



GitHub Cheatsheet

TOC

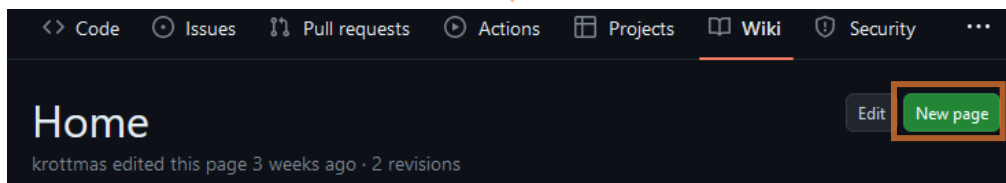
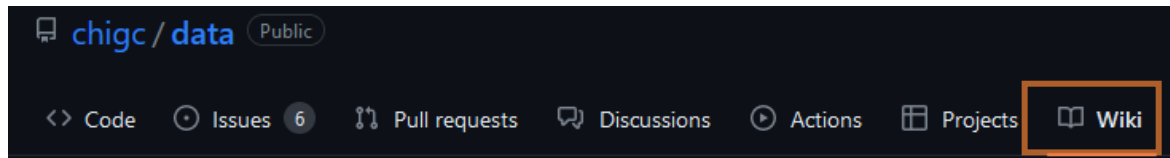
GitHub Cheatsheet	1
GitHub Wiki	2
Create a Wiki page.....	2
Edit a Wiki page	4
GitHub Issues	4
Create an Issue	4
Contribute in an issue.....	8
GitHub Basic Workflow	10
Set up GitHub Desktop	10
Clone (needed) repositories.....	11
Get (pull) changes from the GitHub repository.....	13
Add (Push) changes to the GitHub Repository.....	13
Tips	14

GitHub Wiki

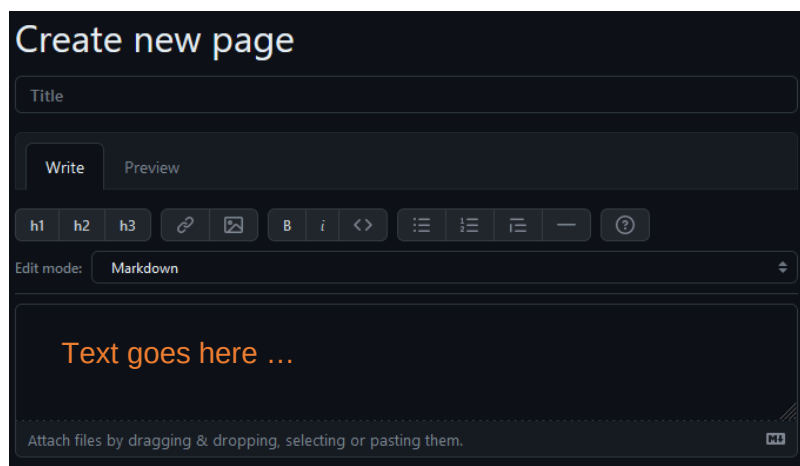


Create a Wiki page

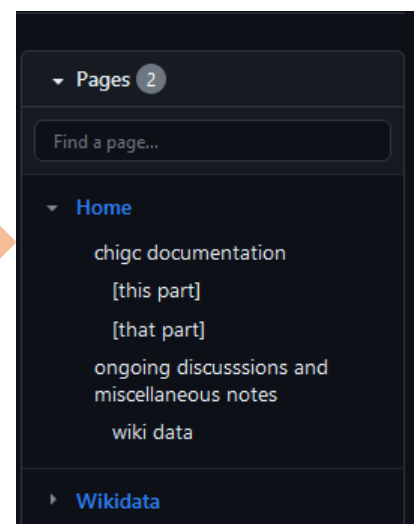
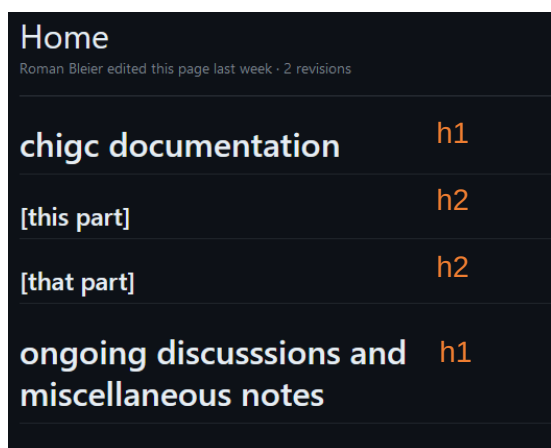
- 1) Navigate to **Wiki** in the repository *data* & click on **New page**




- 2) Give your wiki page a **title** and fill in your **documentation**



- Add headings (h1, h2, h3) >> those structure your text in different segments

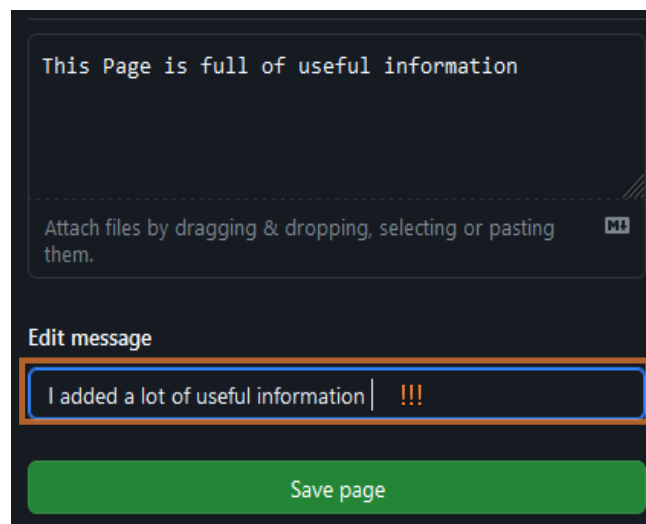


- You can also style your text (bold, italic, quotes, code etc.),
- add images (either by clicking on the upload image icon  providing a URL to an image on the web or in the repository, or by dragging and dropping an image file in the textbox,
- and add links (also to other wiki pages >> [[Link Text|Wikipagename]])



3) After entering your documentation click  and the page is ready to go

- Additionally, you can add a short message under *edit message* explaining what you changed; this becomes especially handy when editing already existing wiki pages.

