# Contributing to Fuse.js

We'd love for you to contribute to our source code and to make Fuse.js better! Here are the guidelines we'd like you to follow:

- [Questions and Problems](#question)

- [Issues and Bugs](#issue)

- [Pull Request Submission Guidelines](#submit-pr)

## <a name="requests"></a> Questions, Bugs, Features

### <a name="question"></a> Got a Question or Problem?

Do not open issues for general support questions as we want to keep GitHub issues for bug reports and feature requests. You've got much better chances of getting your question answered on dedicated support platforms, the best being [Stack Overflow][stackoverflow].

### <a name="issue"></a> Found an Issue or Bug?

If you find a bug in the source code, you can help us by submitting an issue to our

[GitHub Repository][github]. Even better, you can submit a Pull Request with a fix.

When creating issues, it's important to follow common guidelines to make them extra clear. Here is a few links to help you achieve that:

- [GitHub Guides: Mastering Issues](https://guides.github.com/features/issues/)

- [Wiredcraft: How We Write Github Issues](https://wiredcraft.com/blog/how-we-write-our-github-issues/)

- [NYC Planning Digital: Writing Useful Github Issues](https://medium.com/nyc-planning-digital/writing-a-proper-github-issue-97427d62a20f)

## <a name="submit-pr"></a> Pull Request Submission Guidelines

Before you submit your pull request consider the following guidelines:

- Search [GitHub](https://github.com/krisk/Fuse/pulls) for an open or closed Pull Request that relates to your submission. You don't want to duplicate effort.

- Make your changes in a new git branch:

```shell

git checkout -b my-fix-branch master

```

- Create your patch commit, \*\*including appropriate test cases\*\*.

- Run `yarn lint` to check that you have followed the automatically enforced coding rules

- Commit your changes using a descriptive commit message that follows our

[commit message conventions][developers.commits]. Adherence to the

[commit message conventions][developers.commits] is required, because release notes are

automatically generated from these messages.

```shell

git commit -a

```

- Before creating the Pull Request, package and run all tests a last time:

```shell

yarn test

```

- Push your branch to GitHub:

```shell

git push origin my-fix-branch

```

- If we suggest changes, then:

- Make the required updates.

- Re-run the test suite to ensure tests are still passing.

- Commit your changes to your branch (e.g. `my-fix-branch`).

- Push the changes to your GitHub repository (this will update your Pull Request).

You can also amend the initial commits and force push them to the branch.

```shell

git rebase master -i

git push origin my-fix-branch -f

```

This is generally easier to follow, but separate commits are useful if the Pull Request contains

iterations that might be interesting to see side-by-side.

That's it! Thank you for your contribution!

#### After your pull request is merged

After your pull request is merged, you can safely delete your branch and pull the changes

from the main (upstream) repository:

- Delete the remote branch on GitHub either through the GitHub web UI or your local shell as follows:

```shell

git push origin --delete my-fix-branch

```

- Check out the master branch:

```shell

git checkout master -f

```

- Delete the local branch:

```shell

git branch -D my-fix-branch

```

- Update your master with the latest upstream version:

```shell

git pull --ff upstream master

```

[stackoverflow]: http://stackoverflow.com/questions/tagged/fuse.js

[github]: https://github.com/krisk/Fuse/issues

[developers.commits]: DEVELOPERS.md#commits