

Introduction

This is written to guide you through publishing your book online with jupyter-book and hosting it on github. I wrote this guide as an exercise for me to get familiar with the jupyter-book library. At the same time it serves a note for myself. This guide is based on information from [jupyterbook](#) for more information please refer to the link.

Steps to Write a Jupyter Book

This section provides step by step guide to publish your book. I will start with the installation requirement. I am using a Windows machine for executing the steps, if you are using either a Linux or MacOS machine, there might be slight differences to the result. Please refer to the [original guide](#) if you run into any issue with your machine.

Installation

1. Install the Anaconda Individual Edition

- a. [Download the Anaconda Individual.](#)
- b. [Install the Anaconda Individual.](#)

2. Once Anaconda is installed. Create an environment called "jupyterbook" for installing the required libraries. This isolate all changes to within a Python environment.

- a. Create an environment with the following command:

```
conda create --name jupyterbook
```

3. Install Jupyter Book in the environment created.

- a. Activate the environment with the following command:

```
conda activate jupyterbook
```

- b. Install Jupyter-Book with the following command:

```
pip install -U jupyter-book
```

Create a Book Template with Jupyter-book

1. Change your directory to where you want to store your book content.

```
cd specify_your_dir_here/
```

2. Create a template book with this command.

```
jupyter-book create mynewbook/
```

3. Go into the "mynewbook" folder. You will see the files shown below. [Figure 1](#)










Name	Date modified	Type	Size
 _config.yml	11/19/2020 2:28 PM	YML File	1 KB
 _toc.yml	11/19/2020 2:28 PM	YML File	1 KB
 content.md	11/19/2020 2:28 PM	MD File	1 KB
 intro.md	11/19/2020 2:28 PM	MD File	1 KB
 logo.png	11/19/2020 2:28 PM	PNG File	10 KB
 markdown.md	11/19/2020 2:28 PM	MD File	4 KB
 notebooks.ipynb	11/19/2020 2:28 PM	IPYNB File	4 KB
 references.bib	11/19/2020 2:28 PM	BibTeX Document	6 KB
 requirements.txt	11/19/2020 2:28 PM	Text Document	1 KB

Fig. 1 Structure of a template book

Contents

[Installation](#)

[Create a Book Template with Jupyter-book](#)

[Edit the Book](#)

[Add Figure in your book](#)

[Reference to Other Documents,](#)

[Hyperlink and Folder Structure](#)

[Adding a Citation](#)

[Publish your Book on Github](#)

[Generate PDF](#)

4. Build the book with the following command. If the build is successful, you will see the message shown in [Figure 2](#).

```
jupyter-book build mynewbook
```

```
The HTML pages are in mynewbook\_build\html.  
[92m===== [0m  
Finished generating HTML for book.  
Your book's HTML pages are here:  
  mynewbook\_build\html\  
You can look at your book by opening this file in a browser:  
  mynewbook\_build\html\index.html  
Or paste this line directly into your browser bar:  
  file:///F:/mynewbook\_build\html\index.html  
[92m===== [0m
```

Fig. 2 Success message of a build

5. Go to mynewbook -> _build -> html -> index.html. Double click on the index.html. You will be able to view the book in your default browser as shown in [Figure 3](#).

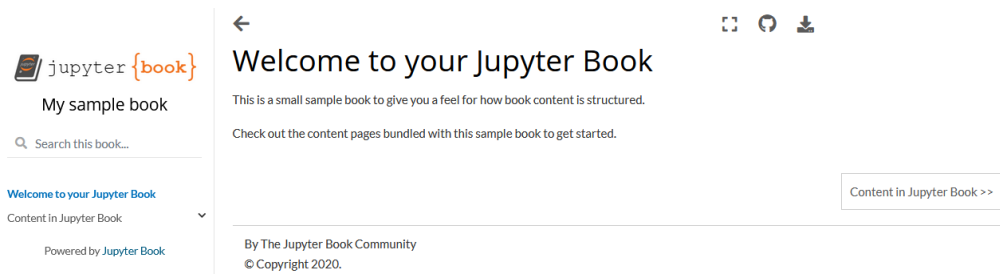


Fig. 3 Successful build of a book viewed in the browser

Edit the Book

1. Use a text editor to edit the files. I am using atom. [Download Atom here](#)
2. Open Atom and add the mynewbook folder. [Figure 4](#)

