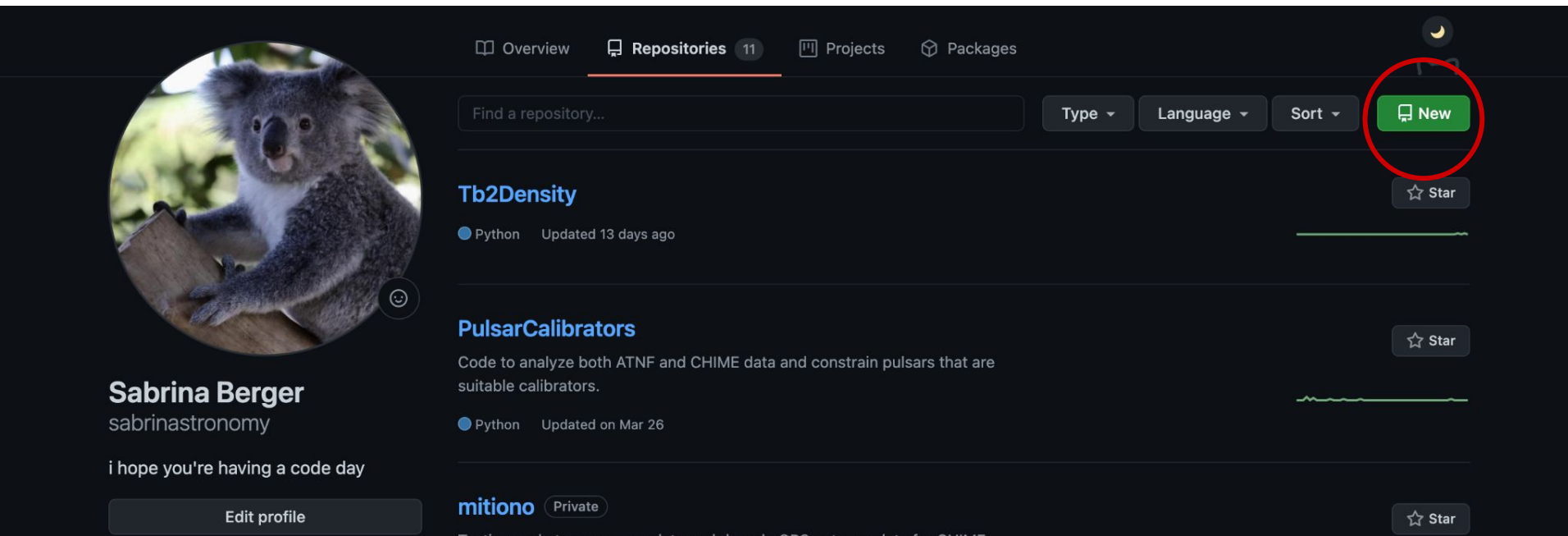


Brief Git Tutorial

Astrophysics CODE CLUB
University of Melbourne

How to make a remote git repository*

1) Click the green new button to create a new repository.



The screenshot shows the GitHub profile of Sabrina Berger (sabrinastronomy). The 'Repositories' tab is active, displaying a list of repositories. A red circle highlights the 'New' button in the top right corner of the repository list.

Sabrina Berger
sabrinastronomy
i hope you're having a code day
Edit profile

Tb2Density
Python Updated 13 days ago
Star

PulsarCalibrators
Code to analyze both ATNF and CHIME data and constrain pulsars that are suitable calibrators.
Python Updated on Mar 26
Star

mitiono Private
Testing code to cross-correlate and deconvolve GPS antenna data for CHIME
Star

*Note that we can also create a git repository locally (*git init*), which creates a **./git** folder locally. We can then push to GitHub later

2) Follow the instructions to create a repository (don't worry about any of the extra features for now).

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

 sabrinastronomy ▾

Repository name *

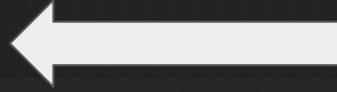
Great repository names are short and memorable. Need inspiration? How about **solid-octo-carnival?**

Description (optional)

3) Now we want to *git clone* the repository we just made on to our local machines (wherever we'd like!). This will copy over all the git history from the online project (**./git** folder!)

