# Contributing

## Project Scope

Picturefill aims to polyfill the standard features of the `picture` element and related `srcset`, `sizes`, `type`, and `media` attributes (some of which work on `img` elements without a `picture` parent as well). Issues and contributions that fall outside of this scope are not likely to be accepted by the project. Some exceptions include workarounds that improve browser support for standard features.

## Submitting issues

While we always welcome new issues, please take a look through our open and closed issues to see if it's been brought up before. We're happy to reopen closed issues if it still persists.

### FAQ / Frequent Issues

\_Q\_: \*\*Firefox isn't updating the src on window resize.\*\*

\_A\_: This is a known issue, ever since firefox added partial support in Firefox 38. It's slated to be fixed in Firefox 46.

\_Q\_: \*\*I have an issue/question about the `picture` element/spec.\*\*

\_A\_: Since Picturefill is only a polyfill, there's a repo for general `picture` discussion [here](https://github.com/ResponsiveImagesCG/picture-element).

\_Q\_: \*\*I have a question about how to use `srcset`, `sizes`, or `picture`.\*\*

\_A\_: Understandable—it's a lot of stuff to use! There's a handy resource from Eric Portis on [srcset & sizes](https://ericportis.com/posts/2014/srcset-sizes/). Additionally, the [picturefill website](https://scottjehl.github.io/picturefill) has a lot of demos and walks through some edge cases and how to solve them.

## Working on Picturefill

### Code License

Picturefill is an open source project falling under the MIT License. By using, distributing, or contributing to this project, you accept and agree that all code within the Picturefill project are licensed under MIT license.

### Issue Discussion

General Picturefill discussion takes place in Slack, at [http://picturefill.slack.com](http://picturefill.slack.com). To join, send an email to `picturefill` at `ricg.io`. Discussion of issues pertaining to the responsive images specifications takes place in the [RICG IRC channel](irc://irc.w3.org:6665/#respimg) and [issue tracker](https://github.com/responsiveimagescg/picture-element/issues/).

You can connect to the Picturefill Slack channel [via IRC](https://slack.zendesk.com/hc/en-us/articles/201727913-Connecting-to-Slack-over-IRC-and-XMPP) (SSL enabled).

### Modifying the code

First, ensure that you have the latest [Node.js](http://nodejs.org/) or [io.js](http://iojs.org) and [npm](http://npmjs.org/) installed.

Test that Grunt's CLI is installed by running `grunt --version`. If the command isn't found, run `npm install -g grunt-cli`. For more information about installing Grunt, see the [getting started guide](http://gruntjs.com/getting-started).

1. Fork and clone the repo.

1. Run `npm install` to install all dependencies (including Grunt).

1. Run `grunt` to grunt this project.

Assuming that you don't see any red, you're ready to go. Just be sure to run `grunt` after making any changes, to ensure that nothing is broken.

### Development Workflow

\*\*Note\*\*: The `master` branch now represents the publishing branch. Anything on master should be publishable to npm and bower. When something goes into master, it gets published. Minor version branches (`2.2`, `2.3`, etc) also exist to capture in-progress work on a particular version of picturefill. Once work on a version branch is complete, it gets merged into master to be published.

1. If no issue already exists for the work you’ll be doing, create one to document the problem(s) being solved and self-assign.

1. Create a new branch based off of the minor version branch you're fixing (`2.3`, `2.2`, etc) — please don't work in the `master` branch directly. We recommend naming the branch to match the issue being addressed (`issue-777`).

1. Add failing tests for the change you want to make. Run `grunt` to see the tests fail.

1. Fix stuff.

1. Run `grunt` to see if the tests pass. Repeat steps 2-4 until done.

1. Open `test/\*.html` unit test file(s) in actual browsers to ensure tests pass everywhere.

1. Update the documentation to reflect any changes.

1. Push to your fork or push your issue-specific branch to the main repo, then submit a pull request against the minor version branch you're fixing. `2.2`, `2.3`, etc.

1. Once tested and +1’d by another team member (with no outstanding objections), self-merge into the branch specified in the pull request.

1. Once ready to release to the public, merge the minor version branch into master and publish to npm and create a tag for bower.

### Versioning

The rules of [semver](http://semver.org/) don’t necessarily apply here in the case of major releases (as a polyfill for a stable spec, we \_shouldn’t\_ have breaking changes), so we’re using the following:

1. MAJOR versions at maintaners’ discretion following significant changes to the codebase (e.g. refactoring core)

1. MINOR versions for backwards-compatible enhancements (e.g. performance improvements)

1. PATCH versions for backwards-compatible bug fixes (e.g. spec compliance bugs, support issues)

### Important notes

Please don't edit files in the `dist` subdirectory as they are generated via Grunt. You'll find source code in the `src` subdirectory!

#### Code style

Regarding code style like indentation and whitespace, \*\*follow the conventions you see used in the source already.\*\*

#### PhantomJS

While Grunt can run the included unit tests via [PhantomJS](http://phantomjs.org/), this shouldn't be considered a substitute for the real thing. Please be sure to test the `test/\*.html` unit test file(s) in \_actual\_ browsers.