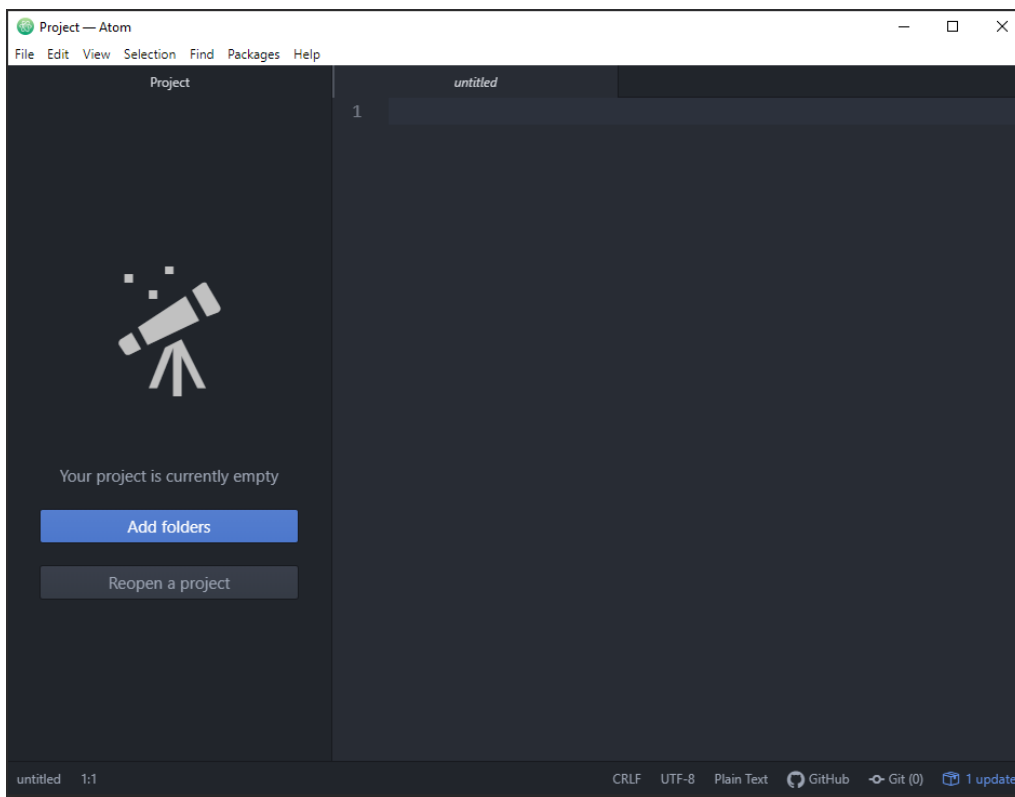


Fig. 4 The open screen of Atom

3. Once added you will be able to see the structure of the book in [Figure 5](#).

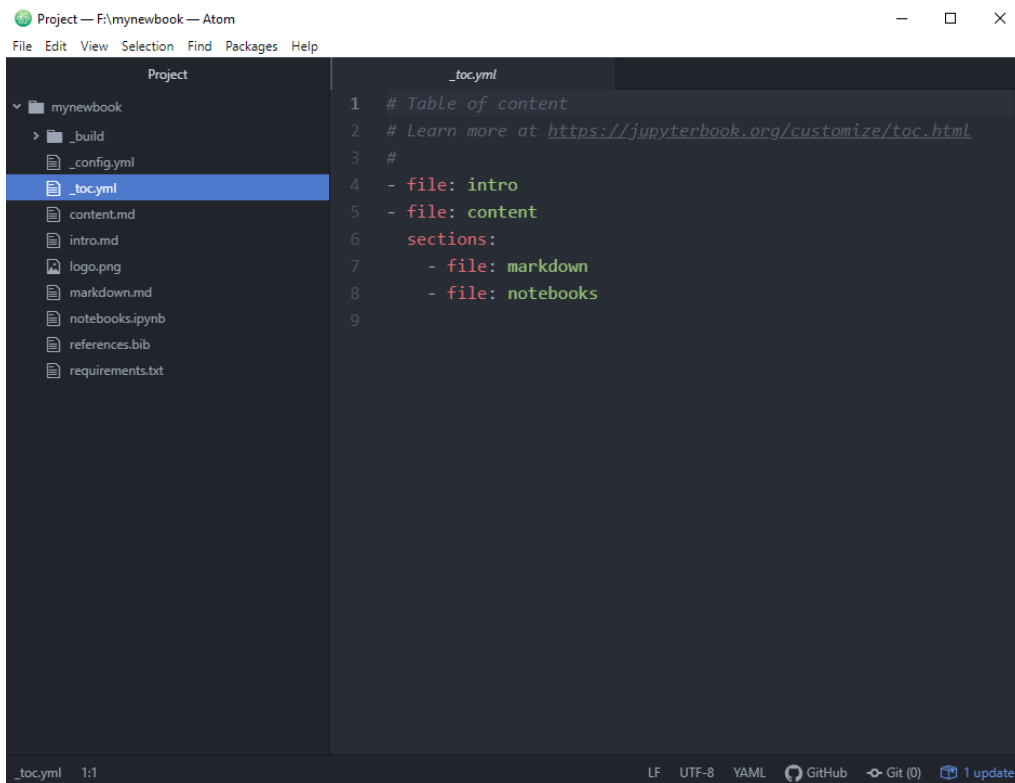


Fig. 5 The folder structure of the book in Atom IDE

4. To change the title of the book, go to the “_config.yml” file. Change the title