pubmatter

A typst library for parsing, normalizing and showing publication frontmatter Rowan Cockett [1,2,3]

[1]Curvenote Inc. , [2]Continuous Science Foundation , [3]Project Jupyter Abstract

Utilities for loading and working with authors, affiliations, abstracts, keywords and other frontmatter information common in scientific publications.

Our goal is to introduce standardized ways of working with this content to expose metadata to scientific publishers who are interested in using typst in a standardized way. The specification for this pubmatter is based on MyST Markdown and Quarto, and can load their YAML files directly.
typst package, open-science, standards

Loading Frontmatter

The frontmatter can contain all information for an article, including title, authors, affiliations, abstracts and keywords. These are then normalized into a standardized format that can be used with a number of `show` functions like `show-authors`. For example, we might have a YAML file that looks like this:

```yaml
author: Rowan Cockett
date: 2024/01/26
```

You can load that file with `yaml`, and pass it to the `load` function:

```
#let fm = pubmatter.load(yaml("pubmatter.yml"))
```

This will give you a normalized data-structure that can be used with the `show` functions for showing various parts of a document.

You can also use a `dictionary` directly:

```
#let fm = load((
  author: (
    (
      name: "Rowan Cockett",
      email: "rowan@curvenote.com",
      orcid: "0000-0002-7859-8394",
      affiliations: "Curvenote Inc.",
    ),
  ),
  date: datetime(year: 2024, month: 01, day: 26),
  doi: "10.1190/tle35080703.1",
))
#show-author-block(fm)
```

Rowan Cockett [1] [1]Curvenote Inc.

Normalized Frontmatter Object

The frontmatter object has the following normalized structure:

```yaml
title: content
subtitle: content
short-title: string             # alias: running-title, running-head
# Authors Array
# simple string provided for author is turned into ((name: string),)
authors:                        # alias: author
  - name: string
    url: string                 # alias: website, homepage
    email: string
    phone: string
    fax: string
    orcid: string               # alias: ORCID
    note: string
    corresponding: boolean      # default: `true` when email set
    equal-contributor: boolean  # alias: equalContributor, equal_contributor
    deceased: boolean
    roles: string[]             # must be a contributor role
    affiliations:               # alias: affiliation
      - id: string
        index: number
# Affiliations Array
affiliations:                   # alias: affiliation
  - string                      # simple string is turned into (name: string)
  - id: string
    index: number
    name: string
    institution: string         # use either name or institution
# Other publication metadata
open-access: boolean
license:                        # Can be set with a SPDX ID for creative commons
  id: string
  url: string
  name: string
doi: string                     # must be only the ID, not the full URL
date: datetime                  # validates from 'YYYY-MM-DD' if a string
citation: content
# Abstracts Array
# content is turned into ((title: "Abstract", content: string),)
abstracts:                      # alias: abstract
  - title: content
    content: content
```

Note that you will usually write the affiliations directly in line, in the following example, we can see that the output is a normalized affiliation object that is linked by the `id` of the affiliation (just the name if it is a string!).

```
#let fm = load((
  authors: (
    (
      name: "Rowan Cockett",
      affiliations: "Curvenote Inc.",
    ),
    (
      name: "Steve Purves",
      affiliations: ("Project Jupyter", "Curvenote Inc."),
    ),
  ),
))
#raw(lang:"yaml", yaml.encode(fm))
```

```yaml
authors:
- name: Rowan Cockett
  affiliations:
  - id: Curvenote Inc.
    index: 1
- name: Steve Purves
  affiliations:
  - id: Project Jupyter
    index: 2
  - id: Curvenote Inc.
    index: 1
affiliations:
- name: Curvenote Inc.
  id: Curvenote Inc.
  index: 1
- name: Project Jupyter
  id: Project Jupyter
  index: 2
date: 'datetime(year: 2025, month: 11, day: 5)'
citation: Cockett & Purves, 2025
```

API Documentation

pubmatter

```
doi-link(doi: str) -> content
```

Create a DOI link

```
#doi-link(doi: "10.1190/tle35080703.1")
```

> https://doi.org/10.1190/tle35080703.1

Parameters:

`doi` (`str` = `none`) – Only include the DOI identifier, not the URL

```
email-link(email: str, email-color) -> content
```

Create a mailto link with an email icon

```
#email-link(email: "rowan@curvenote.com")
```

> ✉

Parameters:

`email` (`str` = `none`) – Email as a string

```
get-corresponding-author(authors: fm array) -> dictionary
```

Get corresponding author

Returns the first author marked as corresponding author, or the first author with an email.

```
#let author = get-corresponding-author(authors)
```

Parameters:

`authors` (`fm` or `array`) – The frontmatter object or authors directly

`github-link(github: str) -> content`

Create a link to a GitHub profile with the GitHub icon.

```
#github-link(github: "rowanc1")
```

Parameters:

`github` (`str` = `none`) – GitHub username (no `@`)

`open-access-link(oa-color) -> content`

Create a link to Wikipedia with an OpenAccess icon.

```
#open-access-link()
```

`orcid-link(orcid: str, orcid-color) -> content`

Create a ORCID link with an ORCID logo

```
#orcid-link(orcid: "0000-0002-7859-8394")
```

iD

Parameters:

`orcid` (`str` = `none`) – Use an ORCID identifier with no URL, e.g. `0000-0000-0000-0000`

`ror-link(ror: str, ror-color) -> content`

Create a ROR link

```
#ror-link(ror: "02mz0e468")
```

ROR

Parameters:

`ror` (`str` = `none`) – Only include the ROR identifier, not the URL

`show-abstract-block(fm: fm) -> content`

Show abstract-block including all abstracts and keywords

```
#show-abstract-block(fm)
```

Abstract

Utilities for loading and working with authors, affiliations, abstracts, keywords and other frontmatter information common in scientific publications.

