

Submitting a paper to JOSS

If you've already developed a fully featured research code, released it under an [OSI-approved license](#), and written good documentation and tests, then we expect that it should take perhaps an hour or two to prepare and submit your paper to JOSS. But please read these instructions carefully for a streamlined submission.

Submission requirements

- The software must be open source as per the [OSI definition](#).
- The software must be hosted at a location where users can browse the source code files, open issues, and propose code changes without manual approval of (or payment for) accounts
- The software must have an **obvious** research application.
- You must be a major contributor to the software you are submitting, and have a GitHub account to participate in the review process.
- Your paper must not focus on new research results accomplished with the software.
- Your paper (`paper.md` and BibTeX files, plus any figures) must be hosted in a Git-based repository together with your software.
- The paper may be in a short-lived branch which is never merged with the default, although if you do this, make sure this branch is *created* from the default so that it also includes the source code of your submission.

In addition, the software associated with your submission must:

- Be stored in a repository that can be cloned without registration.
- Be stored in a repository where the software source files are browsable online without registration.
- Have an issue tracker that is readable without registration.
- Permit individuals to create issues/file tickets against your repository.

What we mean by research software

JOSS publishes articles about research software. This definition includes software that: solves complex modeling problems in a scientific context (physics, mathematics, biology, medicine, social science, neuroscience, engineering); supports the functioning of research instruments or the execution of research experiments; extracts knowledge fi

Scope and significance

JOSS publishes articles about software that demonstrates clear research impact or credible scholarly significance. Your software should represent a meaningful contribution to the research community rather than being a one-off tool for a single analysis. Whether or not you use AI assistance in your development, submissions should demonstrate the irreplaceable human contributions: problem framing, key design decisions, thoughtful architectural choices, and practices that make software usable and sustainable for others.

Some factors that may be considered by editors and reviewers when evaluating scope and significance include:

- **Research impact:** Evidence of publications or analyses using the software, external adopters or integrations, or credible near-term significance demonstrated through comparative benchmarks and reproducible reference materials.
- **Design thinking:** Meaningful architectural decisions and trade-offs considered. We particularly value work that builds upon or extends existing software ecosystems rather than reinventing solutions where quality alternatives already exist.
- **Open development practices:** Sustained development over time with evidence of collaborative effort, public development history, comprehensive testing, clear documentation, and pathways for community contribution.
- Whether a potential user can easily install, understand, and test the software. (If your software is new, please be sure a colleague has tried it.)
- Whether the software is sufficiently useful that it is *likely to be cited* by other researchers in your domain.

Projects developed privately are not eligible until there is a public record of open development: at least six months of public history prior to submission, with evidence of releases, public issues and pull requests. A history of contributions and engagement from individuals beyond the original team, across organizations, is especially welcome, though not essential. We particularly value projects that show evidence of open development practices from early stages, rather than private development followed by public release.

In addition, JOSS requires that software should be feature-complete (i.e., no half-baked solutions), packaged appropriately according to common community standards for the programming language being used (e.g., [Python](#), [R](#)), and designed for maintainable extension (not one-off modifications of existing tools). “Minor utility” packages, including “thin” API clients, and single-function packages are not acceptable.

AI Usage Policy

Author use: The use of generative AI is permitted for most aspects of a JOSS

- **Tool use:** The tools/models used (and versions) and where they were used (code, paper text, docs).
- **The nature and scope of assistance:** e.g., code generation, refactoring, test scaffolding, copy-editing, drafting.
- **Confirmation of review:** Authors must assert that human authors reviewed, edited, validated all AI-assisted outputs and made the core design decisions.

AI is not allowed for conversational interactions between authors and editors or reviewers unless it is being used for translation purposes.

Authors remain fully responsible for the accuracy, originality, licensing, and ethical/legal compliance of all submitted materials. Failure to provide a complete and accurate disclosure of AI usage may be considered an ethical breach. Consequences can include desk rejection, mandatory revisions, and post-publication correction or withdrawal. In cases of intentional misrepresentation or non-disclosure, JOSS reserves the right to notify the authors' institutions, funders, and/or relevant professional or scholarly societies in accordance with standard research-integrity practices.

Reviewer use: Reviewers may use generative AI tools to assist with non-substantive tasks (e.g., grammar checks of their review text, contextualizing public materials and code snippets) and must disclose this briefly at the end of their review.

- **Accountability:** The reviewer – not the AI – owns the evaluation. All judgements, recommendations, and technical claims must be the reviewer's and verified by them.
- **Human-only judgements:** When reviewing JOSS submissions (code or papers), all evaluative decisions – scoring, accept/reject recommendations, and assessments of originality, novelty, correctness, significance, and policy/ethics compliance – must be made by the human reviewer. AI tools may assist with analysis, but they must not render or determine any verdicts.

A note on web-based software

Many web-based research tools are out of scope for JOSS due to a lack of modularity and challenges testing and maintaining the code. Web-based tools may be considered 'in scope' for JOSS, provided that they meet one or both of the following criteria: 1) they are built around and expose a 'core library' through a web-based experience (e.g., R/[Shiny](#) applications) or 2) the web application demonstrates a high-level of rigor with respect to domain modeling and testing (e.g., adopts and implements a design pattern such as [MVC](#) using a framework such as [Django](#)).

Similar to other categories of submission to JOSS, it's essential that any web-based tool can be tested easily by reviewers locally (i.e., on their local machine).

Co-publication of science, methods, and software

Sometimes authors prepare a JOSS publication alongside a contribution describing a science application, details of algorithm development, and/or methods assessment. In this circumstance, JOSS considers submissions for which the implementation of the software itself reflects a substantial scientific effort. This may be represented by the design of the software, the implementation of the algorithms, creation of tutorials, or any other aspect of the software. We ask that authors indicate whether related publications (published, in review, or nearing submission) exist as part of submitting to JOSS.