of the book from “My sample book” -> “My notebook”. Change the author to your name ([Figure 6](#)). Save the change. Build the book with the command from the 4th Step of [Create a Book Template with Jupyter-book](#).

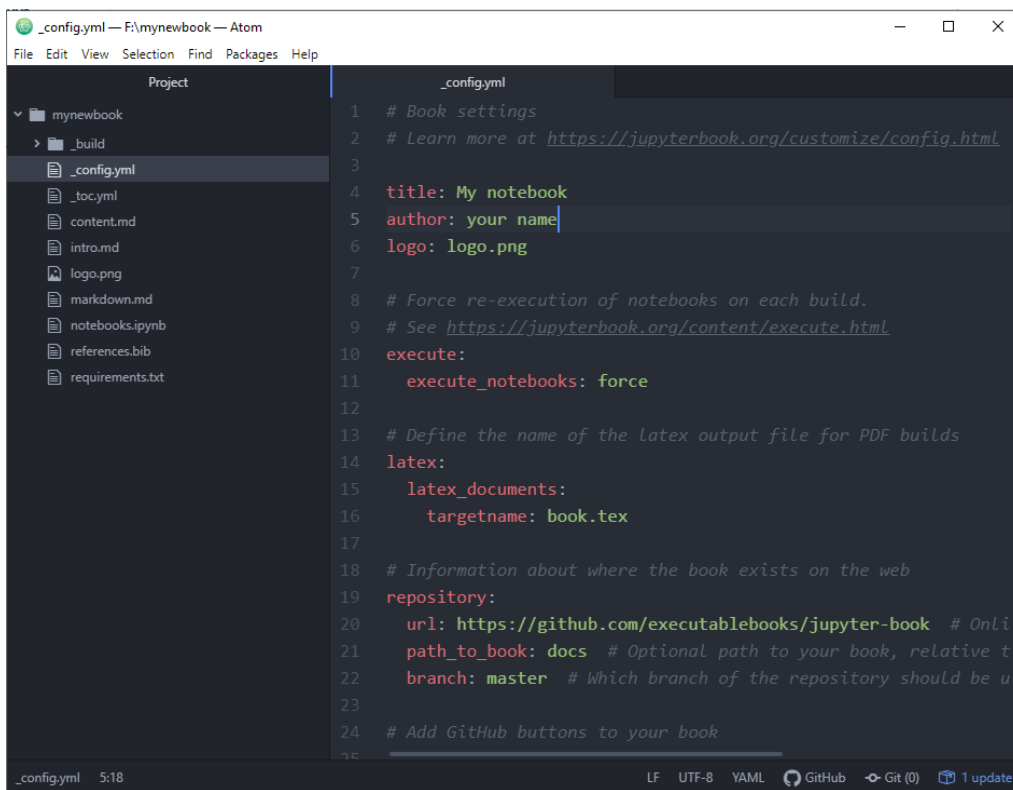


Fig. 6 Editing the _config.yml file

a. The result of the change (Figure 7).

