Jupyter Book 2 Headstart Template

Freek Pols, Luuk Fröling, Robert Lanzafame, Kirstie Whitaker

Preface

Contents

1. Introduction **1**

2. Markdown & Jupyter Notebook **3**

2.1 This is a heading 3

3. Markdown (Cheat sheet) **6**

3.1 Book structure 6

3.2 Basic Formatting 6

3.3 Mathematics and equations 7

3.4 Lists 8

3.5 Tables 8

3.6 Blocks 8

3.7 Admonitions 9

3.8 Figures 9

3.9 YouTube 10

3.10 References & Links 10

4. Get going yourself **12**

4.1 Lesson 0: Head start with GitHub template 13

4.2 Lesson 1: Make your own content 18

4.3 Lesson 2: Include awesomeness 25

4.4 Lesson 3: Interactive python plot 26

4.5 Creating your landing page using CSS 27

5. Advanced start **29**

5.1 Install Jupyter Book 29

5.2 Initialise a project 29

5.3 Deploy to GitHub pages 29

5.4 Next steps 29

6. Software **31**

References **32**

1. Introduction

This is the *Jupyter Book 2 Headstart Template,* designed to quickly and easily produce your own online interactive textbook as well as a high quality pdf enabled with Typst using the Jupyter Book 2 technology.

This template:

- provides a ready-to-use Jupyter Book 2 structure for creating an online book (i.e., website)
- is pre-configured to produce a high quality pdf using Typst
- includes a number of lessons to get you started understanding key components of a book and how to edit it
- includes a GitHub Action to automatically build and deploy your book online and as a pdf

Hence, the template allows you to engage with JB2 without installing any software on your own computer. You only need a web browser and a GitHub account (we provide details on how to work locally on your own computer).

Work in Progress

Materials in this document are under construction; the contents and order of the information and tasks may change, depending on workshop needs and feedback.

Note

The template is and its content are not meant as a replacement of the documentation already available on the Jupyter Book 2 website and the MyST website. It is designed to support new users of Jupyter Book 2 and MystMD, in particular for use in workshop settings where participants may not have time or ability to install the required software on a personal computer.

Enjoyment Warning!

Once you start building JB2 books, it is likely you will get hooked!

2. Markdown & Jupyter Notebook

Jupyter Book 2 (JB2) supports both Markdown files and Jupyter Notebooks as content sources.

2.0.1 Markdown

Markdown is a simple markup language: plain text that is *formatted* with small pieces of 'code'. This allows you to create rich, interactive books that combine text, code, and visualizations. This text can then be quickly exported to various other formats such as PDF, Word, HTML, etc.

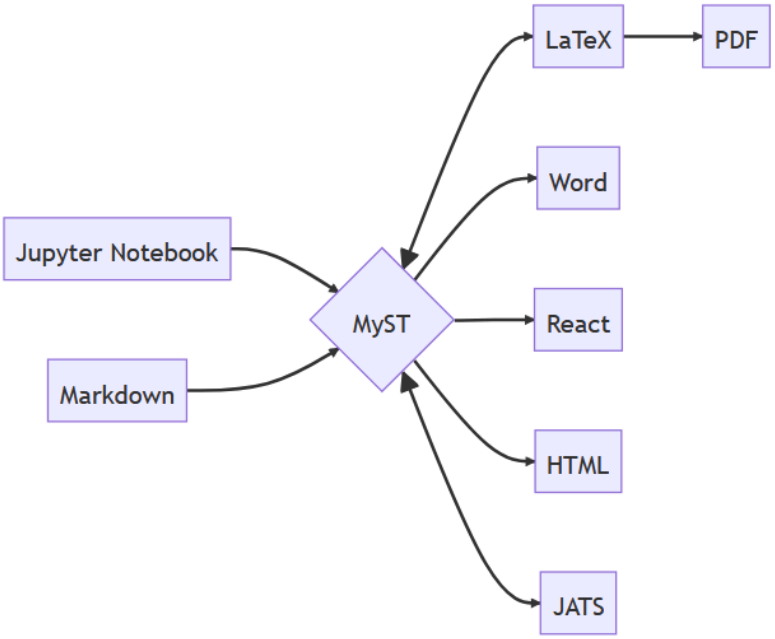

Figure 2.1: Documents made in MyST Markdown can be converted to many different formats. These can be saved as JSON, or rendered to a website (like this one!) or any number of formats including PDF & LaTeX, Word, React, or JATS. Picture taken from the MYST documentation[1].

See below for an example of the Markdown language and how it is rendered in JB2.

rendered

2.1 This is a heading

This is a paragraph with **bold** and *italic* text.

- This is a list item
- Another list item

raw HTML

You can also include raw HTML in your Markdown files, but this will not be rendered in the PDF output.

[1] https://mystmd.org/guide/

markdown

```
# This is a heading
This is a paragraph with **bold** and *italic* text.
- This is a list item
- Another list item
```{warning} raw HTML
You can also include raw HTML in your Markdown files, but this will not
be rendered in the PDF output.
```
```

In the next chapter we cover most of the basic Markdown syntax you will need to create your own content.

2.1.1 Jupyter Notebooks

Jupyter Notebooks are interactive documents that combine live code, equations, visualizations, and narrative text made with Markdown. They are widely used for data analysis, scientific research, and teaching, allowing users to run code in a step-by-step manner and see immediate results.

