# Contribution tips and guidelines

:+1::tada: We're excited you want to contribute! Read on! :tada::+1:

[Questions](#i-just-have-a-question) &bull;

[Pull requests](#pull-requests-are-always-welcome) &bull;

[Tackling issues](#tackling-an-existing-issue) &bull;

[Reporting bugs](#reporting-bugs-and-other-issues) &bull;

[Labels](#labels)

## I just have a question

Before you ask, check our [existing questions](https://github.com/marko-js/marko/issues?page=2&q=is%3Aissue+label%3Atype%3Aquestion&utf8=%E2%9C%93) to see if your question has already been answered. If not, go ahead an open an issue or join us in [Discord](https://discord.gg/RFGxYGs) to ask a question.

Please be sure to use [markdown code blocks](https://help.github.com/articles/creating-and-highlighting-code-blocks/) when posting code on GitHub or Discord:

````

```marko

<div>some marko ${code}</div>

```

```js

const some = js.code;

```

````

## Pull requests are always welcome

Not sure if that typo is worth a pull request? Found a bug and know how to fix it? Do it! We will appreciate it. Any significant improvement should be documented as [a GitHub issue](https://github.com/marko-js/marko/issues) before anybody starts working on it.

We are always thrilled to receive pull requests. We do our best to process them quickly. If your pull request is not accepted on the first try, don't get discouraged! We'll work with you to come to an acceptable solution.

Prior to merging your PR, you will need to sign the [JS Foundation CLA](https://cla.js.foundation/marko-js/marko). It's pretty straight-forward and only takes a minute. You can even sign it now if you're thinking about contributing.

> \*\*TIP:\*\* If you're new to GitHub or open source you can check out this [free course](https://egghead.io/courses/how-to-contribute-to-an-open-source-project-on-github) on how to contribute to an open source project.

### Running tests

Before submitting your PR, make sure that all new and previous tests pass and that [coverage](https://codecov.io/gh/marko-js/marko) has not decreased:

```

npm run test-coverage

```

While developing you can run a single test group and use [grep](https://mochajs.org/#-g---grep-pattern) to filter the tests:

```

npm test -- --grep=lifecycle

```

### Adding tests

Marko makes use of directory based test suites. Take a look at the `render` test suite:

<pre>

<a href="../test/">test/</a>

? <a href="../test/render/">render/</a>

? <a href="../test/render/fixtures/">fixtures/</a>

? attrs/

? <a href="../test/render/fixtures/for-tag/">for-tag/</a>

? <a href="../test/render/fixtures/for-tag/expected.html">expected.html</a>

? <a href="../test/render/fixtures/for-tag/template.marko">template.marko</a>

? <a href="../test/render/fixtures/for-tag/test.js">test.js</a>

? nested-tags/

? while-tag/

? <a href="../test/render/html.test.js">html.test.js</a>

</pre>

The `html.test.js` file will run and read all the directories under `render/fixtures` and for each directory (`attrs`, `for-tag`, etc.) it will run `test.js`, render `template.marko` and assert that it is equivalent to the content of `expected.html`.

To add a new test, you'll find the appropriate test suite, copy a fixture, and modify it to add the new test.

#### Skipping a test

A few of the tests suites use the same fixtures for multiple test scenarios. For example, the `component-browser` tests run once rendering the component in a browser environment and a second time rendering in a server environment, then hydrating in the browser.

For some tests, it might be necessary to skip the test in one of these scenarios. This is done by exporting a [`skip\_hydrate`](https://github.com/marko-js/marko/blob/e3df4936c83a5ef419e8186df14ffc6012fcbdcc/test/components-browser/fixtures/implicit-component/test.js#L10) (or similiarly named) property from the fixture. The value of the property should be a string explaining why the test is skipped.

#### Adding a failing test case

If you've discovered an issue and are able to reproduce it, but don't have a fix, consider submitting a PR with a failing test case. You can mark a fixture as expected to fail by appending exporting a [`fails`](https://github.com/marko-js/marko/blob/0833ada47eeb5c833a11ef01fcd53ae39b0b7491/test/render/fixtures/spread-attribute-function-object/test.js#L1) property from the fixture. The value of the `fails` property should be a string with the issue number. Upon merging a failing test case, a maintainer will update the corresponding issue to add the [`has failing test`](https://github.com/marko-js/marko/labels/has%20failing%20test) label.