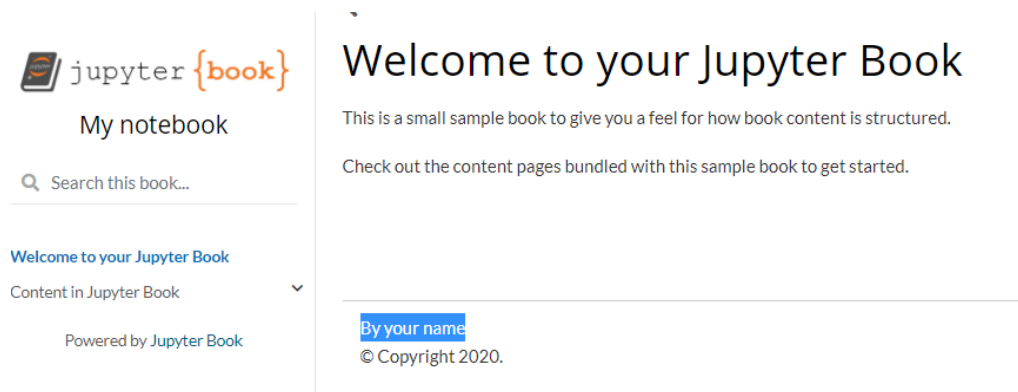


Fig. 7 Result from editing the _config.yml

5. Next I will make changes to the _toc.yml. This file controls the table of content on the left side of the screen. I will add a new file and some content to the file.

a. I add a new file called newcontent as shown in Figure 8. Remember to save the change and then build the book again.

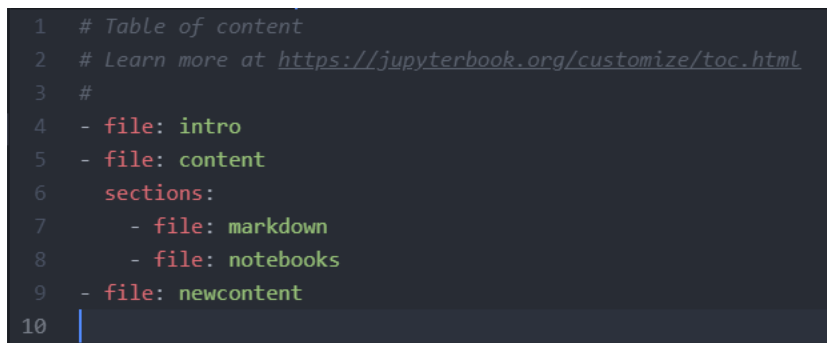


Fig. 8 Editing the _toc.yml file

b. With Atom add a file called “newcontent.md” as shown in [Figure 9](#). Add in the title “# My new content”.

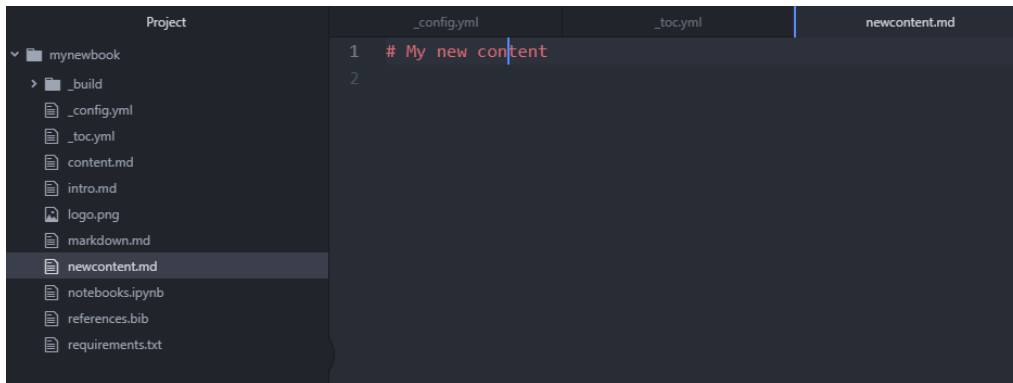


Fig. 9 Create a newcontent.md file

c. Build the book. You will be able to see the change as shown in [Figure 10](#)