JB2 allows to include Jupyter Notebooks directly in your book, making it easy to share interactive content with your readers. It also allows to run code cells and display the output directly in the book. Check the output below by first clicking the ON-button at the top right of this page. This loads the server. Once it is ready, you can run the code cell below by clicking the play button.

```
print("Hello, World!")
```

2.1.2 Cell Tags and Hiding Input

Cell tags in Jupyter Notebooks are metadata labels that you can assign to individual cells. They are useful for customizing the behavior of cells, such as hiding code input.

To hide the input of a code cell (so only the output is visible), you can add a tag like `hide_input` to that cell. JB2 recognizes this tag and will hide the code input when rendering or exporting the notebook.

To add a tag:

1. Select the cell.
2. Open the “View” menu and enable the “Cell Toolbar” arrow.r “Tags”.
3. Add the tag `hide_input` (or another tag recognized by your toolchain).

This feature is especially useful for creating clean, reader-friendly notebooks where you want to focus on results rather than code details, see below.

The code above provides a semi-interactive simulation: you are able to change two parameters but cannot control/adapt the code itself. This will be available in the near feature.

3. Markdown (Cheat sheet)

Below is a set of frequently used markdown commands for Jupyter Book 2 made with MyST. A good practice is to download the source file by clicking the download icon at the top right of this page and inspect the code.

3.1 Book structure

We can distinguish between two structures: that of the book's content (a collection of different documents), and the (internal) structure of the chapters which consists of content structured in sections and subsections.

3.1.1 Table of Contents

In the `myst.yml` file, you can specify the structure of the book as shown in Program 1. Here you can indicate which files belong to the book and in what order. You can also create dropdown menus. When not specifying a ToC, all files are automatically included in alphabetical order.

```yaml
toc:
  - file: index.md                      # the landing page
  - file: content/1_intro.md
  - file: content/2_jup_nb.ipynb
  - file: content/3_cheat_sheet.md
  - file: content/lessons/your_turn.md          # dropdown menu
    children:
      - file: content/lessons/lesson0.md
      - file: content/lessons/lesson1.md
      - file: content/lessons/lesson2.md
      - file: content/lessons/lesson3.ipynb
      - file: content/lessons/lesson4.md
  - file: content/advanced_start.md
  - file: content/software.md
```

Program 3.1: The Table of Contents (ToC) for this book.

If you create a new file, you need to add it to the `myst.yml` file to include it in the book.

3.1.2 Headings

To make sections within a page, use a number of # symbols at the beginning of al line. The more #s increases the level of the heading.

```
# Heading level 1
## Heading level 2
### Heading level 3
```

Typically, a page only has a single level 1 heading, the page's title and higher level headings are used for sections and subsections.

> **Tip**
>
> Do not number your chapters and sections! This happens automatically.

3.2 Basic Formatting

Markdown is a markup language where text formatting is done with small pieces of code (just like HTML). This is a table with some frequently used formatting options.

Element	Syntax	Example

{MyST}

Element	Syntax	Example
Bold	`**bold text**`	**Bold**
Italic	`*italics*`	*Italics*
Emphasis	`***emphasis***`	***emphasis***
Inline Formula	`$F = m \cdot a$`	$F = m \cdot a$
Footnote	`- A footnote reference[^myref] [^myref]: This is an auto-numbered footnote definition.`	• A footnote reference[2]

You can create a new line by either a hard enter and a blank line, a \ at the end of the line and enter, or two spaces at the end of the line.

3.2.1 New Lines

Generally, in markdown, a single hard enter does not create a new line. You need to use one of the options below.

A new line with double space. A new line with a \.
A new line with a hard enter and blank line.

No new line with just a hard enter and text on the next line. Like this

3.3 Mathematics and equations

For STEM subjects, mathematical equations and symbol are essential. You can include equations in JupyterBooks using the LaTeX syntax for mathematics.

Labelled equations using double dollars, such as (1), can be referenced.

$$F_{\text{res}} = m \cdot a \tag{3.1}$$

But you can also write inline equations with single dollars.

$s = v_{\text{avg}} t$

Some examples of the notation

Name	Syntax	Output
Greek script	`\Delta \lambda`	$\Delta\lambda$
root	`\sqrt{4}`	$\sqrt{4}$
superscript/power	`x^{2a}`	x^{2a}
fraction	`\frac{2}{3}`	$\frac{2}{3}$
subscript	`x_{\mathrm{avg}}`	x_{avg}
multiply	`a \cdot b`	$a \cdot b$
derivative	`\frac{\Delta f}{\Delta t}`	$\frac{\Delta f}{\Delta t}$
integral	`\int_a^b f(x) \mathrm{d}x`	$\int_a^b f(x)\mathrm{d}x$
sine	`\sin(x)`	$\sin(x)$

You can read about more of the LaTeX mathematics syntax in the LaTeX wikibook

[2] This is an auto-numbered footnote definition.

3.4 Lists

3.4.1 Ordered lists

Ordered lists can be made with automatically numbered items.

1. item 1
2. item 2.
3. item 3.

Or you can manually number the items

1. item 1
2. item 2.
3. item 3.

3.4.2 Unordered lists

Unordered lists use - or * for each item.

- a
- b
- c

3.4.3 Checklists

You can also create checklists. The checks are interactive, you can tick or untick them.

- Create a markdown cheat sheet
- Publish online
- Let others test

3.5 Tables

Tables are created with the separator |

Header 1	Header 2	Header 3
text 1	text 2	text 3
text 4	text 5	text 6

Or via ...

