Contents

Whether you write your book's content in Jupyter Notebooks (`.ipynb`) or in regular markdown files (`.md`), you'll write in the same flavor of markdown called **MyST Markdown**. This is a simple file to help you get started and show off some syntax.

What is MyST?

MyST stands for "Markedly Structured Text". It is a slight variation on a flavor of markdown called "CommonMark" markdown, with small syntax extensions to allow you to write **roles** and **directives** in the Sphinx ecosystem.

For more about MyST, see the MyST Markdown Overview.

Sample Roles and Directives

Roles and directives are two of the most powerful tools in Jupyter Book. They are kind of like functions, but written in a markup language. They both serve a similar purpose, but **roles are written in one line**, whereas **directives span many lines**. They both accept different kinds of inputs, and what they do with those inputs depends on the specific role or directive that is being called.

Here is a "note" directive:

Note

Here is a note

Skip to main content

Here is an inline directive to refer to a document: Notebooks with MyST Markdown.

Citations

You can also cite references that are stored in a `bibtex` file. For example, the following syntax: `` {cite}`holdgraf_evidence_2014` `` will render like this: [HdHPK14].

Moreover, you can insert a bibliography into your page with this syntax: The `{bibliography}` directive must be used for all the `{cite}` roles to render properly. For example, if the references for your book are stored in `references.bib`, then the bibliography is inserted with:

[dVBSFCustodio22] Mauricio de Vasconcelos Barros, Frederico Schardong, and Ricardo Felipe Custódio. Leveraging self-sovereign identity, blockchain, and zero-knowledge proof to build a privacy-preserving vaccination pass. *Blockchain, and Zero-Knowledge Proof to Build a Privacy-Preserving Vaccination Pass*, 2022.

[HdHPK14] Christopher Ramsay Holdgraf, Wendy de Heer, Brian N. Pasley, and Robert T. Knight. Evidence for Predictive Coding in Human Auditory Cortex. In *International Conference on Cognitive Neuroscience*. Brisbane, Australia, Australia, 2014. Frontiers in Neuroscience.

[JC23] **missing journal in world_bank_2023**

[SHU+22] Mohammed Shuaib, Noor Hafizah Hassan, Sahnius Usman, Shadab Alam, Surbhi Bhatia, Arwa Mashat, Adarsh Kumar, and Manoj Kumar. Self-sovereign identity solution for blockchain-based land registry system: a comparison. *Mobile Information Systems*, 2022:1–17, 2022.

Learn more

This is just a simple starter to get you started. You can learn a lot more at jupyterbook.org.

You can also create content with Jupyter Notebooks. This means that you can include code blocks and their outputs in your book.

Markdown + notebooks

As it is markdown, you can embed images, HTML, etc into your posts!

Skip to main content

{MyST}

Markedly Structured Text

You can also add_{math} and

$$math^{blocks}$$

or

$$\mathrm{mean} la_{tex}$$

$$mathblocks$$

But make sure you $Escape $your $dollar signs $you want to keep!

MyST markdown

MyST markdown works in Jupyter Notebooks as well. For more information about MyST markdown, check out the MyST guide in Jupyter Book, or see the MyST markdown documentation.

Code blocks and outputs

Skip to main content

Jupyter Book will also embed your code blocks and output in your book. For example, here’s some sample Matplotlib code:

```
from matplotlib import rcParams, cycler
import matplotlib.pyplot as plt
import numpy as np
plt.ion()
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 1
----> 1 from matplotlib import rcParams, cycler
      2 import matplotlib.pyplot as plt
      3 import numpy as np

ModuleNotFoundError: No module named 'matplotlib'
```

```
# Fixing random state for reproducibility
np.random.seed(19680801)

N = 10
data = [np.logspace(0, 1, 100) + np.random.randn(100) + ii for ii in range(N)]
data = np.array(data).T
cmap = plt.cm.coolwarm
rcParams['axes.prop_cycle'] = cycler(color=cmap(np.linspace(0, 1, N)))


from matplotlib.lines import Line2D
custom_lines = [Line2D([0], [0], color=cmap(0.), lw=4),
                Line2D([0], [0], color=cmap(.5), lw=4),
                Line2D([0], [0], color=cmap(1.), lw=4)]

fig, ax = plt.subplots(figsize=(10, 5))
lines = ax.plot(data)
ax.legend(custom_lines, ['Cold', 'Medium', 'Hot']);
```

There is a lot more that you can do with outputs (such as including interactive outputs) with your book. For more information about this, see the Jupyter Book documentation

Jupyter Book also lets you write text-based notebooks using MyST Markdown. See the Notebooks with MyST Markdown documentation for more detailed instructions. This page shows off a notebook written in MyST Markdown.