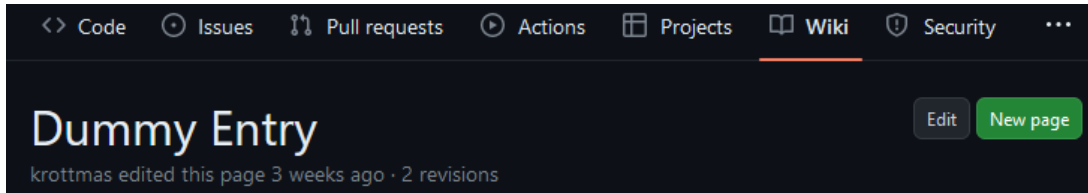




Edit a Wiki page

- 1) Navigate to an existing wiki page in the Wiki and click

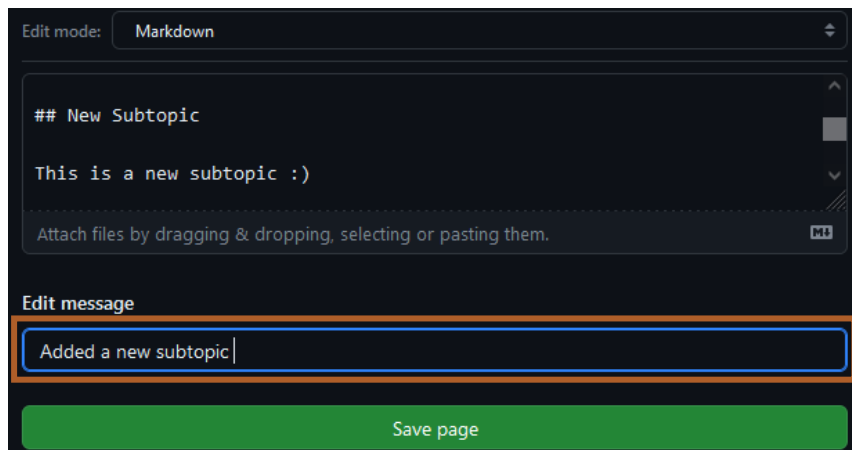
Edit



- 2) Edit the entry, fill in an edit message with a short description of your changes and click on

Save page

The wiki page is now updated!



GitHub Issues



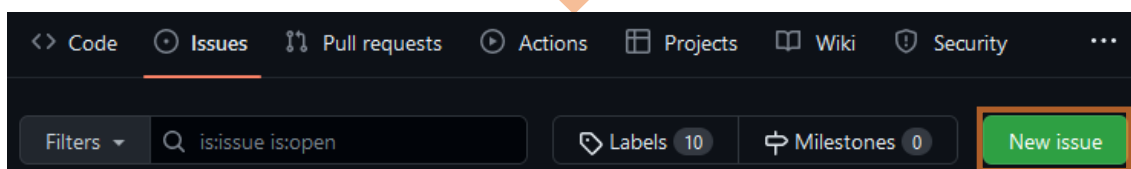
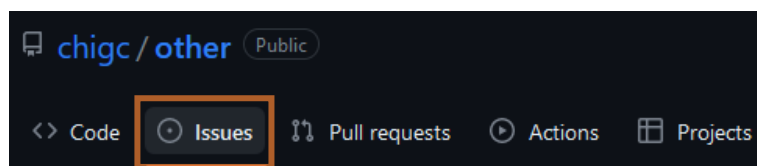
Create an Issue

- 1) Choose a repository (data for data related discussions; software for software related discussions), navigate to