Behavior
• Sanction for 1st time • Sanction for 2nd time
Not (timely or with a valid reason) deregistered
• A penalty • exclusion

Table 3.2: Overview of sanctions for certain behavior

Method 2 has the advantage of allowing references to Table 1

3.6 Blocks

3.6.1 Tabs

Tab 1

Text in tab 1

Tab 2

Text in tab 2

3.6.2 Cards

Title

With some text

3.7 Admonitions

You can add special blocks that are highlighted in the text. See, for example, the warning below.

Warning

Here is a warning

There are different variants such as:

- tip
- admonition
- warning
- note
- objective

3.7.1 Exercises

A special case is the exercises which can be labelled and can come with a solution that is linked to the exercise:

3.8 Figures

A site/book naturally needs figures. There are roughly two ways to add a figure:

Quick figure, without formatting options

```
![](path/to/figure)
```

Better way with more control:

Exercise 3.3: Exercise 1

Calculate $4 + 2$

Solution 3.4: Solution to Exercise 1

6

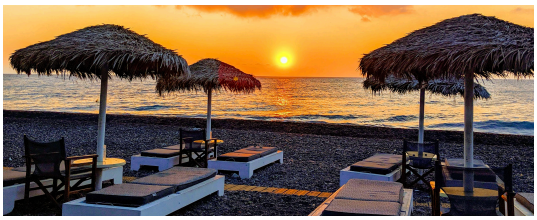

Figure 3.5: With a nice caption

Here we used figures hosted online, but you can also add figures to a folder (*e.g.*, called `figures`), and then use a relative path.

3.9 YouTube

To embed YouTube videos on the site, you need the embed YT link. The code is then:

Figure 3.6: A super fun video from the project Show the Physics

YT in pdf

Embedded YT videos are not included in the PDF. A solution could be to include a QR code, for example.

3.10 References & Links

You can included links like this. With the markdown syntax: `[text](link)`.

It also possible to include reference through the reference.bib file, or directly by including the DOI: Pols (2021) (syntax: `[](https://doi.org/10.1088/1361-6552/abf208)`) or (Pols, 2021) (syntax: `[@doi:10.1088/1361-6552/abf208]`).

Below a few examples of links and references to labeled items.

- This is a hyperlink
- This is a reference to equation (1)
- This is a reference to a table like Table 1
- This is a reference to a figure like Figure 1

4. Get going yourself

Now it's your turn to get going with creating content in a Jupyter Book.

There are some conditions to meet before you can get started:

- Programming experience is not necessary, but it does make things a bit easier.
- A free GitHub account.
- A little bit of patience, it's a steep learning curve.

Optional:

- Software to write your code in, such as Visual Studio Code (VSC).

> **Tip**
>
> If you are feeling confident and would rather start building a Jupyter Book from scratch, you can look at the advanced start.

We provide instructions for building using the GitHub dev IDE, as well as building locally. For the latter we assume knowledge about VSC, and the use of git.

Ready? Let's get started...

4.1 Lesson 0: Head start with GitHub template

This lesson gives you a head start in creating your own online book using GitHub and the template repository JB2_book_template.

4.1.1 Set up your own repository

Follow these instruction to use this template for your own book.

1. Go to this repository
2. Click the green button `use this template` and click `create a new repository`.
3. Choose a proper name of your repository (this will be also part of your URL!) and choose the option `public`.
4. In your repository, click on `settings` and in the left menu on Pages and choose `Github actions`

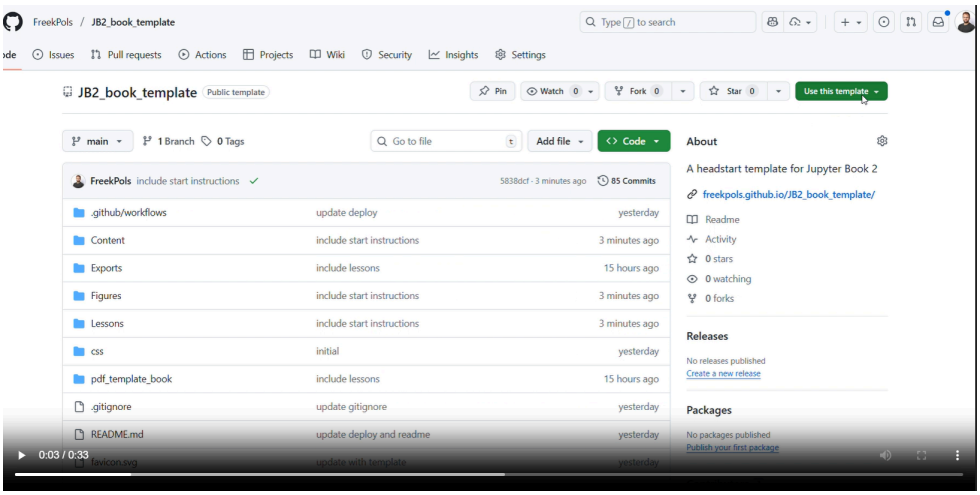

Figure 4.1: Follow these steps to create your own repository from the template.

5. Click on `code` and click on the `gear-icon` (near **About**) at the right site of the page.
6. Check the box **Use your GitHub Pages website**.
7. Go to `actions` in the topmenu, click on the (red) `initial commit` and click `re-run all jobs`

The book will now be deployed again - where now it can actually load GitHub pages.

