# Welcome to your Jupyter Book

## Contents



This is a small sample book to give you a feel for how book content is structured. It shows off a few of the major file types, as well as some sample content. It does not go in-depth into any particular topic - check out the Jupyter Book documentation for more information.

Check out the content pages bundled with this sample book to see more.

## Markdown Files

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

It will be rendered in a special box when you build your book.

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:

**[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.

## Learn more

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

# Content with notebooks

Skip to main content

# Markdown + notebooks

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


You can also $add_{math}$ and

$$math^{blocks}$$

or

$$\text{mean} la_{tex}$$

$$mathblocks$$

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.

Skip to main content

## Code blocks and outputs

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()
```

```
# 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

# Notebooks with MyST Markdown

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.

## An example cell

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

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
```

# Installation Guide for Jupyter Labs and Jupyter Books {-}

In this guide we will outline the process in installing Jupyter Labs and Jupyter Books, as well as nbconvert (used to convert Jupyter Notebooks to PDF's and HTML) and any packages required for their operation for

Skip to main content

While other methods exist, this guide will primarily be using Pythons pip installer to install most of the required packages. This will require using the command terminal.

You can open the command terminal in the following ways:

1. Open your computer's Start menu and search for 'Command Prompt'. You should see it appear in the list of searched items.
2. Right click your computer's Start menu. A list of options should appear; selecting 'Terminal' will open the command terminal.
3. Pressing '⊞Win + R' will open the 'Run Prompt' window. Type in 'cmd' and press ok, this will open the command terminal.

# Table of Contents {-}



# Python Installation {-}

NOTE: Jupyter Books requires Python Version >= 3.7 to operate. However, nbconvert has only been tested, and only supports, Python Versions 3.7 - 3.9. For this reason, it is recomended that if you will be required to convert your Jupyter Notebooks and books to PDF and HTML that you install Python Version 3.9.

You can download the newest version of Python here: Download Python

Download the installer and follow the instructions in order to install the version of Python you choose (preferably version 3.9 if required to convert Jupyter Materials to PDF and HTML).

As we will be using pip to install several required packages, it is important that installing Python is your

Skip to main content

# Pyppeteer Installation {-}

NOTE: Pyppeteer requires Python Version >= 3.6 to operate.

In order to convert Jupyter Books to PDF, you will need to install Pyppeteer. Pyppetteer will also be used if you wish to convert Jupyter Notebooks to PDF using the `--to webpdf` command.

This can be done using the following pip command in your command terminal:

```
pip install pyppeteer
```

The documentation for Pyppeteer can be found here: Pyppeteer Documentation

# TeX Installation {-}

In order to convert Jupyter Notebooks to PDF using the `--to pdf` command you will need to install a complete TeX environment.

Fortunatly, there are packages/installers to simplify this process:

The MiKTeX distribution can be found here: MiKTeX Distribution
Instructions for its installation can be found here: MiKTeX Installation Instructions

The TeX Live distribution and installation instructions can be found here: TeX Live Distribution

Only one of these distributions is required. If having trouble installing one of the distributions it is recomended you try the other.

# nbconvert Installation {-}

NOTE: Nbconvert requires Python Version between (and including) 3.7 and 3.9.

In order to convert Jupyter Notebooks to PDF you will need to install nbconvert.

This can be done using the following pip command in your command terminal:

Skip to main content

The documentation for nbconvert can be found here: nbconvert Documentation

## Jupyter Labs Installation {-}

In order to view, manage, and create your Jupyter Notebooks you will need to install either Jupyter Labs or Jupyter Notebook.

Jupyter Labs is prefered, as it comes with all the functionality of Jupyter Notebook while being easier to navigate.

You can install Jupyter Labs by using the following pip command in your command terminal:

```
pip install jupyterlab
```

Alternatively, you can install Jupyter Notebook by using the following pip command in your command terminal:

```
pip install notebook
```

Installation instructions for these installations can be found here: Jupyter Labs and Notebook Installation Documentation

## Jupyter Books Installation {-}

In order to build and create Jupyter Books, you will need to install the Jupyter Book package.

You can install Jupyter Book by using the following pip command in your command terminal:

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

Installation instructions for Jupyter Books can be found here: Jupyter Book Installation Documentation

## How to Convert Jupyter Notebooks and Jupyter Books to

Skip to main content

Both Jupyter Notebooks and Books can be converted in order to be viewed as either PDF's or HTML's versions of themselves.

Your command terminal must be working in the same directory as the Jupyter Notebook or Jupyter Book you wish to convert. An easy way to ensure this is to place the Jupyter Notebook or Jupyter Book you wish to convert on your desktop. You can then point your command terminal to your desktop by opening your command terminal and using the following command:

```
cd desktop
```

You should now have your command terminal directed towards your desktop.

Alternatively, the path to your Jupyter Notebook or Jupyter Book can be substituted for your Jupyter Notebook or Jupyter Books name in the following commands.

## Converting Jupyter Notebooks to HTML {-}

Jupyter Notebooks can be converted to HTML using the following command in your command terminal:

```
jupyter nbconvert --to html <YourNotebookName.ipynb>
```

Note that this requires that you have installed nbconvert.

Nbconvert provides three templates that can be used to style your html.

1. `--template lab` provides a full static HTML render of the Notebook. This is the default template that will be applied if no template is specified.
2. `--template classic` provides simplified HTML using the classic jupyter look and feel.
3. `--template basic` provides base HTML, rendering with minimal structure and styles.

These templates can be applied by appending them to your conversion command as follows:

```
jupyter nbconvert --to html <YourNotebookName.ipynb> --template <style>
```

Note that this requires you to have installed Jupyter Labs and nbconvert.

Skip to main content

# Converting Jupyter Notebooks to PDF {-}

There are two ways of converting Jupyter Notebooks to PDF currently

First is to use the following command in your command terminal:

```
jupyter nbconvert --to pdf <YourNotebookName.ipynb>
```

Note that this requires you to have installed Jupyter Labs and nbconvert along with the Tex environment.

More information can be found here: PDF Conversion of Jupyter Notebooks

Second is to use the following command in your command terminal:

```
jupyter nbconvert --to webpdf <YourNotebookName.ipynb>
```

Note that this requires you to have installed Jupyter Labs and nbconvert along with pyppeteer.

Using the `--to webpdf` argument generates a PDF by first rendering your Jupyter Notebooks to HTML, as such, the templates that can be used for converting to HTML can be used with the `--to webpdf` argument.

More information can be found here: WebPDF Conversion of Jupyter Notebooks

# Converting Jupyter Books to HTML {-}

Jupyter Books can be converted to HTML by 'building' the book.

This can be done by using the following command in your command terminal:

```
jupyter-book build <YourBookName>
```

Note that this requires you to have installed Jupyter Books.

More information can be found here: Build Your Jupyter Book

Skip to main content

# Converting Jupyter Books to PDF {-}

Jupyter Books can be converted to PDF by using the following command in your command terminal:

```
jupyter-book build <YourBookName> --builder pdfhtml
```

Note that requires you to have installed Jupyter Books and pyppeteer.

More information can be found here: PDF Conversion of Jupyter Books