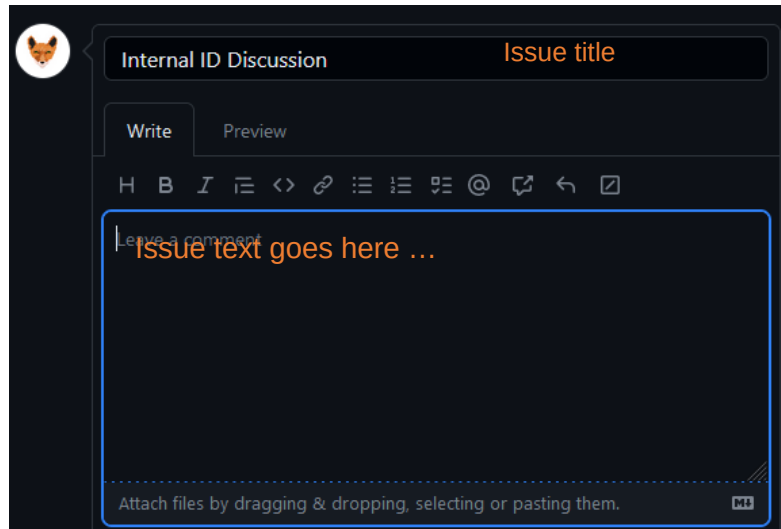
Issues

and click on

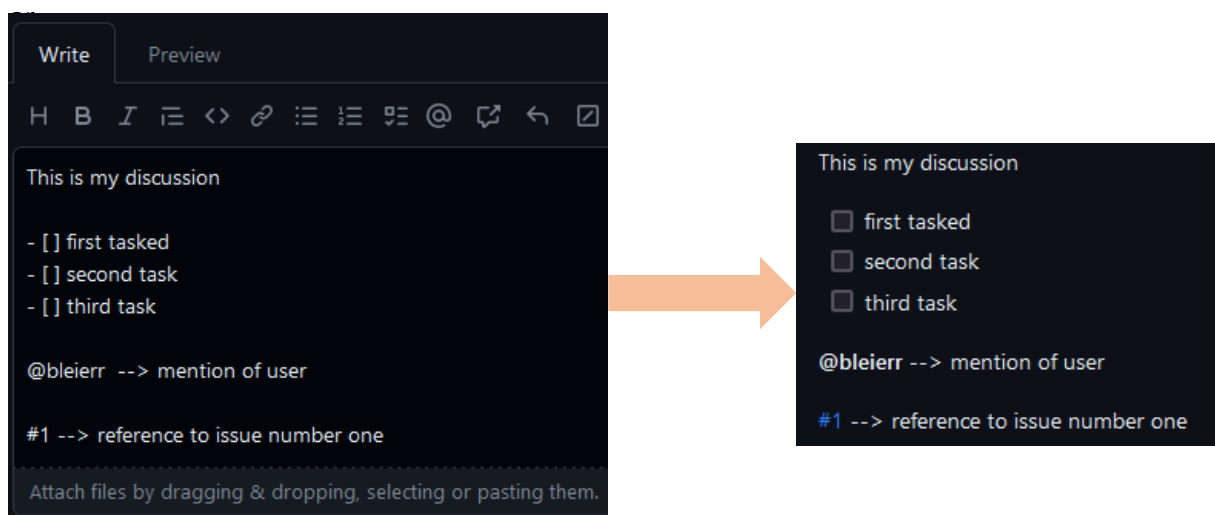
New issue




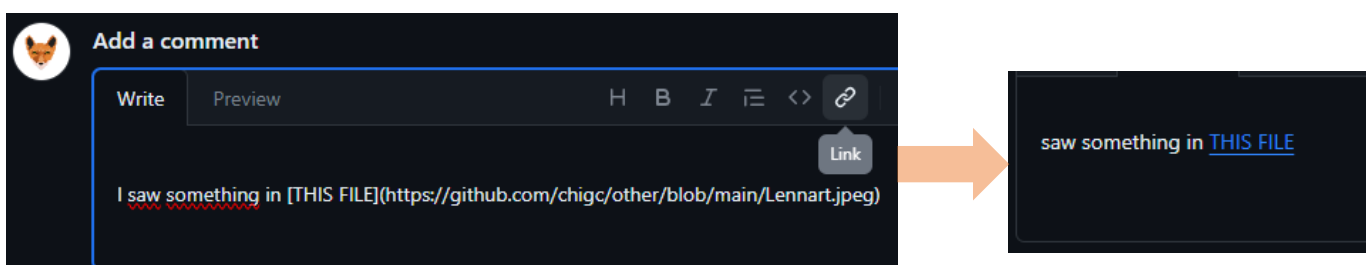
- 2) Give the issue an **expressive title** and **write your issue/discussion topic/question**



- You can also **set up a task list** (with “- [] text”), **mention other team members** (with “@username”) or **reference to other issues** (with “#issuenummer”).

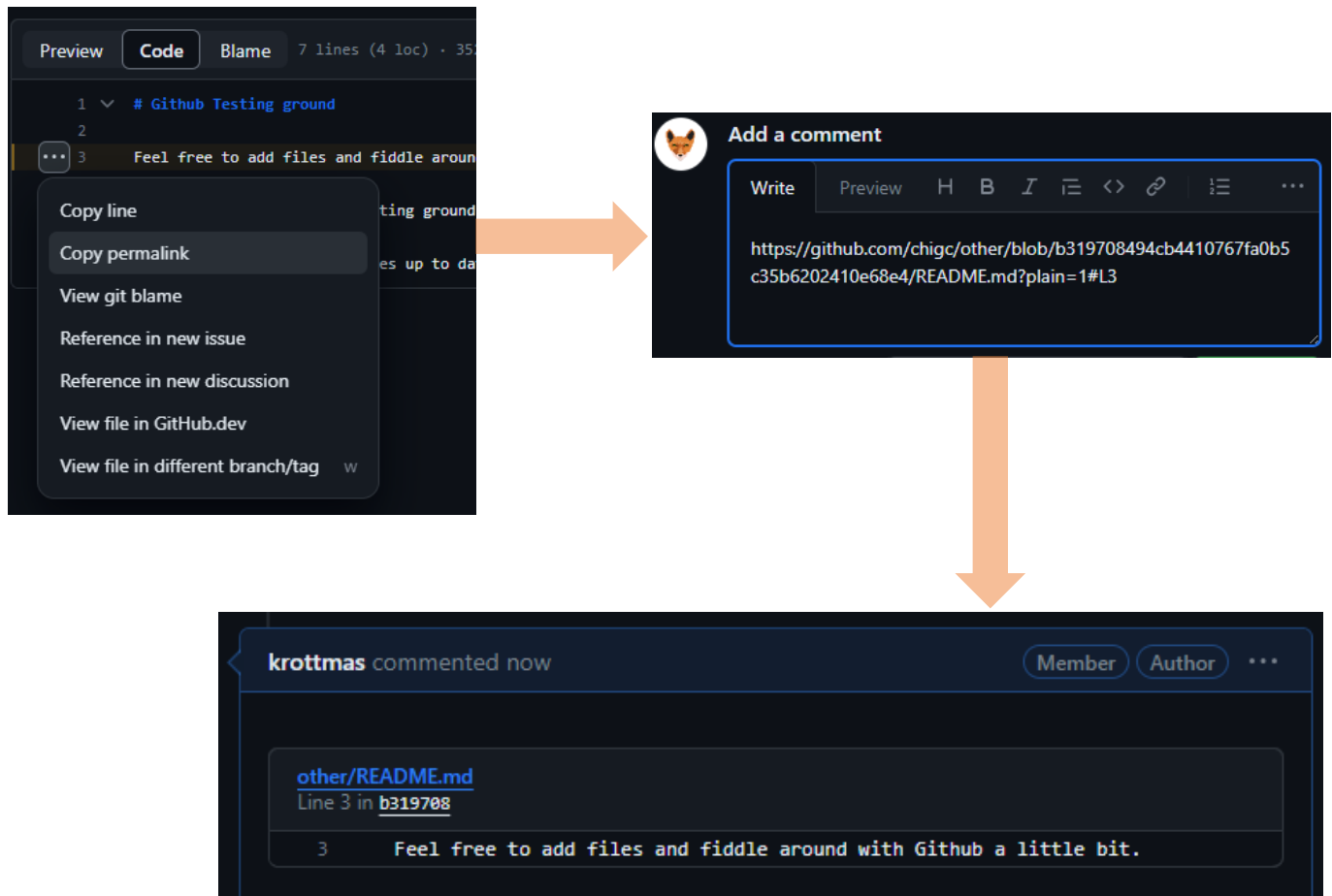


- And you can create a **reference to a code file in the repository** or **reference a part of a code file**.
 - a) **Reference a file:** press on `< > Code` and then on the file you want to mention in the issue. Copy the URL and navigate back to the issue you want to mention the file. Press on the link button  and fill in the brackets >> [Name of file/link name] (Copied URL)

