# Contributing to Cytoscape.js

Cytoscape.js is an open source project, and we greatly appreciate any and all contributions.

A blog post is available [on blog.js.cytoscape.org](http://blog.js.cytoscape.org/2017/06/13/contributing/) geared towards first-time contributors with more in-depth instructions on the project's structure, the process of creating and merging changes to the code, and more.

If you'd like to contribute code to Cytoscape.js but you're not sure exactly what you'd like to implement, take a look at our [current milestones](https://github.com/cytoscape/cytoscape.js/milestones) to see what features we have planned in future --- or anything labelled [`help-wanted`](https://github.com/cytoscape/cytoscape.js/issues?q=is%3Aopen+is%3Aissue+label%3Ahelp-wanted). Of course, we also welcome your own ideas.

Our goal is to make Cytoscape.js as comprehensive as possible. Thank you for taking the time and effort to contribute to help make that happen!

## Submitting issues

Submit issues or feature requests to the [issue tracker](https://github.com/cytoscape/cytoscape.js/issues). If your issue pertains to an extension, you should file the issue on that extension's issue tracker instead.

Before submitting an issue, please ensure that the issue still exists in the latest version of the library. Because we follow semver, you can safely upgrade patch releases (x.y.\*\*z\*\*) and feature releases (x.\*\*y\*\*) without worry of breaking API changes.

Clearly describe your issue. List the steps necessary to reproduce your issue along with the corresponding code (preferably a JSBin, as that makes the issue less ambiguous and much faster to fix).

Make certain to mention the version of the library you are using and version of the browser/environment you are using.

## Where to put changes

New features go in the `unstable` branch, which is used for the next (breaking/major or feature/minor) version. Bugfixes go in the `master` branch for the next bugfix/patch version. This allows us to follow [semver](http://semver.org/) nicely.

To propose a change, [fork](https://help.github.com/articles/fork-a-repo/) the cytoscape.js repository on Github, make a change, and then submit a [pull request](https://help.github.com/articles/creating-a-pull-request/) so that the proposed changes can be reviewed.

The source is organised in relatively the same as the documentation, under `./src`. Try to maintain that organisation as best you can. You are free to create new files and `require()` them using ESM [`import`](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import) and [`export`](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/export).

## Code style

Cytoscape.js is transpiled with Babel, so ES2015/ES6+ language features can be used.

Use two spaces for indentation, and single-quoted strings are preferred. The main thing is to try to keep your code neat and readable. There isn't a strict styleguide; it's more important that your code is easily understood and well tested. We do use [eslint](http://eslint.org/), so you can use `eslint` in the terminal or use eslint support in your editor.

You can run `eslint --fix` to automatically format the code to more or less match the style we use. It will only catch basic things, though.

## Testing

Tests go in the `./test` directory, as Mocha tests usually do. They are just a flat list of `.js` files that Mocha runs. If your change is a bugfix, please add a test case that would fail without your fix. If your change is a new feature, please add tests accordingly.

If your change is visual/rendering-related, then Mocha tests are not pragmatic. Use the debug page in the `debug` directory to try out visual changes. That page contains a sidebar with buttons and dropdowns that make visual and interactive testing easy.

Please run `npm test` to make sure all the unit tests are passing before you make your pull request.

We also have support for running the Mocha tests in IE9+ and other old browsers. You can run the tests in a [Windows IE VM](https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/) while running `npm run watch:umd`. Go to `http://youripaddress:8081/test/ie.html` in IE to open the Mocha test page.