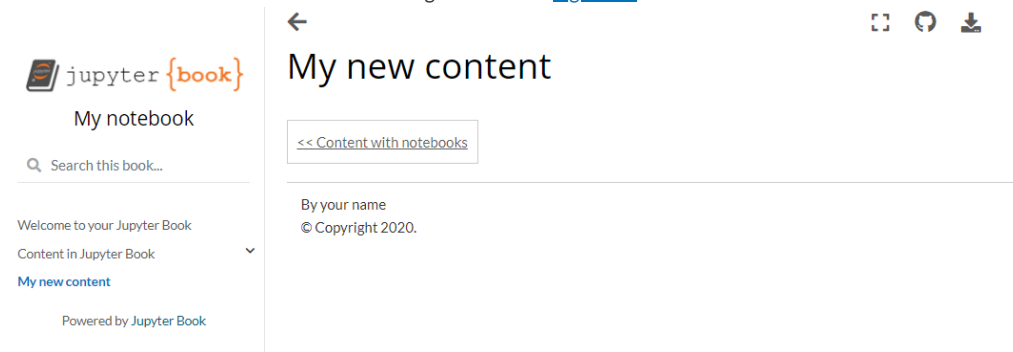


Fig. 10 Result of adding a newcontent.md file

Add Figure in your book

1. Add a new folder “_static” in the book folder as shown in [Figure 11](#).

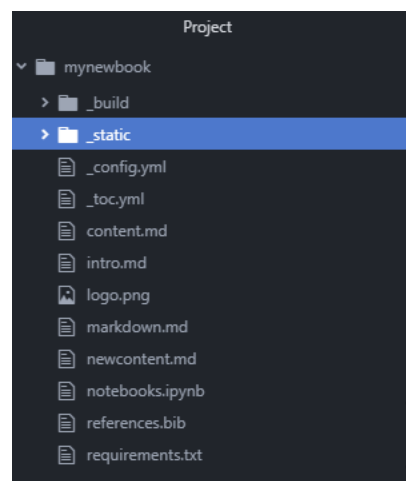


Fig. 11 Add a _static folder

a. Put a figurefile.png file into the folder as shown in [Figure 12](#).

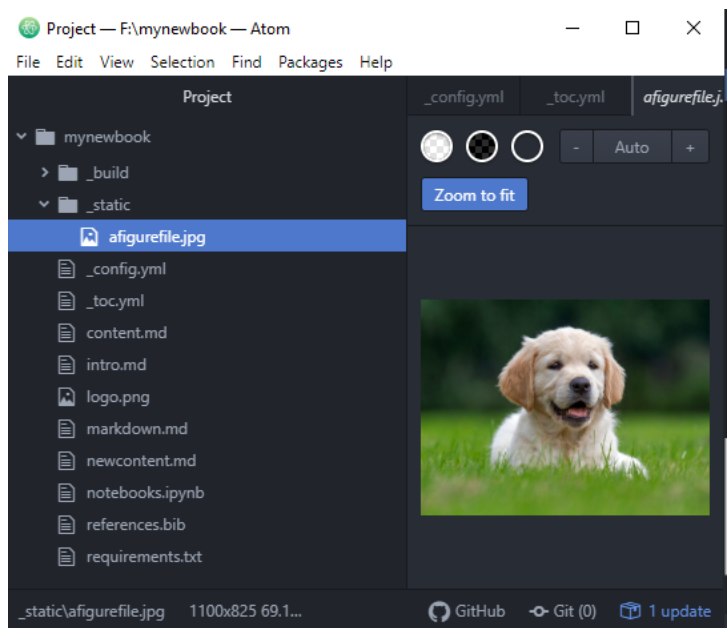


Fig. 12 Add a figure in the _static folder

2. Go to the “newcontent.md” file. In the file add the following line to add the figure into the page as shown in [Figure 13](#).

```
```{figure} /_static/afigurefile.jpg
:name: add_figure

Add a figure in the _static folder
```
```

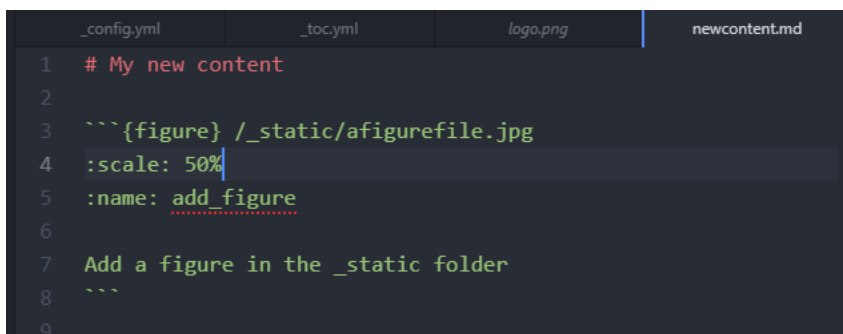


Fig. 13 Add a the command in the “newcontent.md” file

a. Build the book. You can see the result on your browser.