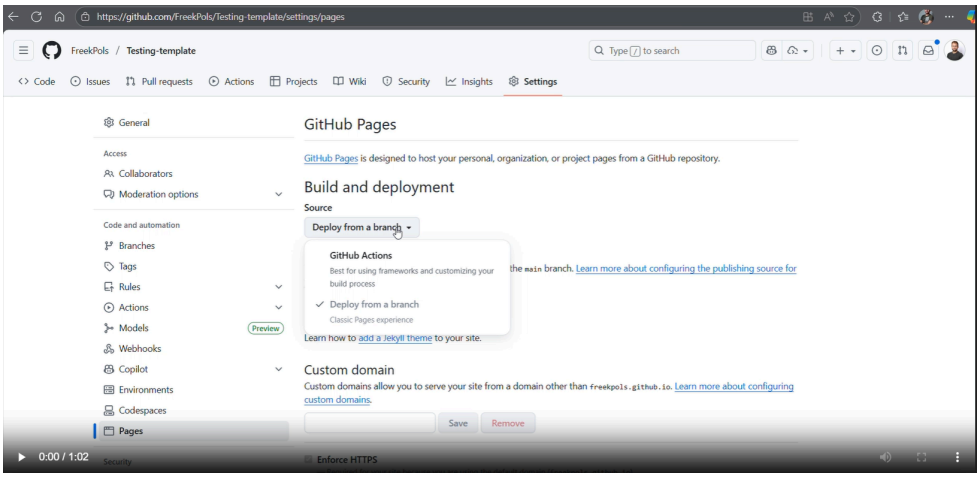

Figure 4.2: Follow these steps to create your own GHpages from the template.

8. Use the book link (`code` → below **About**) to your Github page where the book is hosted.

9. The output resembles Figuur 3.

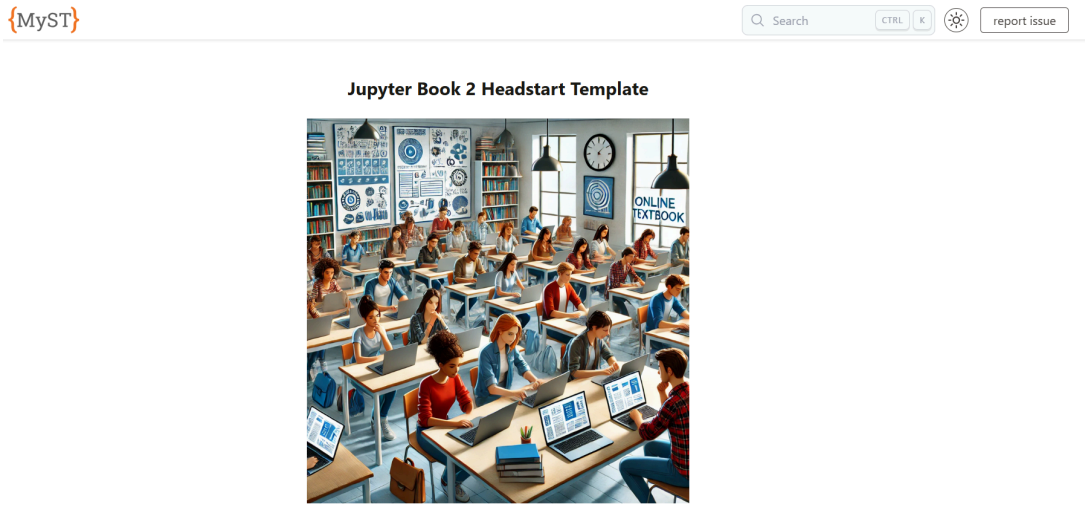

Figure 4.3: Once the book has been deploy, you can visit your site which looks like this.

4.1.2 Repo folder structure

Your GitHub repository looks like the one shown in Figure 4. We have the following subfolders:

- Content: the source files of your book (in markdown or jupyter notebook format)
- Exports: the folder which may include a pdf export of your book
- Figures: the folder which includes figures for your book (*could be in content folder*)
- Lessons: the folder which includes the lessons of this tutorial (*could be in Content folder*)
- .github/workflows: the folder which includes the GitHub actions (automated workflows) to build and deploy your book
- css: the folder which includes the custom css file to change the layout of your book
- pdf_template_book: the folder which includes the typst template to create a pdf export of your book

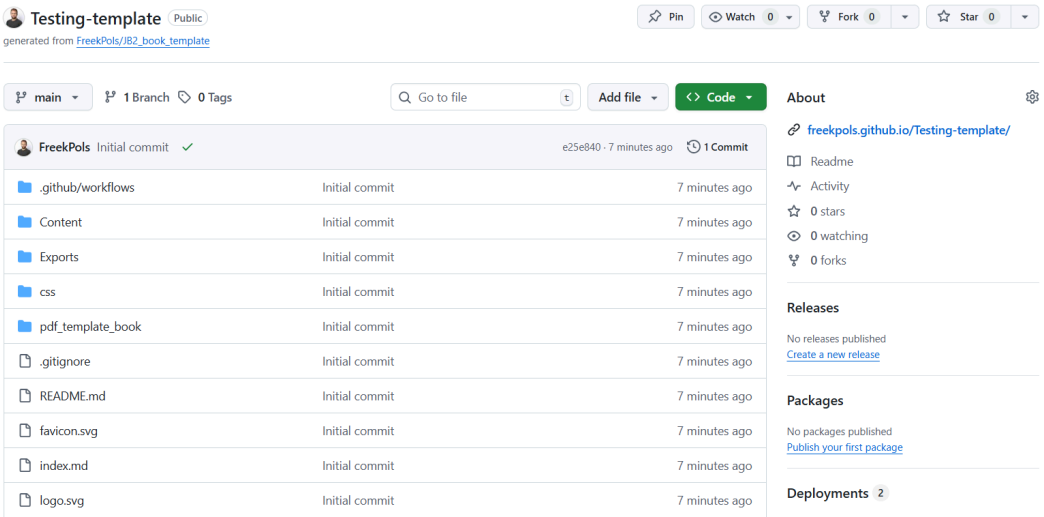

Figure 4.4: Once the book has been deploy, you can visit your site which looks like this.