With MyST Markdown, you can define code cells with a directive like so:

```
print(2 + 2)
```

```
4
```

When your book is built, the contents of any `{code-cell}` blocks will be executed with your default Jupyter kernel, and their outputs will be displayed in-line with the rest of your content.

See also

Jupyter Book uses Jupytext to convert text-based files to notebooks, and can support many other text-based notebook files.

Create a notebook with MyST Markdown

MyST Markdown notebooks are defined by two things:

1. YAML metadata that is needed to understand if / how it should convert text files to notebooks (including information about the kernel needed). See the YAML at the top of this page for example.
2. The presence of `{code-cell}` directives, which will be executed with your book.

That's all that is needed to get started!

Quickly add YAML metadata for MyST Notebooks

If you have a markdown file and you'd like to quickly add YAML metadata to it, so that Jupyter Book will treat it as a MyST Markdown Notebook, run the following command:

```
jupyter-book myst init path/to/markdownfile.md
```

Introduction

Skip to main content

As per World Bank estimates, nearly 850 million people lack an official identity [JC23] and proliferation of digital devices has made it increasingly essential to possess a verifiable digital identity. This has led to a rise in the number of digital transactions and the need for a secure and reliable identity management system. Self-sovereign identity (SSI) is a decentralized identity management system that allows individuals to control their own digital identities. It is a decentralized alternative to the traditional centralised identity management system in which identities are cryptographically verifiable. This allows users to have full control over their digital identities and prevents unauthorized access to their personal information.

Alternative text for the image

Your caption here

Ongoing Research in the field of SSI

SSI and Healthcare

Barros et. al. [dVBSFCustodio22] explore the concepts of self-sovereign identity, blockchain technology and zero-knowledge proofs to propose a solution for presenting proof of vaccination without revealing users' identities. It allows collaboration between health organisations, governments and other stakeholders such as tourism or travel industries. Furthermore, it uses interoperable open-source tools across countries so that this system can be implemented globally at a minimal cost from each country's government.

Land Registry using SSI

The land registry serves as a vital foundation for a country's economic development during nation-building. By documenting land-related information on a blockchain, this technology can enhance security and transparency within the land registry process.Shuaib et. al. [SHU+22] suggest that a blockchain-based land registry system can be combined with a self-sovereign identity (SSI) solution to provide a secure and efficient identity management system for landowners. They evaluate three existing self-sovereign identity solutions for blockchain-based land registry systems: Everest, Evernym, and uPort. These solutions are analyzed based on the self-sovereign identity principles to determine their compliance and effectiveness in addressing identity problems in land registry systems.This can help to

Skip to main content

ensure that only authorized individuals can access and modify land registry records, thereby reducing the risk of fraud and errors.

SSI and Voting

Conclusion

[dVBSFCustodio22] Mauricio de Vasconcelos Barros, Frederico Schardong, and Ricardo Felipe Custódio. Leveraging self-sovereign identity, blockchain, and zero-knowledge proof to build a privacy-preserving vaccination pass. *Blockchain, and Zero-Knowledge Proof to Build a Privacy-Preserving Vaccination Pass*, 2022.

[HdHPK14] Christopher Ramsay Holdgraf, Wendy de Heer, Brian N. Pasley, and Robert T. Knight. Evidence for Predictive Coding in Human Auditory Cortex. In *International Conference on Cognitive Neuroscience*. Brisbane, Australia, Australia, 2014. Frontiers in Neuroscience.

[JC23] **missing journal in world_bank_2023**

[SHU+22] Mohammed Shuaib, Noor Hafizah Hassan, Sahnius Usman, Shadab Alam, Surbhi Bhatia, Arwa Mashat, Adarsh Kumar, and Manoj Kumar. Self-sovereign identity solution for blockchain-based land registry system: a comparison. *Mobile Information Systems*, 2022:1–17, 2022.