# Contributing

:+1::tada: First off, thanks for taking the time to contribute! :tada::+1:

It's people like you that make [todo.txt] such a great tool.

The following is a set of guidelines for contributing to [todo.txt] and its packages. These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project. In return, they should reciprocate that respect in addressing your issue, assessing changes, and helping you finalize your pull requests.

[todo.txt] is an open source project and we love to receive contributions from our community — you! There are many ways to contribute, from writing tutorials or blog posts, improving the documentation, submitting bug reports and feature requests or writing code which can be incorporated into [todo.txt] itself.

Please, don't use the issue tracker for support questions. Check whether our [Gitter.im] channel can help with your issue. Stack Overflow is also worth considering.

# Ground Rules

## Responsibilities

- Be welcoming to newcomers and encourage diverse new contributors from all backgrounds. See our [Code of Conduct].

- Ensure cross-platform compatibility for every change that's accepted. Windows, Mac, Linux.

- Create issues for any major changes and enhancements that you wish to make. Discuss things transparently and get community feedback.

- Don't add any classes to the codebase unless absolutely needed. Err on the side of using functions.

- Keep feature versions as small as possible, preferably one new feature per version.

# Your First Contribution

Unsure where to begin contributing? You can start by looking through these beginner and help-wanted issues:

- Beginner issues - issues which should only require a few lines of code, and a test or two.

- Help wanted issues - issues which should be a bit more involved than beginner issues.

Both issue lists are sorted by total number of comments. While not perfect, number of comments is a reasonable proxy for impact a given change will have.

At this point, you're ready to make your changes! Feel free to ask for help; everyone is a beginner at first :smile\_cat:

If a maintainer asks you to "rebase" your PR, they're saying that a lot of code has changed, and that you need to update your branch so it's easier to merge.

# Getting started

For something that is bigger than a one or two line fix:

1. Create your own fork of the code.

1. Do the changes in your fork.

1. If you like the change and think the project could use it:

- Be sure you have followed the code style for the project.

- Note the [Code of Conduct].

As a rule of thumb, changes are obvious fixes if they do not introduce any new functionality or creative thinking. As long as the change does not affect functionality, some likely examples include the following:

- Spelling / grammar fixes

- Typo correction, white space and formatting changes

- Comment clean up

- Bug fixes that change default return values or error codes stored in constants

- Adding logging messages or debugging output

- Changes to ‘metadata’ files like .gitignore, build scripts, etc.

- Moving source files from one directory or package to another

# How to report a bug

## Security Vulnerability

If you find a security vulnerability, do NOT open an issue. Get ahold of the maintainers personally.

In order to determine whether you are dealing with a security issue, ask yourself these two questions:

- Can I access something that's not mine, or something I shouldn't have access to?

- Can I disable something for other people?

If the answer to either of those two questions are "yes", then you're probably dealing with a security issue. Note that even if you answer "no" to both questions, you may still be dealing with a security issue, so if you're unsure, just email us directly.

## Bug

When filing an issue, make sure to answer these five questions:

1. What version of shell are you using (`echo $0` or `$(echo $SHELL) --version)`)?

1. What operating system and processor architecture are you using?

1. What did you do?

1. What did you expect to see?

1. What did you see instead?

# How to suggest a feature or enhancement

The [todo.txt] philosophy is to provide a plain-text, software-agnostic way to keep track of your tasks.

If you find yourself wishing for a feature that doesn't exist, you are probably not alone. There are bound to be others out there with similar needs. Many of the features that todo.txt-cli has today have been added because our users saw the need. Open an issue on our issues list on GitHub which describes the feature you would like to see, why you need it, and how it should work.

# Code review process

The core team looks at Pull Requests on a regular basis. After feedback has been given we expect responses within two weeks. After two weeks we may close the pull request if it isn't showing any activity.

# Community

You can chat with the core team on https://gitter.im/todotxt/.

[todo.txt]: https://github.com/todotxt/

[Code of Conduct]: ./CODE\_OF\_CONDUCT.md

[Gitter.im]: https://gitter.im/todotxt/