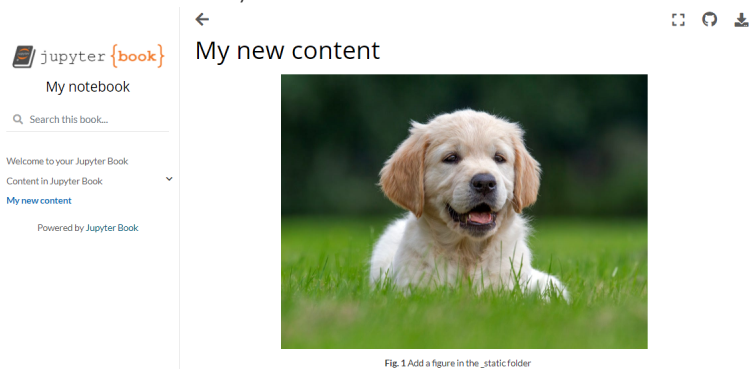


Fig. 14 Result of adding the figure onto the page

3. You can reference the figure in your text with the following command in the “new_content.md” file as shown in [Figure 15](#).

```
{numref}`Figure %s <add_figure>`
```

```

1 # My new content
2
3 ```{figure} /_static/afigurefile.jpg
4 :scale: 50%
5 :name: add_figure
6
7 Add a figure in the _static folder
8 ```
9 {numref}`Figure %s <add_figure>` shows a cute dog
10

```

Fig. 15 Reference the figure in your text

- a. Build the book. You can see the result in your browser.



Fig. 1 Add a figure in the _static folder

Figure 1 shows a cute dog

Fig. 16 Referencing your figure in your text.

Reference to Other Documents, Hyperlink and Folder Structure

1. You can reference to other documents in the book by using this command.

```
{doc}`020steps`
```

2. You can create a hyperlink with this command.

```
[text that will appear in content](https://the-hyperlink.com)
```

3. You can move the documents to a “docs” folder for a cleaner folder structure. You just have to specify the files in the “_toc.yml” file as “docs/file”. This is shown in [Figure 17](#)

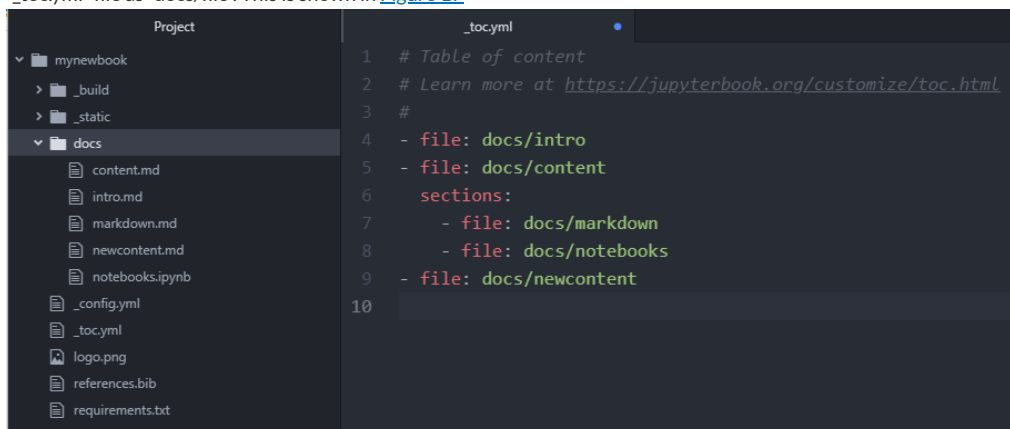


Fig. 17 You can restructure your book folder as shown

Adding a Citation

1. You will need to have a “reference.bib” file in your book folder.
2. You can add a citation with this command

```
{cite}`chen_exploring_2020`
```

3. You can generate the bibliography with this command. Remember to specify the right path to the .bib file. If the .bib file is in the previous folder specify it as “./references.bib”

```
...  
{bibliography} references.bib  
...
```

Example

I want to cite this article [\[CTM+20\]](#).

