# Contributor's Guide

First of all, thank you for contributing to Highcharts! :+1: :confetti\_ball:

This file outlines the guidelines for contributing source code to the Highcharts repository,

as well as writing good issue reports.

## Contents

[Reporting Bugs](#reporting-bugs)

[Suggesting Features and/or Enhancements](#suggesting-features-and-enhancements)

[Contributing Code](#contributing-code)

[Licensing And Legal](#licensing-and-legal)

[Style Guide](#style-guide)

[Pull Requests](#pull-requests)

## Reporting Bugs

We track bugs on the repository's [GitHub issue tracker](https://github.com/highcharts/highcharts/issues).

Before submitting a new bug report, please try to search existing (open and closed) issues to

see if the issue is already reported. If you find an existing open issue, please

add any additional information you might have on the bug to the existing ticket.

If you find a closed issue describing your problem, please open a new issue and

link to the closed one in your issue description (use `#<ticket number>`, e.g. `#1234`).

Please also follow the directions in our [ticket submission template](https://github.com/highcharts/highcharts/blob/master/ISSUE\_TEMPLATE.md).

### Writing a clear bug report

To help us find and resolve issues as quickly as possible, it's important that your

issue description contains a clear description of the problem.

Your issue should:

\* have a clear and descriptive title

\* include any `console.error` output related to the issue

\* describe the expected behaviour/output

\* link to a JSFiddle demo of the issue

\* if it's not reproducable in a minimal demo, explain what actions where done to trigger the bug

\* include the tested Highcharts and browser version(s)

## Suggesting Features and Enhancements

Features and enhancements can be suggested on [user voice](https://highcharts.uservoice.com/).

## Contributing Code

### Licensing And Legal

Please note that when you contribute code to the Highcharts repository that code

will be part of the Highcharts source base, which is a commercial product.

For this reason, submissions may not contain code that prevents or limits commercial usage in closed-source applications.

To ensure this, submissions must be licensed using one of the following licenses:

\* [Apache License 2.0](http://opensource.org/licenses/apache2.0)

\* [BSD Licence](http://www.opensource.org/licenses/BSD-3-Clause)

\* [MIT / X11 License](http://www.opensource.org/licenses/MIT)

\* [Mozilla Public License](http://www.opensource.org/licenses/MPL-2.0)

No other license is permitted for contributions.

A reference to the license and the author of the code \*must\* be present in a comment leading into the submitted code block,

in a way that makes it clear \*which code falls under the license, and where/who it came from\*.

As an example:

```

/\*\*

\* The following function is licensed under the MIT license.

\* Author: John Doe

\*/

function anMITLicensedFunction () {

<some awesome code in here>

}

```

### Style Guide

\*\*Commit Messages\*\*

Commits should use the following format:

\* past tense, e.g. `Fixed..` rather than `Fixes..`: this is because we use the commit messages to generate changelogs

\* try to limit the name to <=100 characters

\* fixes should refer to issues where applicable, i.e. `Fixed #1235, <description>`

\*\*JavaScript\*\*

We use the following JavaScript style:

\* spaces over tabs

\* 4 character indentation

\* max columns in a line is 80

\* top-level var declarations

\* no `console.\*`

\* single quotes

This style is enforced by ESLint, which is ran in a post-commit hook.

### Pull Requests

We utilize a pull request based workflow. This means that all work is done in

feature/fix branches, and then merged through pull requests.

We use a standard naming convention for these branches:

\* `feature/<description>`: the branch contains a new feature

\* `bugfix/<issuenumber>-<description>`: the branch contains a bugfix for an open issue

\* `docs/<description>`: the branch contains a fix to the documentation/doclets

Your pull request should:

\* use the branch name (or follow the branch naming convention if the PR is based on the master branch on a fork) as the title

\* contain a link to an open issue - if there is one - in the description

\* contain a description of what the pull request implements/fixes