- **Note:** this makes a **remote branch** called **main** which is attached to the cloud, this can be a confusing detail if you accidentally make a local branch and then try and merge it into origin

```
Last login: Sun Jul 30 12:16:21 on console
(base) [ 2:52PM ] [ sabrinaberger@sabrinaastronomy:~ ]
$ git clone https://github.com/sabrinaastronomy/CodeClub.git
Cloning into 'CodeClub'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 10 (delta 1), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (10/10), done.
```



4) Try *cd*-ing into your new repository and trying *git status*

```
(base) [ 2:56PM ] [ sabrinaberger@sabrinaastronomy:~ ]  
$ cd CodeClub  
(base) [ 2:56PM ] [ sabrinaberger@sabrinaastronomy:~/CodeClub(main✓) ]  
$ ls  
LICENSE  README.md  Resources  
(base) [ 2:56PM ] [ sabrinaberger@sabrinaastronomy:~/CodeClub(main✓) ]  
$ git status  
On branch main  
Your branch is up to date with 'origin/main'.
```

Important basic git commands

- **git --help**
 - Gives a helpful list of git commands! Can also do `git {command} --help` for specifics on commands
- **git config --global user.name "{username}"**
- **git config --global user.email {email}**
 - **Note:** these are *git* associated names/emails, so they don't have much to do with GitHub! Just useful for identifying yourself in local git projects
- **git status**
 - Shows you current changes
- **git log**
 - Shows you the commit history for your project
- **git fetch**
 - Updates information stored in the local **./git** folder, such as new remote branches

How to make your first commit

- The most basic git workflow consists of three important steps:
 - a. Use ``git add {file}`` to “add” a new file, or to “add” changes to an already existing file
 - You can use “wildcard” operators with this! E.g. ``git add *.py`` to add .py file changes
 - b. Once you’re happy with your additions, use ``git commit -m “{useful message}”`` to add a commit with a helpful commit message explaining your changes
 - c. Finally, we need to push our changes to the cloud. We’ll do this with the command ``git push origin main``
 - Note that the *origin* here specifies the remote cloud, and *main* is the branch we’re committing to. By default, GitHub repos start with only a main branch

How to pull from remote

- To pull in any changes from collaborators, just use `git pull` in the relevant directory
 - This won't do anything for us now, since we just made this repo for ourselves... But it's VERY important to pull the most recent version of the repository before you start making changes!
 - Ideally if everyone was working on their own branches and being responsible about workflow, this wouldn't be an issue... But nobody's perfect :)

Brief Git Workflow Summary

- Step 0: PULL changes that might have been made by collaborators
- Step 1: ADD our changes
- Step 2: COMMIT changes with a message (-m)
- Step 3: PUSH changes to the cloud
- **NOTE:** all the while we can check the STATUS of our additions!

How to make a pull request*

1) When you're working in a collaboration, and you're ready to incorporate your changes into the master branch, you can make a pull request! Fork the repository that you want to add something to.

The screenshot shows the GitHub interface for the repository 'astrobenji / CodeClub'. The repository is public and has 1 watch, 2 forks, and 2 stars. The 'Fork' button is circled in red. Below the repository name, there are buttons for 'Go to file', 'Add file', and 'Code'. The file list shows a commit by 'benmetha' with 5 commits, and a table of files including 'Resources', 'Session1', '.gitignore', 'LICENSE', and 'README.md'. The 'About' section on the right describes the repository as a place for students to show their code and discuss new approaches.

astrobenji / CodeClub

Type to search

Code Issues Pull requests Actions Projects Security Insights

CodeClub Public

Watch 1 Fork 2 Star 2

main 1 branch 0 tags

Go to file Add file Code

benmetha put up code d6d2246 22 minutes ago 5 commits

Resources	Rename resources folder resources	4 hours ago
Session1	put up code	22 minutes ago
.gitignore	Add a first attempt; and slides!	54 minutes ago
LICENSE	Initial commit	2 weeks ago
README.md	Initial commit	2 weeks ago

About

A place to encourage students to show each other their code, and discuss new approaches. All code welcome! :D

