# Contributing to Premake

Thanks for your interest in contributing to Premake! :tada: We love getting [pull requests](https://www.quora.com/GitHub-What-is-a-pull-request) and rely heavily on the contributions of our community to keep Premake healthy and growing.

We want to keep it as easy as possible to contribute changes. These guidelines are intended to help smooth that process, and allow us to review and approve your changes quickly and easily. Improvements are always welcome! Feel free to [open an issue][issue-tracker] or [submit a new pull request][submit-pr]. And finally, these are just guidelines, not rules, so use your best judgement when necessary.

We do everything in [Git][git] hosted on [GitHub][github]. If you're new to this environment, you may want to begin with [Getting Started with GitHub](https://help.github.com/en/categories/getting-started-with-github) and the Thinkful's [GitHub Pull Request Tutorial](https://www.thinkful.com/learn/github-pull-request-tutorial/).

## Reporting Bugs

Bugs should be reported on our [GitHub Issue Tracker][issue-tracker].

Follow the advice in [How do I ask a good question?][how-to-ask]. While the article is intended for people asking questions on [StackOverflow](https://stackoverflow.com/), it all applies to writing a good bug report too.

## Requesting New Features

Feature requests should also be sent to our [GitHub Issue Tracker][issue-tracker].

- Explain the problem that you're having, and anything you've tried to solve it using the currently available features

- Explain how this new feature will help

- If possible, provide an example, like a code snippet, showing what your new feature might look like in use

Also, much of the advice in [How do I ask a good question?][how-to-ask] applies here too.

## Contributing a Fix or Feature

You've created a new fix or feature for Premake. Awesome!

1. If you haven't already, create a fork of the Premake repository

2. Create a topic branch, and make all of your changes on that branch

3. Submit a pull request

4. Give us a moment. Premake is maintained by only a few people, all of whom are doing this on their limited free time, so it may take us a bit to review your request. We're working on improving our turnaround time with resources like this guide and [our OpenCollective][collective].

If you're not sure what any of that means, check out Thinkful's [GitHub Pull Request Tutorial](https://www.thinkful.com/learn/github-pull-request-tutorial/) for a complete walkthrough of the process.

### Writing a Good Pull Request

- Stay focused on a single fix or feature. If you submit multiple changes in a single request, we may like some but spot issues with others. When that happens, we have to reject the whole thing. If you submit each change in its own request it is easier for us to review and approve.

- Limit your changes to only what is required to implement the fix or feature. In particular, avoid style or formatting tools that may modify the formatting of other areas of the code. If your code editor supports [EditorConfig](https://editorconfig.org), turn it on to use [the .editorconfig settings](https://github.com/premake/premake-core/blob/master/.editorconfig) supplied with the Premake sources.

- Write tests! You don't need to go crazy, but we will expect a unit test or two to show that your fix or feature does what it says, and doesn't break in the future. There are many test examples in Premake's source code, covering both the [modules](https://github.com/premake/premake-core/tree/master/modules) and the [core](https://github.com/premake/premake-core/tree/master/tests). Feel free to copy!

- When you submit a change, try to limit the number of commits involved. A single commit is ideal.

- Follow our coding conventions, which we've intentionally kept quite minimal.

### Coding Conventions

- For symbols that will be visible to project script authors, follow the Lua all-lowercase standard for names: `dosomethingcool`. It's a terrible convention, but it helps us be consistent with Lua's core libraries. Everywhere else, use the much more readable camel case: `doSomethingCool`. (We know this is confusing, and may revisit it in a future major release.)

- Use tabs for indentation, not spaces

- Use Unix (LF) end-of-line sequence

- When in doubt, match the code that's already there

[collective]: https://opencollective.com/premake

[git]: https://git-scm.com

[github]: https://github.com

[how-to-ask]: https://stackoverflow.com/help/how-to-ask

[issue-tracker]: https://github.com/premake/premake-core/issues

[submit-pr]: https://github.com/premake/premake-core/pulls