4.1.3 Make and deploy changes

You have a number of options for making changes to the book's source and seeing how they affect the output.

Using the GitHub IDE

It is possible to work directly in the GitHub environment: no need to install anything as this is already covered with the GH actions that we created.

1. Click on the index.md file in the Content folder
2. Click on the drop down icon next to the pencil icon and choose `open in github.dev` This will start the GitHub development environment where you can edit the files directly in your browser.
3. Edit the file by replacing the names with your own and commit your changes, see Figure 5

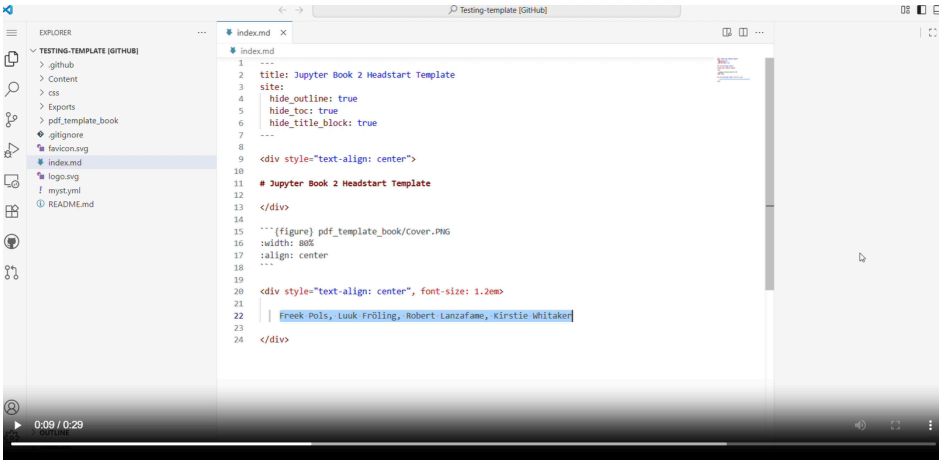

Figure 4.5: Working directly in the GitHub development environment.

Now, if you go back to your repository and click on `actions` you will see that the workflow is running to build and deploy your book. After a few minutes, you can refresh your book page and see your changes live!

Using your favourite editor

You can also make changes locally then push them back to your GitHub repository.

1. Clone the repository to your local machine using Git.

```
git clone git@github.com:<github_user_name>/JB2_book_template.git
```

2. Make changes to the content files in the `content` directory using your text editor.
3. Commit and push your changes. For example

```
git commit -a
git push -u origin main
```

Now, if you go back to your repository and click on `actions` you will see that the workflow is running to build and deploy your book. After a few minutes, you can refresh your book page and see your changes live!

Entirely locally

If you prefer, you can also work entirely locally using command-line tools and a text editor.

1. Clone the repository to your local machine using Git.

```
git clone git@github.com:<github_user_name>/JB2_book_template.git
```

2. Install MyST Markdown. Using `npm` or `pip`

```
npm install -g mystmd
```

```
pip install mystmd
```

> **Hint**
>
> To build a pdf using Typst you will need to install the Typst CLI. There are a number of options for installing Typst.

3. Make changes to the content files in the `content` directory using your text editor.
4. Serve the book locally.

```
myst start
```

5. Preview the book in your browser at http://localhost:3000

> **Tip**
>
> Once the MyST server is running, it will automatically update as you make changes to the source.

4.1.4 Export to pdf

You can export your book to a pdf in two ways, with LaTeX or Typst. Both methods have advantages and disadvantages.

4.1.4.1 Configure pdf output

We specified in the `myst.yml` file that we want to export to pdf using Typst. You can also choose LaTeX. See the `myst.yml` file, or Program 1 for the syntax.

```
  exports:
    - format: typst
      template: https://github.com/myst-templates/plain_typst_book.git
#plain_typst_book
      output: exports/book.pdf
      id: output-pdf
  downloads:
    - id: output-pdf
```

Program 4.6: Example of the export section in the `myst.yml` file.

You can specify the output template. We won't go into detail here, but you can find more information in the MyST Markdown documentations.

4.1.4.2 GitHub workflow

This template includes a GitHub workflow that automatically builds the pdf for you with Typst when you push changes. Using the export options in `myst.yml`, a pdf will be built and committed to the `exports` directory when you push to GitHub.

Note

The GitHub workflow this only works when there are no errors in the markdown files. For instance, it breaks when figures are missing.

4.1.4.3 Local pdf build

If you have the Typst CLI installed, you can also build a pdf locally.

```
myst build --typst
```

The book will be written to `exports/book.pdf`.