In the case that a fixture is used in multiple test scenarios, you can mark the test as failing in a specific scenario by exporting a [`fails\_hydrate`](https://github.com/marko-js/marko/blob/e3df4936c83a5ef419e8186df14ffc6012fcbdcc/test/components-browser/fixtures-deprecated/widget-conditional/test.js#L19) (or similarly named) property from the fixture.

Expected failures won't cause [Travis CI](https://travis-ci.org/marko-js/marko) to report a error, but document that there is an issue and give others a starting point for fixing the problem.

### Debugging tests

If you need to dig a bit deeper into a failing test, use the `--inspect-brk` flag, open Chrome DevTools, and click on the green nodejs icon (<img height="16" src="https://user-images.githubusercontent.com/1958812/37050480-d53e4276-2128-11e8-8c7a-f5d842956c98.png"/>) to start debugging. Learn more about [debugging node](https://www.youtube.com/watch?v=Xb\_0awoShR8&t=103s) from this video.

```

npm test -- --grep=test-name --inspect-brk

```

In addition to [setting breakpoints](https://developers.google.com/web/tools/chrome-devtools/javascript/breakpoints), you can also add [`debugger;`](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/debugger) statements in both your JavaScript files and Marko templates:

```marko

$ debugger;

<div>Hello ${input.name}!</div>

```

### Updating snapshots

A number of the test suites make use snapshot comparisons. For example, the `render` tests compare the rendered html against a stored snapshot. Similarly, the `compiler` tests compare the generated JavaScript module againt a stored snapshot. Any changes compared to the snapshot should be looked at closely, but there are some cases where it is fine that the output has changed and the snapshot needs to be updated.

To update a snapshot, you can copy the contents from the `actual` file to the `expected` file in the fixture directory. You can also use the `UPDATE\_EXPECTATIONS` env variable to cause the test runner to update the `expected` file for all currently failing tests in a suite:

```

UPDATE\_EXPECTATIONS=1 npm test

```

## Tackling an existing issue

Comment on the issue and let us know you'd like to tackle it. If for some reason you aren't going to be able to complete the work, let us know as soon as you can so we can open it up for another developer to work on.

Here's some to get started with:

- [good first issue](https://github.com/marko-js/marko/issues?q=is%3Aissue+is%3Aopen+label%3A%22good+first+issue%22): great for new contributors

- [help wanted](https://github.com/marko-js/marko/issues?q=is%3Aissue+is%3Aopen+label%3A%22help+wanted%22) issues: won't be tackled in the near future by the maintainers... we need your help!

- [unassigned](https://github.com/marko-js/marko/issues?utf8=%E2%9C%93&q=is%3Aissue%20is%3Aopen%20no%3Aassignee%20) issues: open issues that no one has claimed... yet

- [has failing test](https://github.com/marko-js/marko/labels/has%20failing%20test) issues: open issues that already have a failing test case in the repo. make it pass!

## Reporting bugs and other issues

A great way to contribute to the project is to send a detailed report when you encounter an issue. Even better: submit a PR with a failing test case ([see how](#adding-a-failing-test-case)).

Check that [our issue database](https://github.com/marko-js/marko/issues) doesn't already include that problem or suggestion before submitting an issue. If you find a match, you can use the "subscribe" button to get notified on updates. Rather than leaving a "+1" or "I have this too" comment, you can add a [reaction](https://github.com/blog/2119-add-reactions-to-pull-requests-issues-and-comments) to let us know that this is also affecting you without cluttering the conversation. However, if you have ways to reproduce the issue or have additional information that may help resolving the issue, please leave a comment.

We have an [ISSUE\_TEMPLATE](ISSUE\_TEMPLATE.md) that will populate your textarea when you go to open an issue. Use the relevant section and remove the rest.

Please provide as much detail as possible.

### Reporting security issues

If you discover a security issue, please \*\*DO NOT\*\* file a public issue, instead \*\*privately\*\* send your report to one of the members with the `core team` role in the [Discord](https://discord.gg/RFGxYGs).

Security reports are greatly appreciated and we will publicly thank you for it. We currently do not offer a paid security bounty program.

# Labels

Once you post an issue, a maintainer will add one or more labels to it. Below is a guideline for the maintainers and anyone else who is interested in what the various labels mean.