Bibliography

[\[CTM+20\]](#)

Kian Wee Chen, Eric Teitelbaum, Forrest Meggers, Jovan Pantelic, and Adam Rysanek. Exploring membrane-assisted radiant cooling for designing comfortable naturally ventilated spaces in the tropics. *Building Research & Information*, pages 1–13, November 2020. Publisher: Routledge. URL: <https://doi.org/10.1080/09613218.2020.1847025>, [doi:10.1080/09613218.2020.1847025](https://doi.org/10.1080/09613218.2020.1847025).

Publish your Book on Github

1. Register a Github account
2. Once registered. Create a new repository as shown ([Figure 18](#)).

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 *

chenkianwee ▾

Repository name *

pubbkonline ✓

Great repository names are short and memorable. Need inspiration? How about [refactored-guide?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)


☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☒ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

License: Creative Commons Z... ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository

Fig. 18 Create a repository on Github

3. For people who are not comfortable with CMD terminal. [Download Github Desktop](#). The rest of the steps assumed you are using github desktop.
4. Clone the repository you just created onto your local computer as shown in [Figure 19](#).

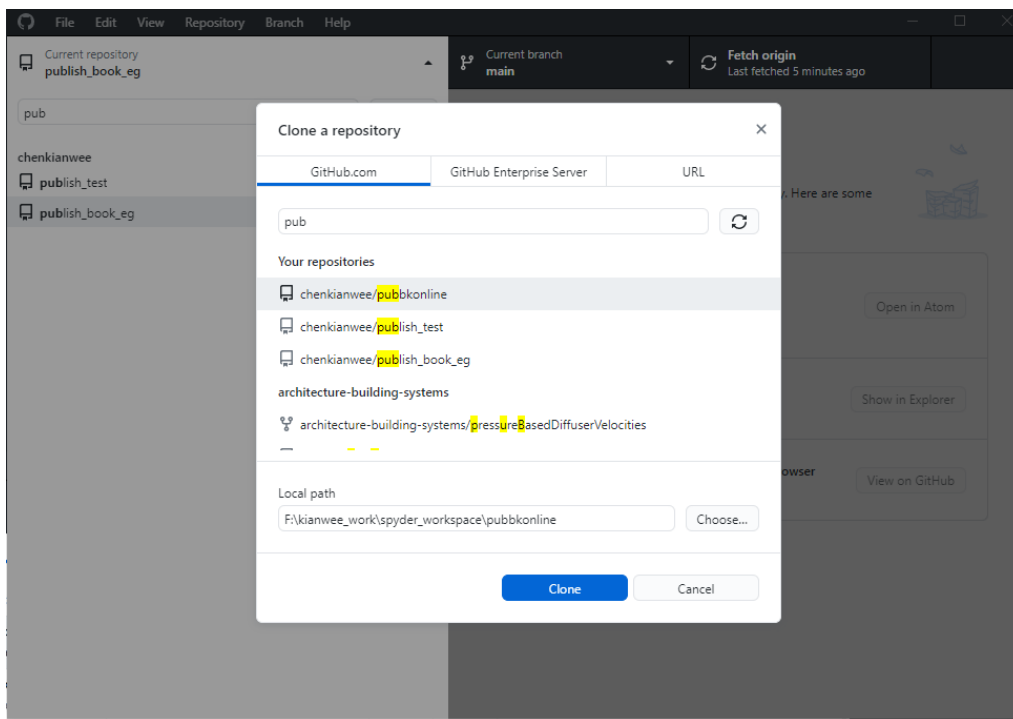


Fig. 19 Clone the repository from github using github desktop

5. Copy and paste your book folder into the local repository. Now on github desktop you will see all the changes.

“Commit to main” and then “Push Origin”. You will see your repository populated with all your files.

