# Contributing to Jekyll

Hi there! Interested in contributing to Jekyll? We'd love your help. Jekyll is an open source project, built one contribution at a time by users like you.

## Where to get help or report a problem

See the [support guidelines](https://jekyllrb.com/docs/support/)

## Ways to contribute

Whether you're a developer, a designer, or just a Jekyll devotee, there are lots of ways to contribute. Here's a few ideas:

\* [Install Jekyll on your computer](https://jekyllrb.com/docs/installation/) and kick the tires. Does it work? Does it do what you'd expect? If not, [open an issue](https://github.com/jekyll/jekyll/issues/new) and let us know.

\* Comment on some of the project's [open issues](https://github.com/jekyll/jekyll/issues). Have you experienced the same problem? Know a work around? Do you have a suggestion for how the feature could be better?

\* Read through the [documentation](https://jekyllrb.com/docs/home/), and click the "improve this page" button, any time you see something confusing, or have a suggestion for something that could be improved.

\* Browse through the [Jekyll discussion forum](https://talk.jekyllrb.com/), and lend a hand answering questions. There's a good chance you've already experienced what another user is experiencing.

\* Find an [open issue](https://github.com/jekyll/jekyll/issues) (especially [those labeled `help-wanted`](https://github.com/jekyll/jekyll/issues?q=is%3Aopen+is%3Aissue+label%3Ahelp-wanted)), and submit a proposed fix. If it's your first pull request, we promise we won't bite, and are glad to answer any questions.

\* Help evaluate [open pull requests](https://github.com/jekyll/jekyll/pulls), by testing the changes locally and reviewing what's proposed.

## Submitting a pull request

### Pull requests generally

\* The smaller the proposed change, the better. If you'd like to propose two unrelated changes, submit two pull requests.

\* The more information, the better. Make judicious use of the pull request body. Describe what changes were made, why you made them, and what impact they will have for users.

\* If this is your first pull request, it may help to [understand GitHub Flow](https://guides.github.com/introduction/flow/).

\* If you're submitting a code contribution, be sure to read the [code contributions](#code-contributions) section below.

### Submitting a pull request via github.com

Many small changes can be made entirely through the github.com web interface.

1. Navigate to the file within [`jekyll/jekyll`](https://github.com/jekyll/jekyll) that you'd like to edit.

2. Click the pencil icon in the top right corner to edit the file

3. Make your proposed changes

4. Click "Propose file change"

5. Click "Create pull request"

6. Add a descriptive title and detailed description for your proposed change. The more information the better.

7. Click "Create pull request"

That's it! You'll be automatically subscribed to receive updates as others review your proposed change and provide feedback.

### Submitting a pull request via Git command line

1. Fork the project by clicking "Fork" in the top right corner of [`jekyll/jekyll`](https://github.com/jekyll/jekyll).

2. Clone the repository locally `git clone https://github.com/<you-username>/jekyll`.

3. Create a new, descriptively named branch to contain your change ( `git checkout -b my-awesome-feature` ).

4. Hack away, add tests. Not necessarily in that order.

5. Make sure everything still passes by running `script/cibuild` (see the [tests section](#running-tests-locally) below)

6. Push the branch up ( `git push origin my-awesome-feature` ).

7. Create a pull request by visiting `https://github.com/<your-username>/jekyll` and following the instructions at the top of the screen.

## Proposing updates to the documentation

We want the Jekyll documentation to be the best it can be. We've open-sourced our docs and we welcome any pull requests if you find it lacking.

### How to submit changes

You can find the documentation for jekyllrb.com in the [docs](https://github.com/jekyll/jekyll/tree/master/docs) directory. See the section above, [submitting a pull request](#submitting-a-pull-request) for information on how to propose a change.

One gotcha, all pull requests should be directed at the `master` branch (the default branch).

### Updating FontAwesome iconset for jekyllrb.com

We use a custom version of FontAwesome which contains just the icons we use.