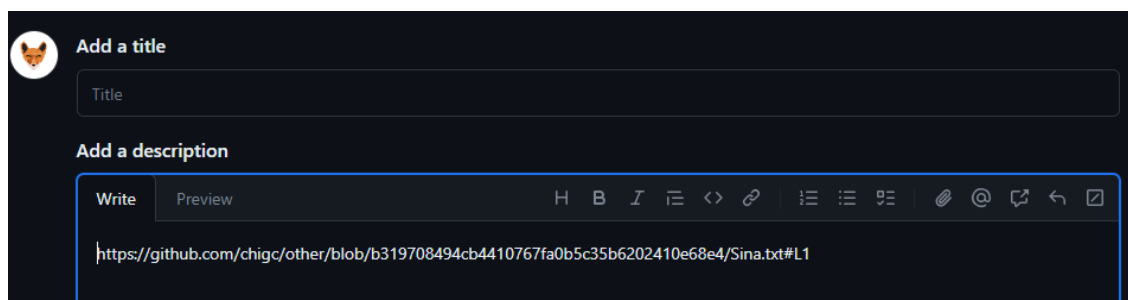


b) **Reference part of code file:** press on `< > Code` and then on the file with the part of code you want to reference. Select the line you want to reference and click on `...`.

For references in existing issues select **Copy Permalink**, head back to the issue and paste in the copied permalink. This will create a small snapshot of the code snippet you want to reference in the issue:



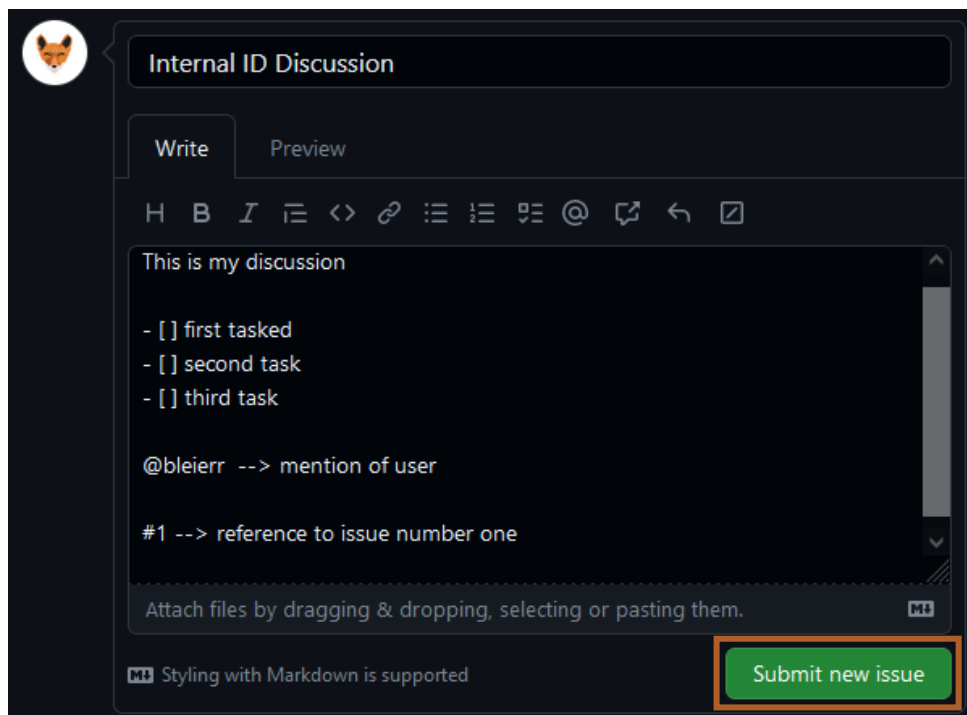
- If you click on **Reference in New Issue**, a new issue will be opened and the URL will already be filled into the description box.



- you can also assign a team member to an issue or label them



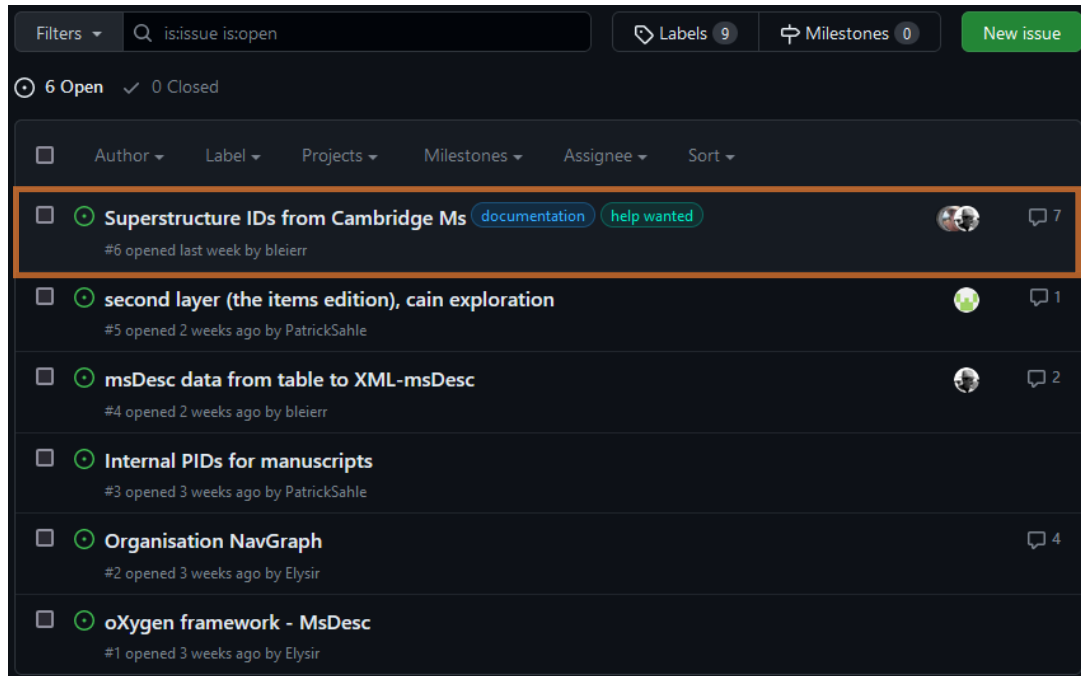
3) Click on **Submit new issue** and start the discussion



