JOSS Paper Format

Submitted articles must use Markdown and must provide a metadata section at the beginning of the article. Format metadata using YAML, a human-friendly data serialization language (The Official YAML Web Site, 2022). The information provided is included in the title and sidebar of the generated PDF.

What should my paper contain?

Important

Begin your paper with a summary of the high-level functionality of your software for a non-specialist reader. Avoid jargon in this section.

JOSS welcomes submissions from broadly diverse research areas. For this reason, we require that authors include in the paper some sentences that explain the software functionality and domain of use to a non-specialist reader. We also require that authors explain the research applications of the software. The paper should be between 750-1750 words. Authors submitting papers significantly longer than 1750 words may be asked to reduce the length of their paper.

Your paper must include the following **required sections**:

- **Summary**: A description of the high-level functionality and purpose of the software for a diverse, *non-specialist audience*.
- **Statement of need**: A section that clearly illustrates the research purpose of the software and places it in the context of related work. This should clearly state what problems the software is designed to solve, who the target audience is, and its relation to other work.
- **State of the field**: A description of how this software compares to other commonly-used packages in the research area. If related tools exist, provide a clear "build vs. contribute" justification explaining your unique scholarly contribution and why existing alternatives are insufficient.
- **Software design**: An explanation of the trade-offs you weighed, the design/architecture you chose, and why it matters for your research application. This should demonstrate meaningful design thinking beyond a superficial code structure description.
- **Research impact statement**: Evidence of realized impact (publications, ex integrations) or credible near-term significance (benchmarks, reproducibl

latest

- **AI usage disclosure**: Transparent disclosure of any use of generative AI in the software creation, documentation, or paper authoring. If no AI tools were used, state this explicitly. If AI tools were used, describe how they were used and how the quality and correctness of AI-generated content was verified.

Your paper must also include:

- A list of the authors of the software and their affiliations, using the correct format (see the example below).
- A list of key references, including to other software addressing related needs. Note that the references should include full names of venues, e.g., journals and conferences, not abbreviations only understood in the context of a specific discipline.
- Mention (if applicable) a representative set of past or ongoing research projects using the software and recent scholarly publications enabled by it.
- Acknowledgement of any financial support.

As this list shows, JOSS papers are expected to contain a limited set of metadata (see example below), and sections for Summary, Statement of need, State of the field, Software design, Research impact statement, AI usage disclosure, Acknowledgements, and References. You can look at an example accepted paper. Given this format, a “full length” paper is not permitted, and software documentation such as API (Application Programming Interface) functionality should not be in the paper and instead should be outlined in the software documentation.

> Important
>
> Your paper will be reviewed by two or more reviewers in a public GitHub issue. Take a look at the review checklist and review criteria to better understand how your submission will be reviewed.

Article metadata

Names

Providing an author name is straight-forward: just set the `name` attribute. However, sometimes more control over the name is required.

Name parts

There are many ways to describe the parts of names; we support the following:

- given names,
- surname,
- dropping particle

latest

- and suffix.

We use a heuristic to parse names into these components. This parsing may produce the wrong result, in which case it is necessary to provide the relevant parts explicitly.

The respective field names are

- `given-names` (aliases: `given`, `first`, `firstname`)
- `surname` (aliases: `family`)
- `suffix`

The full display name will be constructed from these parts, unless the `name` attribute is given as well.

Particles

It's usually enough to place particles like "van", "von", "della", etc. at the end of the given name or at the beginning of the surname, depending on the details of how the name is used.

- `dropping-particle`
- `non-dropping-particle`

Literal names

The automatic construction of the full name from parts is geared towards common Western names. It may therefore be necessary sometimes to provide the display name explicitly. This is possible by setting the `literal` field, e.g., `literal: Tachibana Taki`. This feature should only be used as a last resort.

Example

```yaml
authors:
  - name: John Doe
    affiliation: '1'

  - given-names: Ludwig
    dropping-particle: van
    surname: Beethoven
    affiliation: '3'

  # not recommended, but common aliases can be used for name parts.
  - given: Louis
    non-dropping-particle: de
    family: Broglie
    affiliation: '4'
```

latest

```
authors:
  - name:
      given-names: Kari
      surname: Nordmann
```

Date

The date must be specified using the following format: `%e %B %Y`

```
date: 9 October 2024
```

Affiliations

Each affiliation requires an `index` and `name`.

Optionally, the Research Organization Registry (ROR) identifier for the top-level organization can be annotated with the `ror` key. Note that ROR does not include departments in its scope, so ROR annotations are typically made to the top-level organization.

```
authors:
  - name: Albert Krewinkel
    affiliation: "1, 2, 3"

affiliations:
  - index: 1
    name: Laboratório Nacional de Luz Síncrotron, Brazil
  - index: 2
    name: Gadjah Mada University, Indonesia
  - index: 3
    name: Technische Universitaet Hamburg, Germany
    ror: "04bs1pb34"
```

Internal references

The goal of Open Journals is to provide authors with a seamless and pleasant writing experience. Since Markdown has no default mechanism to handle document internal references, known as “cross-references”, Open Journals supports a limited set of LaTex commands. In brief, elements that were marked with `\label` and can be referenced with `\ref` and `\autoref`.

```
![View of coastal dunes in a nature reserve on Sylt, an island in
the North Sea. Sylt (Danish: *Slid*) is Germany's northernmost
island.](sylt.jpg){#sylt width="100%"}
```

latest

Tables and figures

Tables and figures can be referenced if they are given a *label* in the caption. In pure Markdown, this can be done by adding an empty span `[]{label="floatlabel"}` to the caption. LaTeX syntax is supported as well: `\label{floatlabel}`.