Readme

GPL-3.0 license

Activity

2 stars

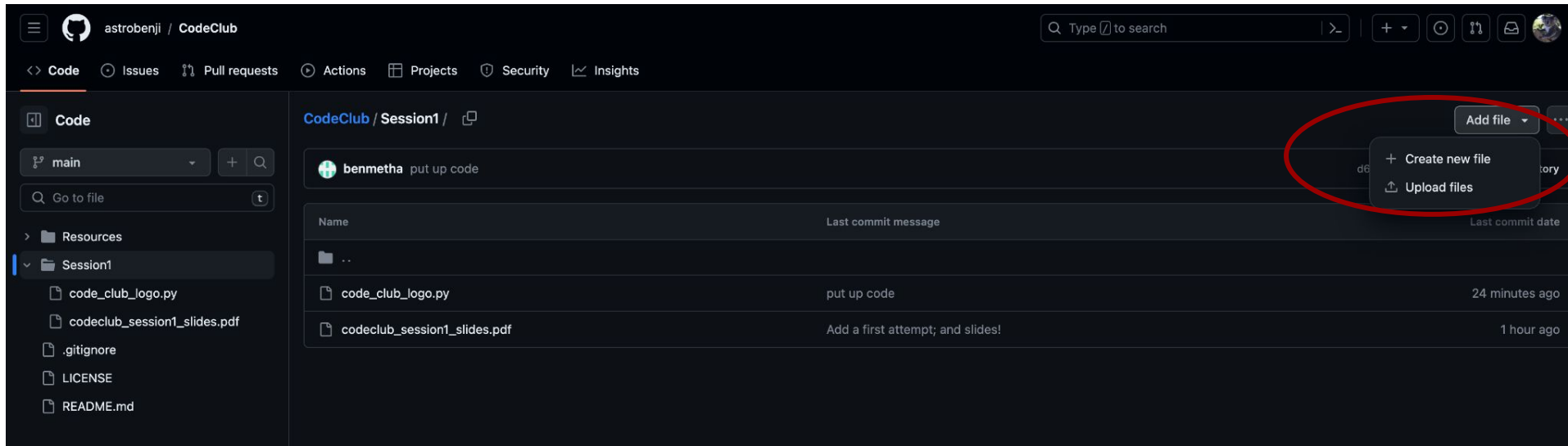
1 watching

2 forks

Report repository

How to make a pull request*

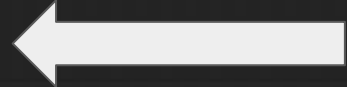
- 1) *Note that we can also make pull requests by editing files directly within the repository. This is easier, but we'll also see how to do it the other way.



How to make a pull request

2) Clone your forked version of the repository.

```
Last login: Sun Jul 30 12:16:21 on console
(base) [ 2:52PM ] [ sabrinaberger@sabrinaastronomy:~ ]
[ $ git clone https://github.com/sabrinaastronomy/CodeClub.git
Cloning into 'CodeClub'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 10 (delta 1), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (10/10), done.
```



How to make a pull request

3) Make your necessary changes, then add, commit, and push!

Changes

adding

committing

Push to your forked repo

```
(base) [ 3:10PM ] [ sabrinaberger@sabrinastronomy:~/CodeClub(main✓) ]
$ ls
LICENSE  README.md  Resources
(base) [ 3:10PM ] [ sabrinaberger@sabrinastronomy:~/CodeClub(main✓) ]
$ mv ~/Desktop/git_tutorial.pdf ./
(base) [ 3:10PM ] [ sabrinaberger@sabrinastronomy:~/CodeClub(mainx) ]
$ ls
LICENSE      README.md    Resources    git_tutorial.pdf
(base) [ 3:10PM ] [ sabrinaberger@sabrinastronomy:~/CodeClub(mainx) ]
$ git add .
(base) [ 3:11PM ] [ sabrinaberger@sabrinastronomy:~/CodeClub(mainx) ]
$ git commit -m "brief git tutorial"
[main 47e7de4] brief git tutorial
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 git_tutorial.pdf
(base) [ 3:11PM ] [ sabrinaberger@sabrinastronomy:~/CodeClub(main✓) ]
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 370.04 KiB | 18.50 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sabrinastronomy/CodeClub.git
8ca12c1..47e7de4  main -> main
```

How to make a pull request

4) Go back to the originally repository and click create pull request.

