# Contribution

Yeay! You want to contribute to gremlins.js. That's amazing!

To smoothen everyone's experience involved with the project please take note of the following guidelines and rules.

## Found an Issue?

Thank you for reporting any issues you find. We do our best to test and make gremlins.js as solid as possible, but any reported issue is a real help.

Please follow these guidelines when reporting issues:

- Tag your issue with the tag `bug`

- Provide a short summary of what you are trying to do

- Provide the log of the encountered error if applicable

- Provide the exact version of gremlins.js. Check `npm ls gremlins.js` when in doubt

- Be awesome and consider contributing a [pull request](#want-to-contribute)

## Want to contribute?

You consider contributing changes to gremlins.js – we dig that!

Please consider these guidelines when filing a pull request:

- Follow the [Coding Rules](#coding-rules)

- Follow the [Commit Rules](#commit-rules)

- Make sure you rebased the current master branch when filing the pull request

- Squash your commits when filing the pull request

- Provide a short title with a maximum of 100 characters

- Provide a more detailed description containing

\_ What you want to achieve

\_ What you changed

\_ What you added

\_ What you removed

## Coding Rules

To keep the code base of gremlins.js neat and tidy the following rules apply to every change

- `eslint` and `prettier` is king, use `make lint` and `make format`

- Be awesome

## Commit Rules

To help everyone with understanding the commit history of gremlins.js the following commit rules are enforced. We follow the [conventional commit messages](https://www.conventionalcommits.org/en/v1.0.0/) convention in order to automate CHANGELOG generation and to automate semantic versioning.

For example:

- `feat: A new feature`

- `fix: A bug fix`

Commits types such as as `docs:`,`style:`,`refactor:`,`perf:`,`test:` and `chore:` are valid but have no effect on versioning. \*\*It would be great if you use them.\*\*

All commits message are going to be validated when they are created using husky hooks.

\*\*PRs that do not follow the commit message guidelines will not be merged.\*\*