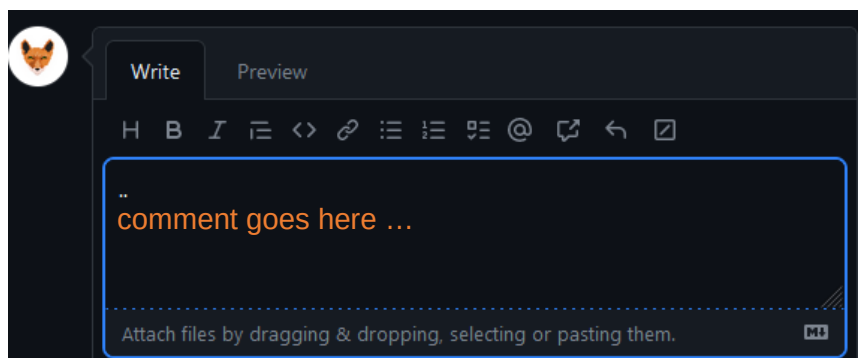


Contribute in an issue

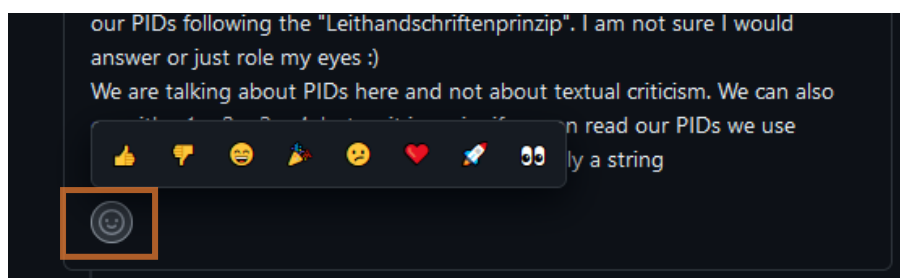
1) Navigate to  and click on the issue you want to contribute to.




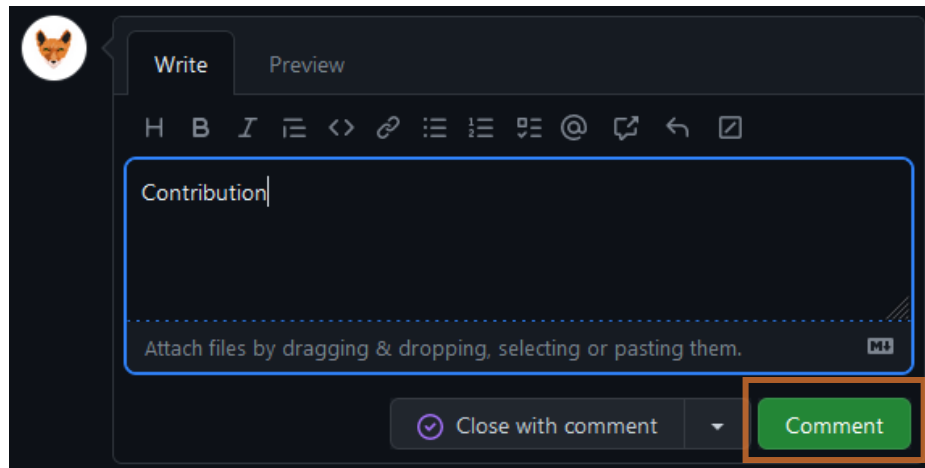
2) Navigate to the **bottom of the discussion** to write your comment.



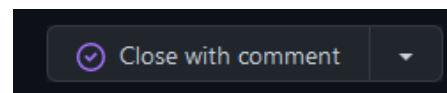
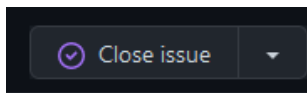
- You can also react to comments from others



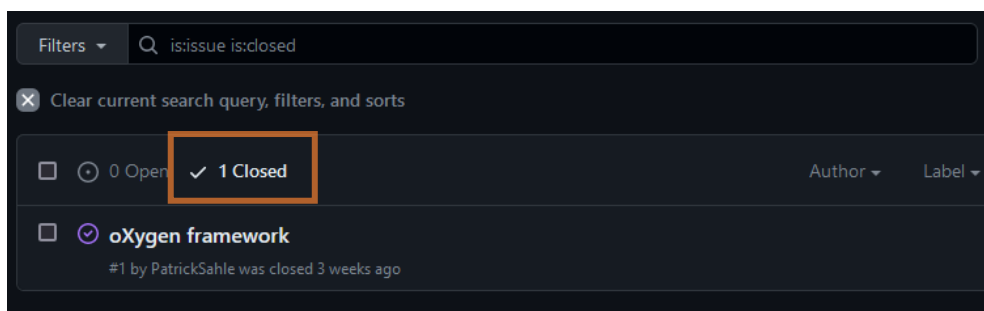
3) Click on  to participate in the discussion.