The screenshot shows a GitHub pull request interface. At the top, two white arrows point towards the repository selection dropdowns. The left arrow is labeled "Original repo" and points to the "base repository: astrobenji/CodeClub" dropdown. The right arrow is labeled "Your forked updated repo" and points to the "head repository: sabrinaastronomy/CodeClub" dropdown. Below these, a green status bar indicates "✓ Able to merge. These branches can be automatically merged." In the center, a dark blue box contains the text "Discuss and review the changes in this comparison with others. [Learn about pull requests](#)". To the right of this box, a green button labeled "Create pull request" is circled in red. Below this, a summary bar shows "1 commit", "1 file changed", and "1 contributor". Further down, a commit list shows a commit titled "brief git tutorial" by "sabrinaastronomy" committed 11 minutes ago. At the bottom, a file diff section shows a binary file "git_tutorial.pdf" with a size change of "+468 KB". The text "Binary file not shown." is visible below the file name. In the bottom right corner, there are "Split" and "Unified" view toggle buttons.

Original repo

base repository: astrobenji/CodeClub

base: main

head repository: sabrinaastronomy/CodeClub

compare: main

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

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

1 commit

1 file changed

1 contributor

Commits on Aug 3, 2023

brief git tutorial

sabrinaastronomy committed 11 minutes ago

47e7de4

Showing 1 changed file with 0 additions and 0 deletions.

Split Unified

BIN +468 KB git_tutorial.pdf

Binary file not shown.

How to make a pull request

5) Comment on what you added with your pull request and for it to be approved! 🕒

The screenshot shows a GitHub pull request page for the repository 'astrobenji / CodeClub'. The pull request is titled 'brief git tutorial #1' and is created by 'sabrinastronomy'. It shows a merge of 1 commit into 'astrobenji:main' from 'sabrinastronomy:main'. The interface includes a navigation bar with tabs for Code, Issues, Pull requests (1), Actions, Projects, Security, and Insights. Below the title, there are statistics for Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from 'sabrinastronomy' is visible, stating 'making a tutorial about a pull request by making a pull request'. A green checkmark indicates 'This branch has no conflicts with the base branch'. The right sidebar shows sections for Reviewers, Assignees, Labels, Projects, Milestone, Development, and Notifications. At the bottom, there are buttons for 'Close pull request' and 'Comment'.

astrobenji / CodeClub

Q Type to search

<> Code

brief git tutorial #1

Open sabrinastronomy wants to merge 1 commit into astrobenji:main from sabrinastronomy:main

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

sabrinastronomy commented now

making a tutorial about a pull request by making a pull request

brief git tutorial 47e7de4

Add more commits by pushing to the main branch on sabrinastronomy/CodeClub.

This branch has no conflicts with the base branch
Only those with write access to this repository can merge pull requests.

Write Preview

Leave a comment

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

Close pull request Comment

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.

None yet

Notifications
Customize

Brief Pull Request Summary

- Pull requests are a super useful way of keeping major code changes organized
- Rule of thumb: **main** branch should ALWAYS be deployable
 - This is why pull requests exist: typically if you're a part of a collaboration, there will be other people working on the code base with you. Usually there'll be one/a few people who manage most of a given repository, and making a pull request allows you to give them a chance to view your code, review it, suggest changes, and then finally accept the merge into master once it's deemed ready
- Workflow goes something like
 - Propose a new feature
 - Checkout a new branch to start working on your feature (make sure nobody interferes with your work so you don't get merge conflicts)
 - Keep pushing changes to your branch until things are stable/finished, then make a pull request

How to go back to an old commit when your new code breaks

- You can get a log of all previous commits with
 - *git log*
- This should return your previous commits along with their corresponding hash, e.g.,

```
(base) sabrinaberger@sabrinaastronomy summerschool % git log  
commit 5d29d753054e787005f0202364e286d1211032db (HEAD)
```

- You can revert to a previous commit with
 - *git checkout <commit hash>* (where commit hash is that long green string after the commit above)

Resources

- [Amazing git branching tutorial](#)
- [How to install git on any OS](#)
- [A nice ELI5 git series](#)
- ["What is git" from Atlassian](#)
- [Basic git tutorial](#)
- [An in-depth summary of remote branches](#)
- [Tutorial on how to deal with merge conflicts](#)