# Contributing to XO

? Thanks for contributing to XO! ?

Please note that this project is released with a [Contributor Code of Conduct](code-of-conduct.md). By participating in this project you agree to abide by its terms.

## How can I contribute?

### Improve documentation

As a user of XO you're the perfect candidate to help us improve our documentation. Typo corrections, error fixes, better explanations, more examples, etc. Open issues for things that could be improved. Anything. Even improvements to this document.

### Improve issues

Some issues are created with missing information, not reproducible, or plain invalid. Help make them easier to resolve. Handling issues takes a lot of time that we could rather spend on fixing bugs and adding features.

### Give feedback on issues

We're always looking for more opinions on discussions in the issue tracker. It's a good opportunity to influence the future direction of XO.

### Submitting an issue

- The issue tracker is for issues. Use Stack Overflow for support.

- Search the issue tracker before opening an issue.

- Ensure you're using the latest version of XO.

- Use a clear and descriptive title.

- Include as much information as possible: Steps to reproduce the issue, error message, Node.js version, operating system, etc.

- The more time you put into an issue, the more we will.

- [The best issue report is a failing test proving it.](https://twitter.com/sindresorhus/status/579306280495357953)

### Submitting a pull request

- Non-trivial changes are often best discussed in an issue first, to prevent you from doing unnecessary work.

- For ambitious tasks, you should try to get your work in front of the community for feedback as soon as possible. Open a pull request as soon as you have done the minimum needed to demonstrate your idea. At this early stage, don't worry about making things perfect, or 100% complete. Add a [WIP] prefix to the title, and describe what you still need to do. This lets reviewers know not to nit-pick small details or point out improvements you already know you need to make.

- New features should be accompanied with tests and documentation.

- Don't include unrelated changes.

- Lint and test before submitting the pull request by running `$ npm test`.

- Make the pull request from a [topic branch](https://github.com/dchelimsky/rspec/wiki/Topic-Branches), not master.

- Use a clear and descriptive title for the pull request and commits.

- Write a convincing description of why we should land your pull request. It's your job to convince us. Answer "why" it's needed and provide use-cases.

- You might be asked to do changes to your pull request. There's never a need to open another pull request. [Just update the existing one.](https://github.com/RichardLitt/knowledge/blob/master/github/amending-a-commit-guide.md)