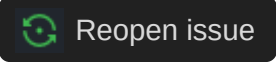


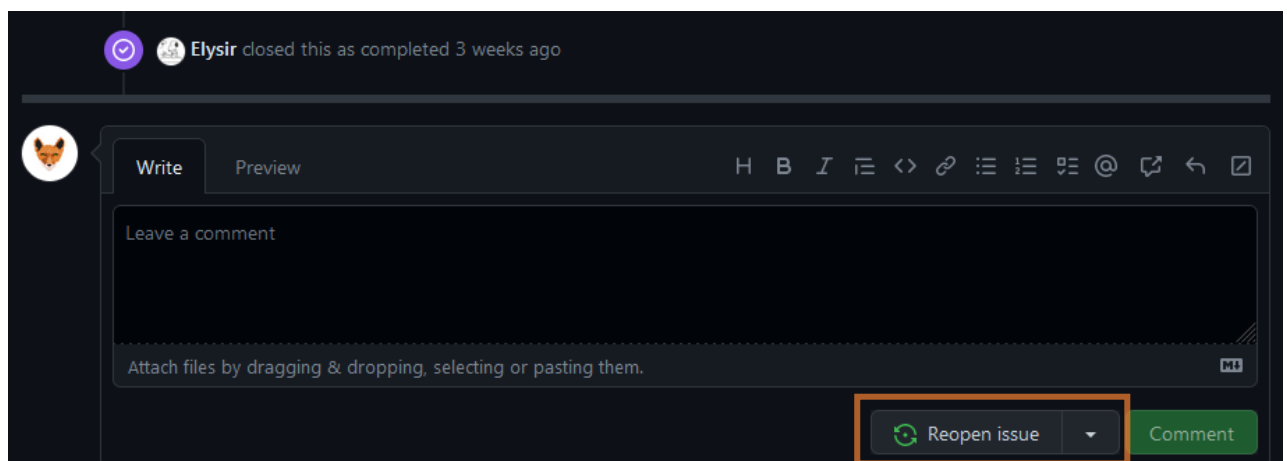
- 4) Close the issue if the discussion came to a conclusion or the problem is fixed (either with a last comment, or without one) and **don't forget to document the results in the wiki where they fit.**



- Closed issues can be found in the issue list under ✓ ***Closed***



- You can reopen a closed issue, by selecting  **Reopen issue** **inside** the closed issue.



GitHub (Desktop) Basic Workflow



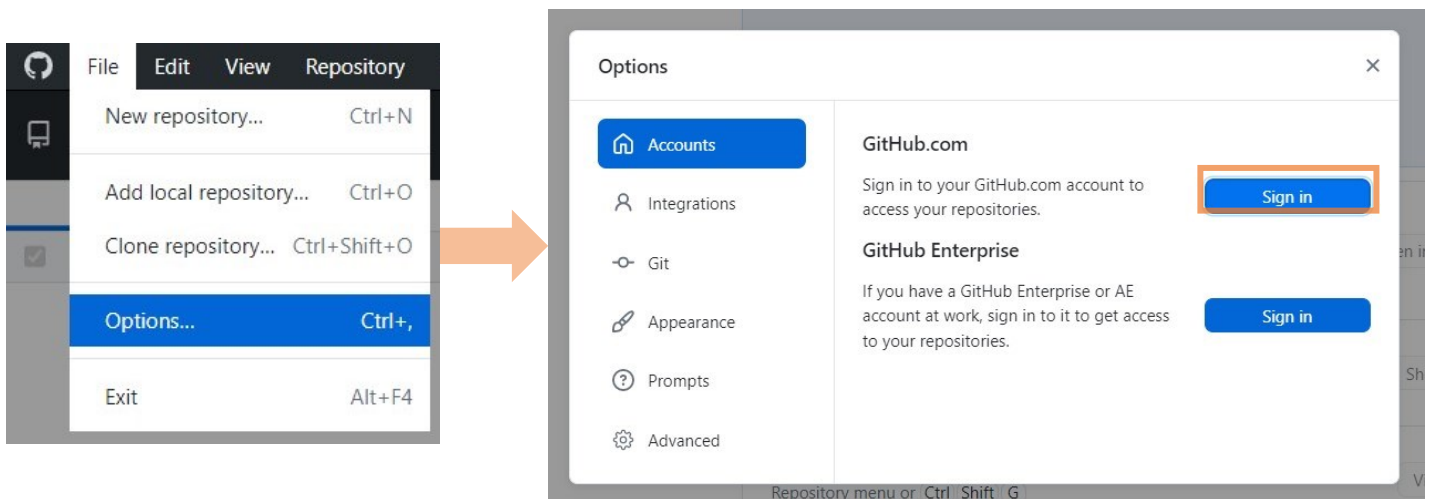
Set up GitHub Desktop

- 1) Download and install [GitHub Desktop](#)



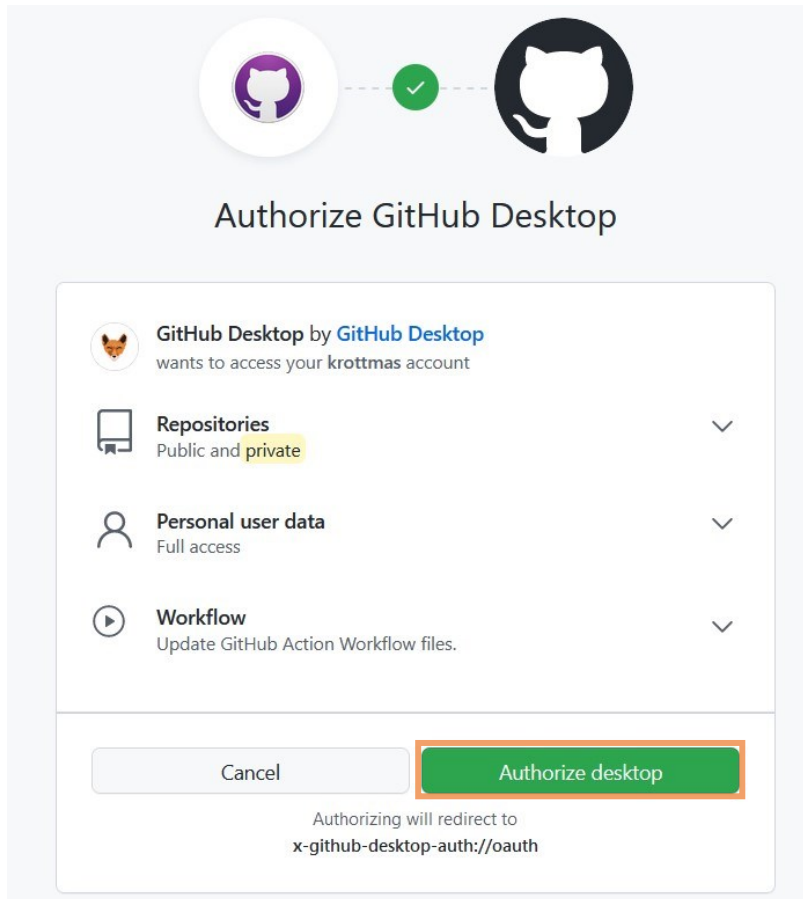
- 2) Start the program and connect your GitHub Account