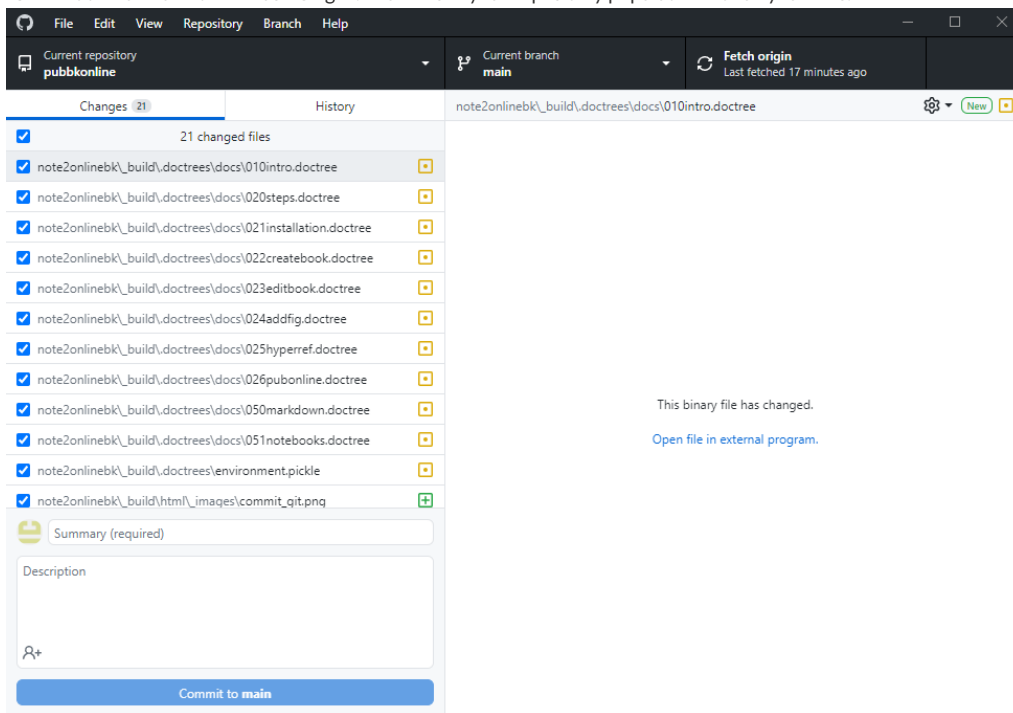


Fig. 20 Commit and sync the files onto the github repository


6. Install ghp-import to compile and host your book on Github-pages. Go to your anaconda environment and activate your jupyterbook environment as shown here [Installation](#). Once you activated the environment, execute this command.

```
pip install ghp-import
```

7. Go to the settings of your github repository. Turn on gh-pages, set your branch to gh-pages and folder to /root as shown in [Figure 21](#).

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

 Your site is published at <https://chenkianwee.github.io/pubbkonline/>

Source
Your GitHub Pages site is currently being built from the `gh-pages` branch. [Learn more.](#)

🔗 Branch: `gh-pages` 📁 `/ (root)` Save

Custom domain
Custom domains allow you to serve your site from a domain other than `chenkianwee.github.io`. [Learn more.](#)

Save

☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (`chenkianwee.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

Fig. 21 Turn on the gh-pages settings on your repository.

- Once that is done. On your local computer, cd to your book folder (the folder should contain the `_build` directory) and execute the following command.

```
ghp-import -n -p -f _build/html
```

```
(jupyterbook) F:\kianwee_work\spyder_workspace\pubbkonline\note2onlinebk>ghp-import -n -p -f _build/html
Enumerating objects: 154, done.
Counting objects: 100% (154/154), done.
Delta compression using up to 4 threads
Compressing objects: 100% (123/123), done.
Writing objects: 100% (154/154), 3.62 MiB | 5.30 MiB/s, done.
Total 154 (delta 24), reused 154 (delta 24), pack-reused 0
remote: Resolving deltas: 100% (24/24), done.
remote:
remote: Create a pull request for 'gh-pages' on GitHub by visiting:
remote:   https://github.com/chenkianwee/pubbkonline/pull/new/gh-pages
remote:
remote: To https://github.com/chenkianwee/pubbkonline.git
* [new branch]      gh-pages -> gh-pages
```

Fig. 22 Successfully executed ghp-import.

Generate PDF

- Install pypeteer with the following command.

```
pip install pypeteer
```

- Then build your book with the following command.

```
jupyter-book build mybookname/ --builder pdfhtml
```