Other venues for reviewing and publishing software packages

Authors wishing to publish software deemed out of scope for JOSS have a few options available to them:

- Follow [GitHub's guide](#) on how to create a permanent archive and DOI for your software. This DOI can then be used by others to cite your work.
- Enquire whether your software might be considered by communities such as [rOpenSci](#) and [pyOpenSci](#).

Should I write my own software or contribute to an existing package?

While we are happy to review submissions in standalone repositories, we also review submissions that are significant contributions made to existing packages. It is often better to have an integrated library or package of methods than a large number of single-method packages.

Policies

Disclosure: All authors must disclose any potential conflicts of interest related to the research in their manuscript, including financial, personal, or professional relationships that may affect their objectivity. This includes any financial relationships, such as employment, consultancies, honoraria, stock ownership, or other financial interests that may be relevant to the research.

Acknowledgement: Authors should acknowledge all sources of financial support for the work and include a statement indicating whether or not the sponsor had any involvement in it.

Review process: Editors and reviewers must be informed of any potential conflicts of interest before reviewing the manuscript to ensure unbiased evaluation of the research.

Compliance: Authors who fail to comply with the COI policy may have their

Review and Update: This COI policy will be reviewed and updated regularly to ensure it remains relevant and effective.

Conflict of Interest policy for authors

An author conflict of interest (COI) arises when an author has financial, personal, or other interests that may influence their research or the interpretation of its results. In order to maintain the integrity of the work published in JOSS, we require that authors disclose any potential conflicts of interest at submission time.

Preprint Policy

Authors are welcome to submit their papers to a preprint server ([arXiv](#), [bioRxiv](#), [SocArXiv](#), [PsyArXiv](#) etc.) at any point before, during, or after the submission and review process.

Submission to a preprint server is *not* considered a previous publication.

Authorship

Purely financial (such as being named on an award) and organizational (such as general supervision of a research group) contributions are not considered sufficient for co-authorship of JOSS submissions, but active project direction and other forms of non-code contributions are. The authors themselves assume responsibility for deciding who should be credited with co-authorship, and co-authors must always agree to be listed. In addition, co-authors agree to be accountable for all aspects of the work, and to notify JOSS if any retraction or correction of mistakes are needed after publication.

Submissions using proprietary languages/development environments

We strongly prefer software that doesn't rely upon proprietary (paid for) development environments/programming languages. However, provided *your submission meets our requirements* (including having a valid open source license) then we will consider your submission for review. Should your submission be accepted for review, we may ask you, the submitting author, to help us find reviewers who already have the required development environment installed.

Submission Process

Before you submit, you should:

- Make your software available in an open repository (GitHub, Bitbucket, etc) and include an [OSI approved open source license](#)

- Make sure that the software complies with the [JOSS review criteria](#). In particular, your software should be full-featured, well-documented, and contain procedures (such as automated tests) for checking correctness.
- Write a short paper in Markdown format using `paper.md` as file name, including a title, summary, author names, affiliations, and key references. See our [example paper](#) to follow the correct format.
- (Optional) create a metadata file describing your software and include it in your repository. We provide [a script](#) that automates the generation of this metadata.

Submitting your paper

Submission is as simple as:

- Filling in the [short submission form](#)
- Waiting for the managing editor to start a pre-review issue over in the JOSS reviews repository: <https://github.com/openjournals/joss-reviews>

No submission fees

There are no fees for submitting or publishing in JOSS. You can read more about our [cost and sustainability model](#).

Review Process

After submission:

- An Associate Editor-in-Chief will carry out an initial check of your submission, and proceed to assign a handling editor.
- The handling editor will assign two or more JOSS reviewers, and the review will be carried out in the [JOSS reviews repository](#).
- Authors will respond to reviewer-raised issues (if any are raised) on the submission repository's issue tracker. Reviewer and editor contributions, like any other contributions, should be acknowledged in the repository.
- **JOSS reviews are iterative and conversational in nature.** Reviewers are encouraged to post comments/questions/suggestions in the review thread as they arise, and authors are expected to respond in a timely fashion.
- **Authors are expected to respond to reviewer feedback promptly.** We generally ask that authors respond to reviewer comments and questions within 2 weeks, and complete requested changes within 4-6 weeks, matching the commitment we ask of reviewers, unless otherwise negotiated with the editor. If you need more time (for example, if the changes requested are particularly substantial), please communicate this to the reviewers and editor on the review thread. Prolonged unresponsiveness may result in the paper being rejected due to lack of engagement.

- Authors and reviewers are asked to be patient when waiting for a response from an editor. Please allow a week for an editor to respond to a question before prompting them for further action.
- Upon successful completion of the review, authors will make a tagged release of the software, and deposit a copy of the repository with a data-archiving service such as [Zenodo](#) or [figshare](#), get a DOI for the archive, and update the review issue thread with the version number and DOI.
- After we assign a DOI for your accepted JOSS paper, its metadata is deposited with CrossRef and listed on the JOSS website.
- The review issue will be closed, and an automatic post from [@JOSS at Mastodon](#) will announce it!

If you want to learn more details about the review process, take a look at the [reviewer guidelines](#).

Confidential requests

Please write admin@theoj.org with confidential matters such as retraction requests, report of misconduct, and retroactive author name changes.

In the event of a name change request, the DOI will remain unchanged, and the paper will be updated without the publication of a correction notice. Please note that because JOSS submissions are managed publicly, updates to papers are visible in the public record (e.g., in the [JOSS papers repository](#) commit history).

JOSS will also update Crossref metadata.