

# Read Me

This document outlines the basic structure of the FEDEASLab documentation website, and the primary tools which are used for it's development.

The current repository (claudioperez/FEDEASdoc) contains the working source files from which FEDEASLab documentation is built. The static website files which are generated from the source in this repository are hosted in the separate claudioperez/FEDEASdocs repository, which is currently serving the website.

## Site structure

---

This section outlines 3 basic types of pages used by the website, and explains the different workflows they entail:

1. Pages which are automatically generated from comments in program source code.
2. Pages which are manually written for the purpose of explaining certain concepts.
3. Pages which are deeply linked to scripts or code, but also contain a lot of written content (e.g. examples).

### Type I: Text from code - API/Functions

These pages are automatically extracted from comments or docstrings in source code files, and serve the purpose of documenting specific function APIs.

#### To build API documentation from a FEDEASLab release

*Assuming no changes are made to the current `m2html` scripts.*

1. build docs with `m2html`
2. copy created directories into `docs/FEDEASLab/`
3. run find-replace from `'html'` to `'/'` on all `FEDEASLab.html` files
4. Change file extensions from `.html` to `.md`.
5. *delete* `<head>` section from files

### Type II: Pure text - Data Structures/Fundamentals

Type II pages describe aspects of FEDEASLab in detail beyond what is written in the code comments. These pages are not directly linked to any program files and are provided directly in either Markdown or HTML files.

## Latex documents

Markdown and HTML files can be automatically generated from Latex using tools like Pandoc. A simple example of this conversion is presented. If interest arises in this method, a system can be arranged to accomodate more complex document elements such as custom Latex definitions and Bibtex referencing.

## Data Structures

### Type III: Text with code - Examples

These pages are intimately linked with code, but contain substantial blocks of writing. Examples of this page type include annotated application demonstrations and documents with dynamic or data-driven figures. Ideal source file formats include notebooks (.ipynb, .mlx).

## Tool Overview

---

### Primary

1. **Github**

File sharing, version control and site serving (via Github Pages)

2. **Anaconda**

Provides an implementation of Python which is required to run some of the other tools (e.g. mkdocs, sphinx, plotting). In addition to Python, Anaconda also provides a package manager which allows the organized management of software and dependencies.

To install all project dependencies, run `conda env create -f environment.yml`  
- `conda config --add channels conda-forge` - `conda install mkdocs` -  
`conda install python-markdown-math` - `conda install mkdocs-bootswatch`

### Secondary

1. **JSON**

Also see yaml.

2. **Static site:**

- Dynamic Sites: (Wordpress)
- mkdocs (Md) vs Sphinx (RST)

- Markdown arguably has a larger developer base than RST with more tools - *arguably*.
- Markdown is easier/more intuitive than RST.

### 3. Markdown

- Read about markdown from its creator here
- Find a quick markdown reference here

## Extra

### 1. Pandoc

## Other notes

---

### Site layout

The layout of the website is specified in the file `/mkdocs.yml`. For more information go to [mkdocs.org](http://mkdocs.org).

### Project layout

### Site styling

Colors schemes and fonts used throughout the site are specified in the file `css/extra.css`.

## Commands

### Powershell / Cmd

Build site html files. `> mkdocs build`

Create a local server for real-time editing. `> mkdocs serve`

Convert all html files in the active directory to markdown. `> gci -r -i *.html |foreach{$md=__directoryname+"+"$.basename+".md";pandoc $.name -o $md}`

### Bash (Linux subsystem)

`make all`

`make doc-serve`

## Equation Handling

Details regarding equation handling and supported Latex commands can be found [here](#).

The displaying of equations typeset in Latex is handled by the python package `python-markdown-math`. This can be installed by running the following command in an Anaconda-activated command line environment.

```
conda install -c conda-forge python-markdown-math
```

[stackoverflow answer](#)

## YAML Notes

There are two groups of styles, block and flow. Block styles use indentation to denote nesting and scope within the document. In contrast, flow styles rely on explicit indicators to denote nesting and scope [3.2.3.1. Node Styles].

There are 5 styles of scalars in YAML: plain, single-quoted, double-quoted, literal, and folded

## Json Tools

- Schema Forms

## Inspiration

Layout references

- pandas- Sphinx
- Stripe
- pyramid
- seaborn
- Heroku
- colorlib
- numba - Sphinx
- julia