4.2 Lesson 1: Make your own content

4.2.0.1 Anatomy of a Jupyter Book

A Jupyter Book is a collection of files and folders that together make up the content and structure of your book. The structure of the book is specified in the `myst.yml` file, which is located in the root directory of your book. This file contains information about the title, author, and other metadata of the book, as well as documents and its structure to build the book itself.

```yaml
# See docs at: https://mystmd.org/guide/frontmatter
version: 1
project:
  id: 7204695b-7485-4989-863c-16cf6e4db155
  title: Jupyter Book 2 Headstart Template
  # description:
  # keywords: []
  authors:
    - name: Freek Pols
    - name: Luuk Fröling
    - name: Robert Lanzafame
    - name: Kirstie Whitaker
  license:
    content: CC-BY-4.0
```

Program 4.7: The head of this book's `myst.yml`

Official documentation

See here for a full explanation of the structure and TOC and here for more the website options.

4.2.0.2 Your first changes via GitHub 🌶️

As explained in the previous chapter, your files are on GitHub and the template ensures the book is built. You can make changes directly to the files online in GitHub, and create or upload new files.

Some files are already present in the template book. The folder structure is shown below

```yaml
toc:
  - file: index.md                      # the landing page
  - file: content/1_intro.md
  - file: content/2_jup_nb.ipynb
  - file: content/3_cheat_sheet.md
  - file: content/lessons/your_turn.md          # dropdown menu
    children:
      - file: content/lessons/lesson0.md
      - file: content/lessons/lesson1.md
      - file: content/lessons/lesson2.md
      - file: content/lessons/lesson3.ipynb
      - file: content/lessons/lesson4.md
  - file: content/advanced_start.md
  - file: content/software.md
```

Program 4.8: The Table of Contents (ToC) for this book.

We will now make a small change to one of the files and then look at the result of that change.

Exercise 4.9: Your first change

GH IDE

Navigate to the file `book/some_content/overview.md`. Then click on the pencil on the right (edit this file).

Change the text after the `#`. This is the title of the file.

Figure 4.10:

Optionally, make other changes in the text editor and when you're done, commit your changes to the "remote repository" by clicking the green `Commit changes` button.

The book will now be rebuilt. Once that's done, you can view the result on the GitHub page.

Local

Navigate to the file `book/some_content/overview.md` and open the file.

Change the text after the `#`. This is the title of the file.

Optionally, make other changes in the text editor. Check the results by running `myst start` in the CLI and opening the local server `http://localhost:3000/`. When you're done, commit your changes to the "remote repository".

The book will now be rebuilt. Once that's done, you can view the result on the GitHub page as well.

4.2.0.3 Create a chapter

4.2.0.3.1 Create a section

4.2.0.3.2 Include an equation

4.2.1 Your first changes via GitHub

As explained in the previous chapter, your files are on GitHub and the template ensures the book is built. You can make changes directly to the files online in GitHub, and create or upload new files.

Some files are already present in the template book. The folder structure is shown in … BROKEN REF

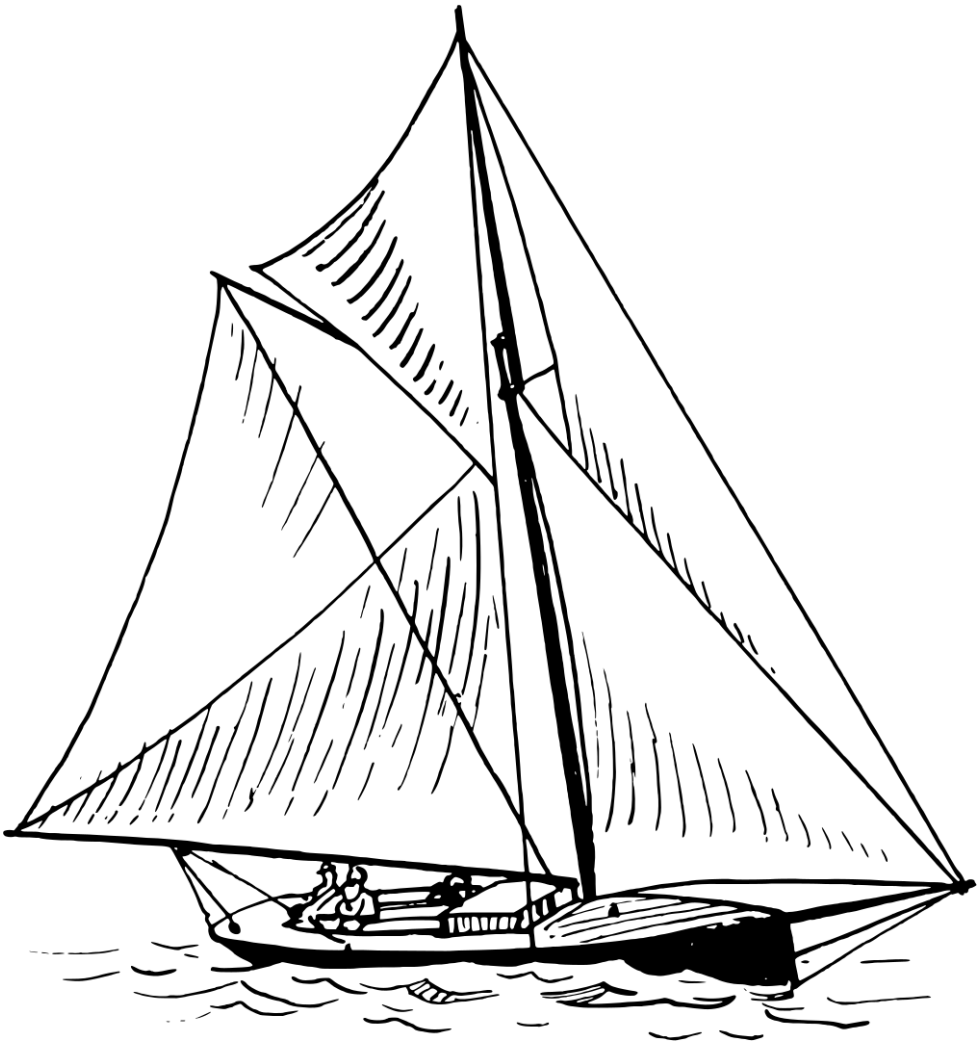

Figure 4.11:

4.2.1.1 width: 80% name: fig_templatecontent

The folder structure in the template book.