Navigate to *File > Options > Accounts* and Sign in with your GitHub credentials



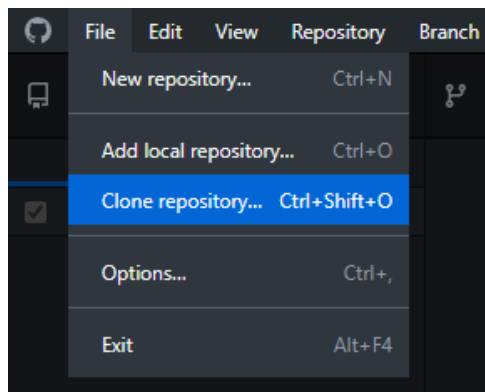
3) Give GitHub Desktop access to your account >>

Authorize desktop



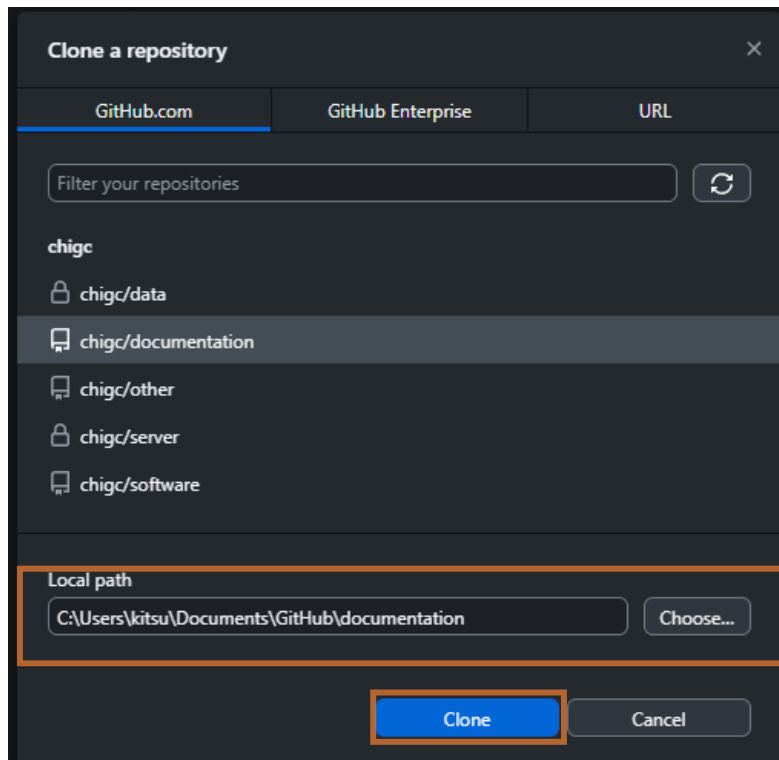
Clone (Copy) repositories

1) Navigate to *File > Clone Repository...*



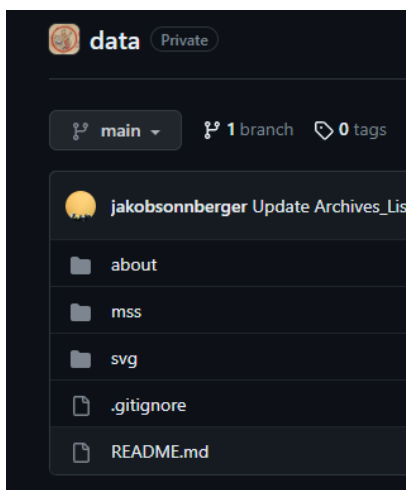
- 2) Select the repository you want to clone/copy from the list. You can only clone one repository at a time. Also, per default GitHub Desktop will clone/copy the selected repository into a new folder called *GitHub* in your *Documents* folder. You can change that if you like.

Click on 

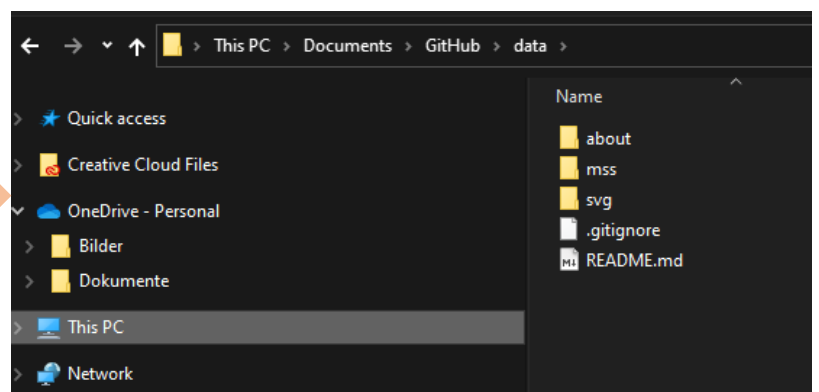


Now all the data from the GitHub repository is also available on your PC in your *Documents > GitHub* folder. Here you can access the files and work on them, e.g., open a manuscript-file and add new information (via the framework).

