_hello_world.git (fetch)
origin    git@github.com:tiagoams/jag_hello_world.git (push)
original  git@github.com:jenniferagraham/jag_hello_world.git (fetch)
original  git@github.com:jenniferagraham/jag_hello_world.git (push)

$ git fetch original
From github.com:jenniferagraham/jag_hello_world
 * [new branch]      main      -> original/main

$ git checkout main
Switched to branch 'main'

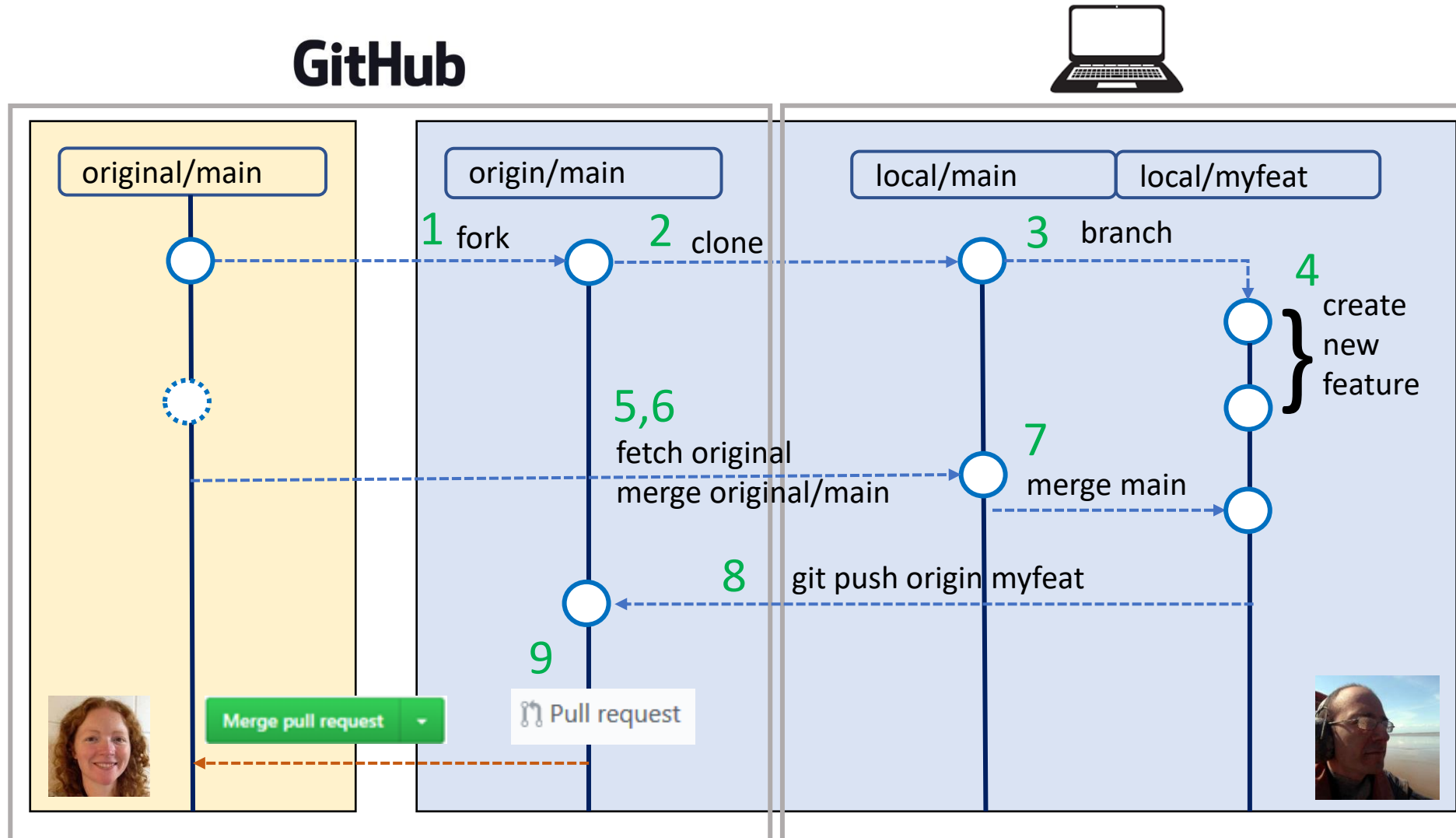
$ git merge original/main
Already up to date.

$ git checkout myfeat
Switched to branch 'myfeat'

$ git rebase main
Current branch myfeat is up to date.

$
```

# Fork workflow: no privileged access required



# Fork workflow: push to *origin*

## 8. Push changes to your GitHub repo

```
$ git push origin myfeat
Counting objects: 3, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 447 bytes | 223.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'myfeat' on GitHub by visiting:
remote:
https://github.com/tiagoams/jag_hello_world/pull/new/myfeat
remote:
To github.com:tiagoams/jag_hello_world.git
 * [new branch]      myfeat -> myfeat
$
```

# Fork workflow: 9. create a pull request

## 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 repository:  base:  head repository:  compare:

✓ **Able to merge.** These branches can be automatically merged.



Added time dependent salutations

Write Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ 🚩 ↶

I added a different salutation for different times of the day. It requires the datetime module.

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

☒ Allow edits from maintainers. [Learn more](#)

Reviewers

Suggestions

jenniferagraham

[Request](#)

Assignees

file

Upload files

Find File

Clone or download

Pull request

Compare

Latest commit 9536440 11 minutes ago

2 days ago

11 minutes ago

Showing 1 changed file with 9 additions and 2 deletions.

Unified Split