We will now make a small change to one of the files and then look at the result of that change.

4.2.1.2 Adding a figure

A figure sometimes says more than 1000 words…

Exercise 4.12: Your first change

Navigate to the file `book/some_content/overview.md`. Then click on the pencil on the right (edit this file).

Change the text after the `#`. This is the title of the file.

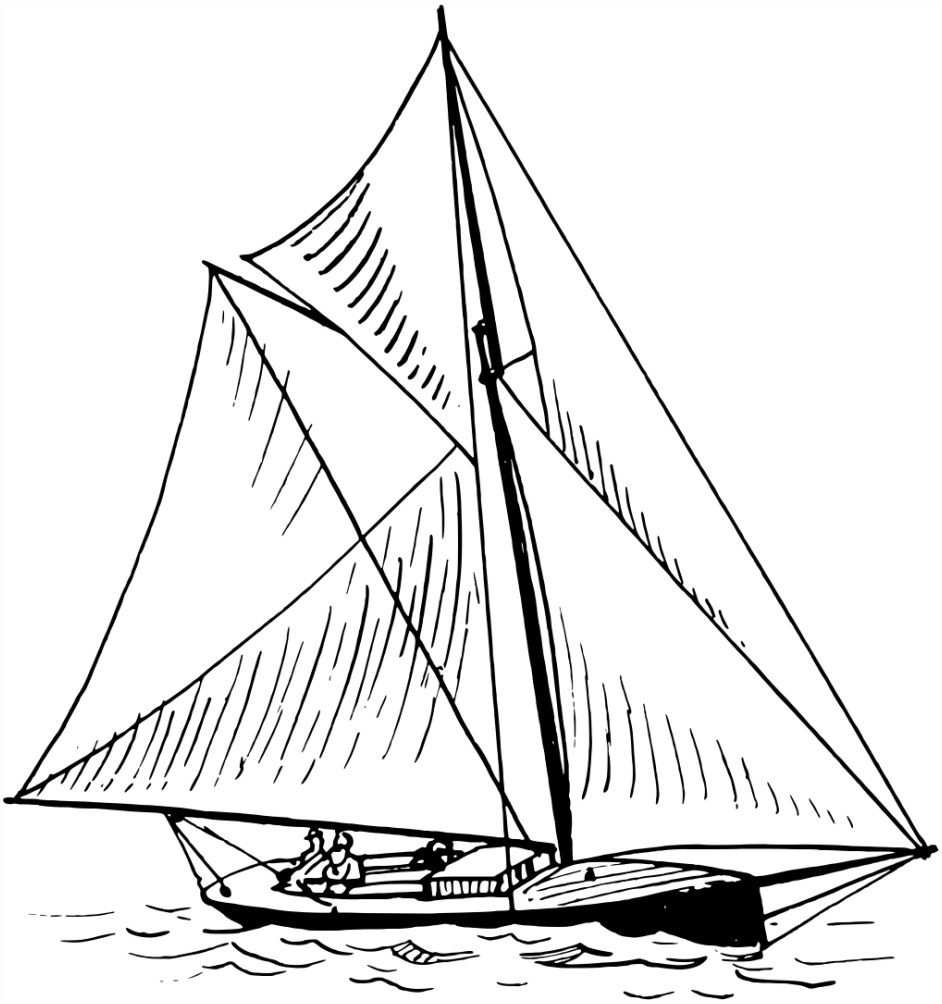

Figure 4.13:

Optionally, make other changes in the text editor and when you're done, commit your changes to the "remote repository" by clicking the green `Commit changes` button.

The book will now be rebuilt. Once that's done, you can view the result on the GitHub page.

If we want to add a figure to our book, we can refer to another website (as in the figure above). However, this carries the risk that the figure will no longer be visible if it is moved from its location. It is therefore better to have the figure as a local source file.

So, we first need to upload a figure to GitHub and then refer to that figure in our file. (Note! This will become much easier later.)

Warning

The code is case sensitive. So it matters whether your extension is .png or .PNG

Tip

We have a page with all important codes.

You can find more information about figure options here.

You can position figures in different places (left / center / right / margin), adjust the size, add a caption, etc. Check the documentation above and try out the different settings.

{MyST}

Exercise 4.14:

On GitHub, under the `code` tab...