### Type

![](https://img.shields.io/badge/type-bug-dd0000.svg)

![](https://img.shields.io/badge/type-unverified%20bug-aa3300.svg)

![](https://img.shields.io/badge/type-feature-0099dd.svg)

![](https://img.shields.io/badge/type-question-99cc00.svg)

![](https://img.shields.io/badge/type-community-ff8800.svg)

![](https://img.shields.io/badge/type-tech%20debt-bfdadc.svg)

![](https://img.shields.io/badge/type-docs-bbbbbb.svg)

Every issue should be assigned one of these.

- \*\*bug\*\*: A bug report

- \*\*unverified-bug\*\*: A bug report that has not been verified

- \*\*feature\*\*: A feature request

- \*\*question\*\*: A question about how to do something in Marko

- \*\*community\*\*: Related to community building, improving the contribution process, etc.

- \*\*tech debt\*\*: Related to refactoring code, test structure, etc.

- \*\*docs\*\*: Related to documentation/website

### Scope

![](https://img.shields.io/badge/scope-parser-5500cc.svg)

![](https://img.shields.io/badge/scope-compiler-cc0077.svg)

![](https://img.shields.io/badge/scope-runtime-eebb00.svg)

![](https://img.shields.io/badge/scope-core%20taglib-00cccc.svg)

![](https://img.shields.io/badge/scope-components-9900aa.svg)

![](https://img.shields.io/badge/scope-tools-fef2c0.svg)

What part of the Marko stack does this issue apply to? In most cases there should only be one of these.

- \*\*parser\*\*: Relates to [`htmljs-parser`](https://github.com/marko-js/htmljs-parser)

- \*\*compiler\*\*: Relates to the [compiler](../src/compiler) (server only)

- \*\*runtime\*\*: Relates to the [runtime](../src/runtime) (isomorphic/universal)

- \*\*core-taglib\*\*: Relates to [custom tags](../src/taglib) that ship with Marko

- \*\*components\*\*: Relates to [components](../src/components)

- \*\*tools\*\*: Relates to editor plugins, commandline tools, etc.

### Status

![](https://img.shields.io/badge/status-backlog-223344.svg)

![](https://img.shields.io/badge/status-in%20progress-006b75.svg)

![](https://img.shields.io/badge/status-needs%20review-0e8a16.svg)

In many cases, additional \_actions\_ should be taken when applying one of these.

- \*\*backlog\*\*: Tasks planned to be worked on

- \*\*in progress\*\*: This is currently being worked on.

- \*\*needs review\*\*: This issue needs to be followed up on.

### Reason closed

![](https://img.shields.io/badge/reason%20closed-resolved-99cc99.svg)

![](https://img.shields.io/badge/reason%20closed-duplicate-cc99cc.svg)

![](https://img.shields.io/badge/reason%20closed-declined-bb6666.svg)

![](https://img.shields.io/badge/reason%20closed-not%20a%20bug-997744.svg)

![](https://img.shields.io/badge/reason%20closed-inactivity-bfd4f2.svg)

![](https://img.shields.io/badge/reason%20closed-no%20issue-c5def5.svg)

- \*\*resolved\*\*: The question was answered, the bug was fixed, or the feature was implemented.

- \*\*duplicate\*\*: Someone has already posted the same or a very similar issue. A comment should be added that references the original issue.

- \*\*declined\*\*: This feature will not be implemented.

- \*\*not a bug\*\*: This is not a bug, but either user error or intended behavior.

- \*\*inactivity\*\*: There was not enough info to reproduce the bug or not enough interest in the feature to hash out an implementation plan and the conversation has stalled.

- \*\*no issue\*\*: This wasn't so much an issue as a comment

### Other

![](https://img.shields.io/badge/-good%20first%20issue-00cccc.svg)

![](https://img.shields.io/badge/-help%20wanted-33cc88.svg)

![](https://img.shields.io/badge/-blocked-6b0c0c.svg)

![](https://img.shields.io/badge/-needs%20more%20info-dd9944.svg)

![](https://img.shields.io/badge/-user%20land-e8c9c9.svg)

- \*\*good first issue\*\*: Small tasks that would be good for first time contributors.

- \*\*help wanted\*\*: Not on the roadmap, but we'd love for someone in the community to tackle it.

- \*\*blocked\*\*: Cannot be completed until something else happens first. This should be described in a comment with a link to the blocking issue.

- \*\*needs more info\*\*: The original poster needs to provide more information before action can be taken.

- \*\*user land\*\*: Something that probably won't be added to core, but could be implemented/proved out as a separate module.