Data repository on GitHub



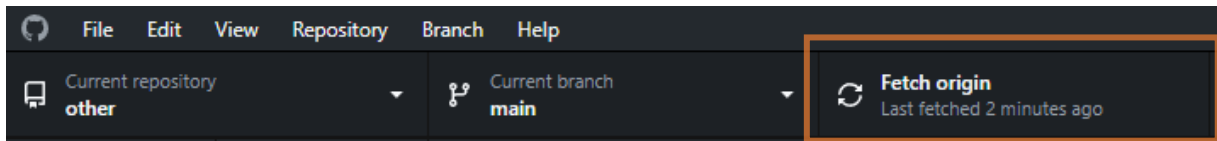
Local Clone/Copy



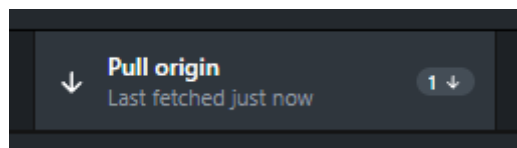


Get (pull) changes from the GitHub repository

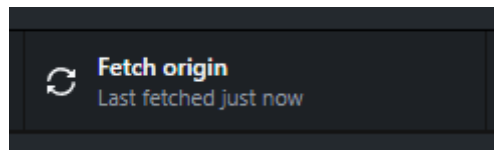
- 1) Open GitHub Desktop and click on **Fetch origin**



- 2) If there are changes made to the repository on GitHub, the button will change to **Pull origin** with the number of changes (pushes) made by other users. Click on **Pull origin** to update your files.

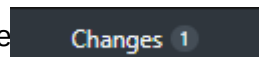


If there are no changes, the button will stay called **Fetch origin**.



Add (Push) changes to the GitHub Repository

- 1) Open GitHub Desktop. You can see a list of files with changes under the **Changes** Tab. If you select a File, you can also see, which changes were made:

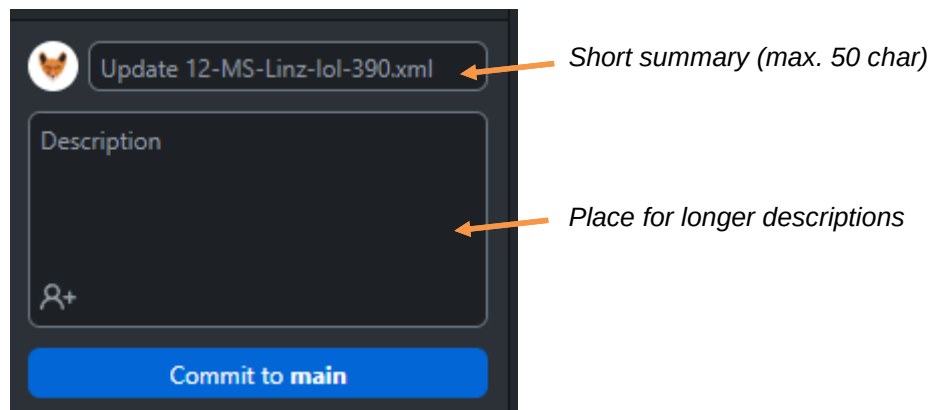


Changes 1	History	mss\tei\msDesc-data\12-MS-Linz-lol-390.xml
<input checked="" type="checkbox"/> 1 changed file		@@ -7,7 +7,8 @@
<input checked="" type="checkbox"/> mss\tei\...12-MS-Linz-lol-390.xml		7 7 <teiHeader>
		8 8 <fileDesc>
		9 9 <titleStmt>
		10 - <title type="main" ana="rdfs:label">Compendium Historiae in Oberösterreichische Lande
		10 + <title type="main" ana="rdfs:label">Compendium Historiae in Oberösterreichische
		11 + Landesbibliothek, Hs. 390 (Schiffmann 272)</title>
		11 12 <author ref="https://viaf.org/viaf/17579848/">Peter of Poitiers</author>
		12 13 </titleStmt>
		13 14 <publicationStmt>
		@@ -55,8 +56,9 @@
		55 56 <placeName>Linz</placeName>
		56 57 <country>AUT</country>
		57 58 </settlement>
		58 - <repository ana="chigc:hostedBy" corresp="#l01">Oberösterreichische Landesbibli
		59 - <idno type="signature" ana="chigc:signature">Hs.-390 (Schiffmann 272)</idno>

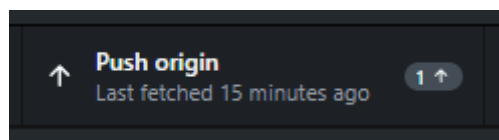
- 2) **Commit your changes.** Since those changes were made to your local files, you have to commit your changes to your local repository. Before you do so, write a short summary (max. 50 characters) of the changes – a commit message – in the text field next to your user-
icon on the left on the bottom of the page. If you need more space, you can use the Description field below.

Then click on

Commit to **main**



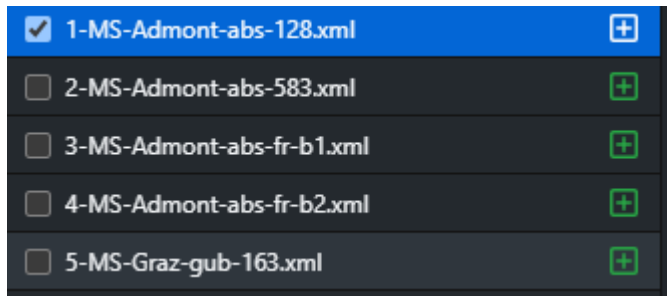
- 3) Click on **Push Origin** to add your changes to the GitHub Repository. The GitHub repository has now been updated and everyone can **fetch** and **pull** the most recent version.



Tips

- Check regularly if there have been updates
 - Before start working on data, **!!fetch and pull!!**
- Upload (commit and push) your updates regularly (at least at the end of your workday), so that everyone has the current version of the files
- Write commit messages
 - Commit messages do not need to be very long, but should made clear **what was done**
 - There are a few guidelines on how to write a good commit message (see [here](#)) but the most important thing is to be clear in what you have done:
 - e.g.
 - “Update XY” → rather meaningless message ❌
 - “Added Co-Transmissions and infos on Graph” or “Reviewed manuscript, flagged it ready to be published” → informative ✅

- You can commit one file at a time by unchecking the other files in GitHub Desktop



This way you can still write specific commit messages per file even if you have multiple changes in multiple files to upload to the repository on GitHub.

- Make sure that there are **NOT two or more people working on the same file at the same time** → divide the work on certain manuscripts beforehand.