1. Navigate to book/figures and click on `add file` → `Upload files`.

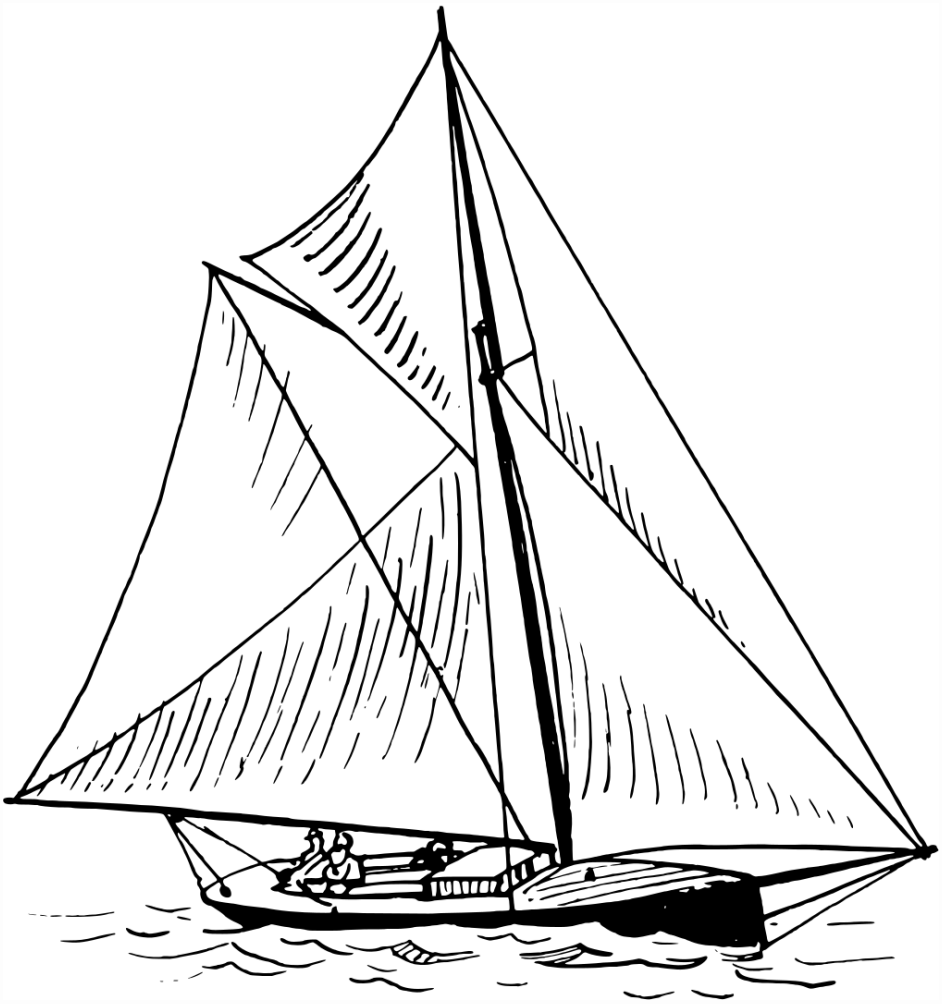

Figure 4.15:

4.2.1.3 width: 50%

add file in the folder

2. Choose the figure you want to add (remember the file name!).
3. Commit your changes to GitHub (the file will be uploaded).
4. Navigate to book and open `intro.md` and click `edit this file`.
5. Copy the code below into that file, and change the figure name to your own figure's name.

````
``` {figure} figures/incl_fig.PNG
---
width: 50%
name: fig_myfirstfigure
---
add file in the folder
```
````

6. Commit your change and view the result on GitHub pages.

{MyST}

Exercise 4.16: Add an equation

Take a look at the Cheatsheet page to see how to add a formula and give it a try...

Exercise 4.17: Other changes

Try making some other changes, for example by further developing the structure of the page into sections and subsections, each with some text. View the result.

4.2.1.4 Your favorite equation

4.2.1.5 Other changes

4.3 Lesson 2: Include awesomeness

4.3.1 Embed video from YouTube using iframe

4.3.2 Include figure

4.3.3 Include python code using code-block

4.4 Lesson 3: Interactive python plot

4.5 Creating your landing page using CSS

5. Advanced start

The Jupyter Book and MyST CLIs both include tools to help get started writing quickly. If you are feeling confident and want to jump straight into creating a document without the template, this is a good option.

You might want to follow this approach if,

- You are already familiar with using Markdown dialects to build documents and websites
- You have previously used Jupyter Book 1
- You are comfortable working with command line tools and reading documentation

The MyST Markdown guide will be a valuable resource if you follow this route.

5.1 Install Jupyter Book

There are a number of options for installing Jupyter Book. Two common methods are using `pip`

```
pip install "jupyter-book>=2.0.0a0"
```

or `npm`

```
npm install -g "jupyter-book@>=2.0.0-a0"
```

> **Hint**
>
> Note the version specifications, which ensure the MyST-based Jupyter Book 2 is installed, not the Sphinx-based Jupyter Book 1.

5.2 Initialise a project

To initialise a Jupyter Book project, use the CLI

```
jupyter-book init
```

This will create a basic book, including the `myst.yml` file. You can serve your book locally

```
jupyter-book start
```

Open http://localhost:3000 in your browser to see the output.

5.3 Deploy to GitHub pages

The CLI can also prepare a GitHub workflow to deploy your book to GitHub Pages, similar to the template.

```
jupyter-book init --gh-pages
```

The command will print instructions to finish setting up GitHub Pages.

5.4 Next steps

Once you have a working book, you can start to explore the options of MyST and Jupyter Book. You may want to follow the lessons here or experiment yourself with the MyST Markdown guide.

6. Software

References

Pols, F. (2021). What's inside the pink box? A nature of science activity for teachers and students. *Physics Education*, *56*(4), 45004. https://doi.org/10.1088/1361-6552/abf208