If you ever need to update our documentation with an icon that is not already available in our custom iconset, you'll have to regenerate the iconset using Icomoon's Generator:

1. Go to <https://icomoon.io/app/>.

2. Click `Import Icons` on the top-horizontal-bar and upload the existing `<jekyll>/docs/icomoon-selection.json`.

3. Click `Add Icons from Library..` further down on the page, and add 'Font Awesome'.

4. Select the required icon(s) from the Library (make sure its the 'FontAwesome' library instead of 'IcoMoon-Free' library).

5. Click `Generate Font` on the bottom-horizontal-bar.

6. Inspect the included icons and proceed by clicking `Download`.

7. Extract the font files and adapt the CSS to the paths we use in Jekyll:

- Copy the entire `fonts` directory over and overwrite existing ones at `<jekyll>/docs/`.

- Copy the contents of `selection.json` and overwrite existing content inside `<jekyll>/docs/icomoon-selection.json`.

- Copy the entire `@font-face {}` declaration and only the \*\*new-icon(s)' css declarations\*\* further below, to update the

`<jekyll>/docs/\_sass/\_font-awesome.scss` sass partial.

- Fix paths in the `@font-face {}` declaration by adding `../` before `fonts/FontAwesome.\*` like so:

`('../fonts/Fontawesome.woff?9h6hxj')`.

### Adding plugins

If you want to add your plugin to the [list of plugins](https://jekyllrb.com/docs/plugins/#available-plugins), please submit a pull request modifying the [plugins page source file](https://github.com/jekyll/jekyll/blob/master/docs/\_docs/plugins.md) by adding a link to your plugin under the proper subheading depending upon its type.

## Code Contributions

Interested in submitting a pull request? Awesome. Read on. There's a few common gotchas that we'd love to help you avoid.

### Tests and documentation

Any time you propose a code change, you should also include updates to the documentation and tests within the same pull request.

#### Documentation

If your contribution changes any Jekyll behavior, make sure to update the documentation. Documentation lives in the `docs/\_docs` folder (spoiler alert: it's a Jekyll site!). If the docs are missing information, please feel free to add it in. Great docs make a great project. Include changes to the documentation within your pull request, and once merged, `jekyllrb.com` will be updated.

#### Tests

\* If you're creating a small fix or patch to an existing feature, a simple test is more than enough. You can usually copy/paste from an existing example in the `tests` folder, but if you need you can find out about our tests suites [Shoulda](https://github.com/thoughtbot/shoulda/tree/master) and [RSpec-Mocks](https://github.com/rspec/rspec-mocks).

\* If it's a brand new feature, create a new [Cucumber](https://github.com/cucumber/cucumber/) feature, reusing existing steps where appropriate.

### Code contributions generally

\* Jekyll uses the [Rubocop](https://github.com/bbatsov/rubocop) static analyzer to ensure that contributions follow the [GitHub Ruby Styleguide](https://github.com/styleguide/ruby). Please check your code using `script/fmt` and resolve any errors before pushing your branch.

\* Don't bump the Gem version in your pull request (if you don't know what that means, you probably didn't).

\* You can use the command `script/console` to start a REPL to explore the result of

Jekyll's methods. It also provides you with helpful methods to quickly create a

site or configuration. [Feel free to check it out!](https://github.com/jekyll/jekyll/blob/master/script/console)

\* Previously, we've used the WIP Probot app to help contributors determine whether their pull request is ready for review. Please use a [draft pull request](https://help.github.com/en/articles/about-pull-requests#draft-pull-requests) instead. When you're ready, [mark the pull request as ready for review](https://help.github.com/en/articles/changing-the-stage-of-a-pull-request)

## Running tests locally

### Test Dependencies

To run the test suite and build the gem you'll need to install Jekyll's dependencies by running the following command:

```sh

script/bootstrap

```

Before you make any changes, run the tests and make sure that they pass (to confirm your environment is configured properly):

```sh

script/cibuild

```

If you are only updating a file in `test/`, you can use the command:

```sh

script/test test/blah\_test.rb

```

If you are only updating a `.feature` file, you can use the command:

```sh

script/cucumber features/blah.feature

```

Both `script/test` and `script/cucumber` can be run without arguments to

run its entire respective suite.

## A thank you

Thanks! Hacking on Jekyll should be fun. If you find any of this hard to figure out, let us know so we can improve our process or documentation!