```
11 hello_world.py
@@ -4,6 +4,13 @@
4      @author JGraham
5      created 25/06/19
6      ...
7      + import datetime
8      + current_time = datetime.datetime.now()
7      9
8      - hello = 'hello world'
9      - print(hello)
10     +
11     + if current_time.hour < 12:
12     +     print("Good morning")
13     + elif current_time.hour < 18:
14     +     print("Good afternoon")
15     + else:
16     +     print("Good evening")
```



# Fork workflow: the owners view

jenniferagraham / jag\_hello\_world Private Watch 0 Star 0 Fork 2

<> Code Issues 0 Pull requests 1 Projects 0 Security Insights Settings

# Added time dependent salutations #1 Edit

Open tiagoams wants to merge 1 commit into jenniferagraham:master from tiagoams:myfeat

Conversation 0 Commits 1 Checks 0 Files changed 1 +9 -2

tiagoams commented 1 hour ago

I added a different salutation for different times of the day. It requires the datetime module.

Added time dependent salutations 9536440

Add more commits by pushing to the **myfeat** branch on tiagoams/jag\_hello\_world.

**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

AA B i “ <> 🔗 ☰ ☷ ✎ @ 📌 ↶

Leave a comment

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

Close pull request **Comment**

Reviewers

Suggestions

jenniferagraham [Request](#)

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications [Customize](#)

Subscribe

You're not receiving notifications from this thread.

1 participant

←

→

↺

https://github.com/jenniferagraham/jag\_hello\_world/pull/1/commits/9536440b9aa74386a5fa9fbf7814ed1363ba24de

📁 jenniferagraham / jag\_hello\_world

Private

👁 Watch 0

★ Star 0

🍴 Fork 2

<> Code

! Issues 0

🔗 Pull requests 1

📁 Projects 0

🛡 Security

📊 Insights

⚙ Settings

Added time dependent salutations #1

Edit

🔗 Open

tiagoams wants to merge 1 commit into jenniferagraham:master from tiagoams:myfeat

💬 Conversation 0

🔗 Commits 1

📋 Checks 0

📄 Files changed 1

+9 -2

Changes from all commits

File filter...

Jump to...

⚙

Review changes

Added time dependent salutations

tiagoams committed 1 hour ago

11 hello\_world.py

@@ -4,6 +4,13 @@

4 @author JGraham

5 created 25/06/19

6 ...

7

8 -hello = 'hello world'

9 -print(hello)

4 @author JGraham

5 created 25/06/19

6 ...

7 +import datetime

8 +current\_time = datetime.datetime.now()

9

10 +

11 +if current\_time.hour < 12:

12 + print("Good morning")

13 +elif current\_time.hour < 18:

14 + print("Good afternoon")

15 +else:

16 + print("Good evening")

💡 ProTip! Use `n` and `p` to navigate between commits in a pull request.

Write

Preview

AA B i “ <> 🔗 ☰ ☷ ☹ @ 📌 ↶

This is a nice change!

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

Comment

Submit general feedback without explicit approval.

Approve

Submit feedback and approve merging these changes.

Request changes

Submit feedback that must be addressed before merging.

Submit review



© 2019 GitHub, Inc. Terms Privacy Security Status Help




🐙

Contact GitHub Pricing API Training Blog About

75

13:09


 **Open** Added time dependent salutations #1  
tiagoams wants to merge 1 commit into `jenniferagraham:master` from `tiagoams:myfeat` 


 Added time dependent salutations 9536440  
  jenniferagraham approved these changes now [View changes](#)

jenniferagraham left a comment


This is a nice change!


Add more commits by pushing to the `myfeat` branch on `tiagoams/jag_hello_world`.



 **Changes approved** [Show all reviews](#)


1 approving review [Learn more](#).

 1 approval

 **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


AA B i “ <> 🔗 ⋮ ⋮ ✓ @ 🚩 ↶


Leave a comment


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


 Close pull request

Comment


Assignees   
No one—assign yourself

Labels   
None yet



Projects   
None yet


Milestone   
No milestone

Notifications [Customize](#)  

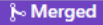

 Unsubscribe

You're receiving notifications because you commented.

2 participants  
 

 Lock conversation

💡 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.


 **Merged** jenniferagraham merged 1 commit into `jenniferagraham:master` from `tiagoams:myfeat`  now


🗨 Conversation **1**

🔗 Commits **1**

📋 Checks **0**

📄 Files changed **1**


+9 -2 





tiagoams commented 1 hour ago

+ 😊 ...

I added a different salutation for different times of the day. It requires the datetime module.

  Added time dependent salutations 9536440





 jenniferagraham approved these changes 1 minute ago

[View changes](#)


jenniferagraham left a comment

+ 😊 ...

This is a nice change!

  jenniferagraham merged commit 2883dfa into `jenniferagraham:master` now


[Revert](#)



**Pull request successfully merged and closed**

You're all set—the `tiagoams:myfeat` branch can be safely deleted.


[Delete branch](#)



Write Preview



AA B i “ <> 🔗 ⋮ ⋮ ✓ @ 📌 ↶

Leave a comment

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

[Comment](#)

Reviewers

 jenniferagraham 

Assignees

No one—assign yourself

Labels

None yet


Projects

None yet

Milestone



No milestone


Notifications [Customize](#)


 [Unsubscribe](#)

You're receiving notifications because you modified the open/close state.

2 participants

 [Lock conversation](#)

 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.