Link to a float element, i.e., a table or figure, with `\ref{identifier}` or `\autoref{identifier}`, where `identifier` must be defined in the float's caption. The former command results in just the float's number, while the latter inserts the type and number of the referenced float. E.g., in this document `\autoref{proglangs}` yields "\autoref{proglangs}", while `\ref{proglangs}` gives "???".

: Comparison of programming languages used in the publishing tool.

```
| Language | Typing          | Garbage Collected | Evaluation | Created |
|----------|:---------------:|:-----------------:|------------|---------|
| Haskell  | static, strong  | yes               | non-strict | 1990    |
| Lua      | dynamic, strong | yes               | strict     | 1993    |
| C        | static, weak    | no                | strict     | 1972    |
```

Equations

Cross-references to equations work similarly to those for floating elements. The difference is that, since captions are not supported for equations, the label must be included in the equation:

```
$$a^n + b^n = c^n \label{fermat}$$
```

Referencing, however, is identical, with `\autoref{eq:fermat}` resulting in "\autoref{eq:fermat}".

$$a^n + b^n = c^n$$

Authors who do not wish to include the label directly in the formula can use a Markdown span to add the label:

```
[$$a^n + b^n = c^n$$]{label="eq:fermat"}
```

latest

Markdown

Markdown is a lightweight markup language used frequently in software development and online environments. Based on email conventions, it was developed in 2004 by John Gruber and Aaron Swartz.

Inline markup

The markup in Markdown should be semantic, not presentations. The table below has some basic examples.

Markup	Markdown example	Rendered output
emphasis	`*this*`	*this*
strong emphasis	`**that**`	**that**
strikeout	`~~not this~~`	~~not this~~
subscript	`H~2~O`	H_2O
superscript	`Ca^2+^`	Ca^{2+}
underline	`[underline]{.ul}`	underline
small caps	`[Small Caps]{.sc}`	Small Caps
inline code	`` `return 23` ``	`return 23`

Links

Link syntax is `[link description](targetURL)`. E.g., this link to the Journal of Open Source Software is written as `[Journal of Open Source Software](https://joss.theoj.org/)`.

Open Journal publications are not limited by the constraints of print publications. We encourage authors to use hyperlinks for websites and other external resources. However, the standard scientific practice of citing the relevant publications should be followed regardless.

Grid tables

Grid tables are made up of special characters which form the rows and columns, and also change table and style variables.

Complex information can be conveyed by using the following features not found in other table styles:

latest

- adding footers
- using intra-cellular body elements
- creating multi-row headers

Grid table syntax uses the characters "-", "=", "|", and "+" to represent the table outline:

- Hyphens (-) separate horizontal rows.
- Equals signs (=) produce a header when used to create the row under the header text.
- Equals signs (=) create a footer when used to enclose the last row of the table.
- Vertical bars (|) separate columns and also adjusts the depth of a row.
- Plus signs (+) indicates a juncture between horizontal and parallel lines.

Note: Inserting a colon (:) at the boundaries of the separator line after the header will change text alignment. If there is no header, insert colons into the top line.

Sample grid table:

```
+-------------------+------------+----------+----------+
| Header 1          | Header 2   | Header 3 | Header 4 |
|                   |            |          |          |
+:=================:+:==========:+:========:+:========:+
| row 1, column 1   | column 2   | column 3 | column 4 |
+-------------------+------------+----------+----------+
| row 2             | cells span columns               |
+-------------------+------------+---------------------+
| row 3             | cells      | - body              |
+-------------------+ span rows  | - elements          |
| row 4             |            | - here              |
+===================+============+=====================+
| Footer                                               |
+===================+============+=====================+
```

Figures and images

To create a figure, a captioned image must appear by itself in a paragraph. The Markdown syntax for a figure is a link, preceded by an exclamation point (!) and a description.
Example:
`![This description will be the figure caption](path/to/image.png)`

In order to create a figure rather than an image, there must be a description included and the figure syntax must be the only element in the paragraph, i.e., it must be surrounded by blank lines.

Images that are larger than the text area are scaled to fit the page. You can give images an explicit height and/or width, e.g. when adding an image as part of a paragrap
Markdown `![Nyan cat](nyan-cat.png){height="9pt"}` includes the image sa

latest

Citations

Bibliographic data should be collected in a file `paper.bib`; it should be formatted in the BibLaTeX format, although plain BibTeX is acceptable as well. All major citation managers offer to export these formats.

