# Contributing

If you discover issues, have ideas for improvements or new features,

please report them to the [issue tracker][1] of the repository or

submit a pull request. Please, try to follow these guidelines when you

do so.

## Issue reporting

\* Check that the issue has not already been reported.

\* Check that the issue has not already been fixed in the latest code

(a.k.a. `master`).

\* Be clear, concise and precise in your description of the problem.

\* Open an issue with a descriptive title and a summary in grammatically correct,

complete sentences.

\* Include the output of `rubocop -V`:

```

$ rubocop -V

0.50.0 (using Parser 2.4.0.0, running on ruby 2.4.2 x86\_64-linux)

```

\* Include any relevant code to the issue summary.

## Pull requests

\* Read [how to properly contribute to open source projects on GitHub][2].

\* Fork the project.

\* If you're adding or making changes to cops, read the [Development docs](https://docs.rubocop.org/rubocop/development.html)

\* Use a topic/feature branch to easily amend a pull request later, if necessary.

\* Write [good commit messages][3].

\* Use the same coding conventions as the rest of the project.

\* Commit and push until you are happy with your contribution.

\* If your change has a corresponding open GitHub issue, prefix the commit message with `[Fix #github-issue-number]`.

\* Make sure to add tests for it. This is important so I don't break it

in a future version unintentionally.

\* Add an entry to the [Changelog](CHANGELOG.md) accordingly. See [changelog entry format](#changelog-entry-format).

\* Please try not to mess with the Rakefile, version, or history. If

you want to have your own version, or is otherwise necessary, that

is fine, but please isolate to its own commit so I can cherry-pick

around it.

\* Make sure the test suite is passing and the code you wrote doesn't produce

RuboCop offenses (usually this is as simple as running `bundle exec rake`).

\* [Squash related commits together][5].

\* Open a [pull request][4] that relates to \*only\* one subject with a clear title

and description in grammatically correct, complete sentences.

### Changelog entry format

Here are a few examples:

```

\* [#716](https://github.com/rubocop-hq/rubocop/issues/716): Fixed a regression in the auto-correction logic of `MethodDefParentheses`. ([@bbatsov][])

\* New cop `ElseLayout` checks for odd arrangement of code in the `else` branch of a conditional expression. ([@bbatsov][])

\* [#7542](https://github.com/rubocop-hq/rubocop/pull/7542): \*\*(Breaking)\*\* Move `LineLength` cop from `Metrics` department to `Layout` department. ([@koic][])

```

\* Mark it up in [Markdown syntax][6].

\* The entry line should start with `\* ` (an asterisk and a space).

\* If the change has a related GitHub issue (e.g. a bug fix for a reported issue), put a link to the issue as `[#123](https://github.com/rubocop-hq/rubocop/issues/123): `.

\* Describe the brief of the change. The sentence should end with a punctuation.

\* If this is a breaking change, mark it with `\*\*(Breaking)\*\*`.

\* At the end of the entry, add an implicit link to your GitHub user page as `([@username][])`.

\* If this is your first contribution to RuboCop project, add a link definition for the implicit link to the bottom of the changelog as `[@username]: https://github.com/username`.

[1]: https://github.com/rubocop-hq/rubocop/issues

[2]: https://www.gun.io/blog/how-to-github-fork-branch-and-pull-request

[3]: https://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html

[4]: https://help.github.com/articles/about-pull-requests

[5]: http://gitready.com/advanced/2009/02/10/squashing-commits-with-rebase.html

[6]: https://daringfireball.net/projects/markdown/syntax