Our goal is to introduce standardized ways of working with this content to expose metadata to scientific publishers who are interested in using typst in a standardized way. The specification for this pubmatter is based on MyST Markdown and Quarto, and can load their YAML files directly.
typst package, open-science, standards

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-abstracts(fm: fm ) -> content
```

Show all abstracts (e.g. abstract, plain language summary)

```
#show-abstracts(fm)
```

Abstract

Utilities for loading and working with authors, affiliations, abstracts, keywords and other frontmatter information common in scientific publications.

Our goal is to introduce standardized ways of working with this content to expose metadata to scientific publishers who are interested in using typst in a standardized way. The specification for this pubmatter is based on MyST Markdown and Quarto, and can load their YAML files directly.

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-affiliations(show-ror: boolean , show-equal-contributor: boolean , separator: str ,
affiliations: fm array ) -> content
```

Show affiliations

```
#show-affiliations(affiliations)
```

[1]Curvenote Inc. ROR, [2]Continuous Science Foundation , [3]Project Jupyter

Parameters:

`show-ror` (`boolean` = `true`) – Show ror logo

`show-equal-contributor` (`boolean` = `true`) – Show equal contributor note

`separator` (`str` = `", "`) – Separator between affiliations

`affiliations` (`fm` or `array`) – The frontmatter object or affiliations directly

```
show-author-block(fm: fm ) -> content
```

Show author block, including author, icon links (e.g. ORCID, email, etc.) and affiliations

> ```
> #show-author-block(fm)
> ```
>
> Rowan Cockett [1,2,3] [1]Curvenote Inc. , [2]Continuous Science Foundation , [3]Project Jupyter

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-authors(
  show-affiliations: boolean ,
  show-orcid: boolean ,
  show-email: boolean ,
  show-github: boolean ,
  show-equal-contributor: boolean ,
  authors: fm  array
) -> content
```

Show authors

> ```
> #show-authors(authors)
> ```
>
> Rowan Cockett [1,2,3]

Parameters:

`show-affiliations` (`boolean` = `true`) – Show affiliations text

`show-orcid` (`boolean` = `true`) – Show orcid logo

`show-email` (`boolean` = `true`) – Show email logo

`show-github` (`boolean` = `true`) – Show github logo

`show-equal-contributor` (`boolean` = `true`) – Show equal contributor asterisk

`authors` (`fm` or `array`) – The frontmatter object or authors directly

```
show-copyright(fm: fm ) -> content
```

Show copyright

Function chose a short citation with the copyright year followed by the license text. If the license is a Creative Commons License, additional explainer text is shown.

> ```
> #show-copyright(fm)
> ```
>
> Copyright © 2024 Cockett. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International license, which enables reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator.

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-keywords(fm: fm ) -> content
```

Show keywords

```
#show-keywords(fm)
```

typst package, open-science, standards

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-license-badge(license-color, fm: fm ) -> content
```

Show license badge

Works for creative common license and other license.

```
#show-license-badge(load((license: "CC0")))
```

🅭🄍

```
#show-license-badge(load((license: "CC-BY-4.0")))
```

🅭🅯

```
#show-license-badge(load((license: "CC-BY-NC-4.0")))
```

🅭🅯🄏

```
#show-license-badge(load((license: "CC-BY-NC-ND-4.0")))
```

🅭🅯🄏⊜

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-spaced-content(spacer: content , content: array ) -> content
```

Create a spaced content array separated with a `spacer`.

The default spacer is | , and undefined elements are removed.

```
#show-spaced-content(("Hello", "There"))
```

Hello | There

Parameters:

`spacer` (`content` = `text(fill: gray)[#h(8pt) | #h(8pt)]`) – How to join the content

`content` (`array`) – The various things to going together

```
show-title(fm: fm ) -> content
```

Show title and subtitle

```
#show-title(fm)
```

> pubmatter
>
> A typst library for parsing, normalizing and showing publication frontmatter

Parameters:

`fm` (`fm`) – The frontmatter object

```
show-title-block(fm: fm ) -> content
```

Show title block - title, authors and affiliations

```
#show-title-block(fm)
```

> pubmatter
>
> A typst library for parsing, normalizing and showing publication frontmatter Rowan Cockett [1,2,3] ✉ [1]Curvenote Inc. , [2]Continuous Science Foundation , [3]Project Jupyter

Parameters:

`fm` (`fm`) – The frontmatter object

validate-frontmatter

- show-citation()

```
show-citation(show-year: boolean , fm: fm ) -> content
```

Create a short citation in APA format, e.g. Cockett *et al.*, 2023

Parameters:

`show-year` (`boolean` = `true`) – Include the year in the citation

`fm` (`fm`) – The frontmatter object