Cite a bibliography entry by referencing its identifier: `[@upper1974]` will create the reference "(Upper 1974)". Omit the brackets when referring to the author as part of a sentence: "For a case study on writers block, see Upper (1974)." Please refer to the pandoc manual for additional features, including page locators, prefixes, suffixes, and suppression of author names in citations.

The full citation will display as

> Upper, D. 1974. "The Unsuccessful Self-Treatment of a Case of "Writer's Block"." *Journal of Applied Behavior Analysis* 7 (3): 497. https://doi.org/10.1901/jaba.1974.7-497a.

Mathematical formulæ

Mark equations and other math content with dollar signs ($). Use a single dollar sign ($) for math that will appear directly within the text. Use two dollar signs ($$) when the formula is to be presented centered and on a separate line, in "display" style. The formula itself must be given using TeX syntax.

To give some examples: When discussing a variable *x* or a short formula like

$$\sin \frac{\pi}{2}$$

we would write $x$ and

```
$\sin \frac{\pi}{2}$
```

respectively. However, for more complex formulæ, display style is more appropriate. Writing

```
$$\int_{-\infty}^{+\infty} e^{-x^2} \, dx = \sqrt{\pi}$$
```

will give us

$$\int_{-\infty}^{+\infty} e^{-x^2}.\, dx = \sqrt{\pi}$$

latest

Footnotes

Syntax for footnotes centers around the “caret” character `^`. The symbol is also used as a delimiter for superscript text and thereby mirrors the superscript numbers used to mark a footnote in the final text.

```
Articles are published under a Creative Commons license[^1].
Software should use an OSI-approved license.

[^1]: An open license that allows reuse.
```

The above example results in the following output:

> Articles are published under a Creative Commons license . Software should use an OSI-approved license.
>
> **Footnotes**
>
> An open license that allows reuse.

Note: numbers do not have to be sequential, they will be reordered automatically in the publishing step. In fact, the identifier of a note can be any sequence of characters, like `[^marker]`, but may not contain whitespace characters.

Blocks

The larger components of a document are called “blocks”.

Headings

Headings are added with `#` followed by a space, where each additional `#` demotes the heading to a level lower in the hierarchy:

```
# Section

## Subsection

### Subsubsection
```

Please start headings on the first level. The maximum supported level is 5, but paper authors are encouraged to limit themselves to headings of the first two or three leve

Fourth- and fifth-level subsections – like this one and the following heading – are supported by the system; however, their use is discouraged. Use lists instead of fourth- and fifth-level headings.

Lists

Bullet lists and numbered lists, a.k.a. enumerations, offer an additional method to present sequential and hierarchical information.

```
- apples
- citrus fruits
  - lemons
  - oranges
```

- apples
- citrus fruits
  - lemons
  - oranges

Enumerations start with the number of the first item. Using the the first two laws of thermodynamics as example,

```
0. If two systems are each in thermal equilibrium with a third, they are
   also in thermal equilibrium with each other.
1. In a process without transfer of matter, the change in internal
   energy, $\Delta U$, of a thermodynamic system is equal to the energy
   gained as heat, $Q$, less the thermodynamic work, $W$, done by the
   system on its surroundings. $$\Delta U = Q - W$$
```

Rendered:

0. If two systems are each in thermal equilibrium with a third, they are also in thermal equilibrium with each other.
1. In a process without transfer of matter, the change in internal energy, $\Delta U$, of a thermodynamic system is equal to the energy gained as heat, $Q$, less the thermodynamic work, $W$, done by the system on its surroundings.

$$\Delta U = Q - W$$

Checking that your paper compiles

JOSS uses Pandoc to compile papers from their Markdown form into a PDF. There are a few different ways you can test that your paper is going to compile properly for J

latest

GitHub Action

If you’re using GitHub for your repository, you can use the Open Journals GitHub Action to automatically compile your paper each time you update your repository.

The PDF is available via the Actions tab in your project and click on the latest workflow run. The zip archive file (including the `paper.pdf`) is listed in the run’s Artifacts section.

Docker

If you have Docker installed on your local machine, you can use the same Docker Image to compile a draft of your paper locally. In the example below, the `paper.md` file is in the `paper` directory. Upon successful execution of the command, the `paper.pdf` file will be created in the same location as the `paper.md` file:

```
docker run --rm \
    --volume $PWD/paper:/data \
    --user $(id -u):$(id -g) \
    --env JOURNAL=joss \
    openjournals/inara
```

Locally

The materials for the `inara` container image above are themselves open source and available in its own repository. You can clone that repository and run the `inara` script locally with `make` after installing the necessary dependencies, which can be inferred from the `Dockerfile`.

Behind the scenes

Readers may wonder about the reasons behind some of the choices made for paper writing. Most often, the decisions were driven by radical pragmatism. For example, Markdown is not only nearly ubiquitous in the realms of software, but it can also be converted into many different output formats. The archiving standard for scientific articles is JATS, and the most popular publishing format is PDF. Open Journals has built its pipeline based on pandoc, a universal document converter that can produce both of these publishing formats as well as many more.

A common method for PDF generation is to go via LaTeX. However, support for tagging – a requirement for accessible PDFs – is not readily available for LaTeX. The current method used ConTeXt, to produce tagged PDF/A-3.

latest