# Contributing to Docusaurus

[Docusaurus](https://docusaurus.io) is our way to hopefully help creating open source documentation easier. We currently have [multiple open source projects using it](https://docusaurus.io/en/users.html), with many more planned. If you're interested in contributing to Docusaurus, hopefully this document makes the process for contributing clear.

The [Open Source Guides](https://opensource.guide/) website has a collection of resources for individuals, communities, and companies who want to learn how to run and contribute to an open source project. Contributors and people new to open source alike will find the following guides especially useful:

- [How to Contribute to Open Source](https://opensource.guide/how-to-contribute/)

- [Building Welcoming Communities](https://opensource.guide/building-community/)

## [Code of Conduct](https://code.fb.com/codeofconduct)

Facebook has adopted a Code of Conduct that we expect project participants to adhere to. Please read [the full text](https://code.fb.com/codeofconduct) so that you can understand what actions will and will not be tolerated.

## Get Involved

There are many ways to contribute to Docusaurus, and many of them do not involve writing any code. Here's a few ideas to get started:

- Simply start using Docusaurus. Go through the [Getting Started](https://docusaurus.io/docs/en/installation.html) guide. Does everything work as expected? If not, we're always looking for improvements. Let us know by [opening an issue](#reporting-new-issues).

- Look through the [open issues](https://github.com/facebook/docusaurus/issues). Provide workarounds, ask for clarification, or suggest labels. Help [triage issues](#triaging-issues-and-pull-requests).

- If you find an issue you would like to fix, [open a pull request](#your-first-pull-request). Issues tagged as [\_Good first issue\_](https://github.com/facebook/docusaurus/labels/Good%20first%20issue) are a good place to get started.

- Read through the [Docusaurus docs](https://docusaurus.io/docs/en/installation). If you find anything that is confusing or can be improved, you can make edits by clicking "Edit" at the top of most docs.

- Take a look at the [features requested](https://github.com/facebook/docusaurus/labels/enhancement) by others in the community and consider opening a pull request if you see something you want to work on.

Contributions are very welcome. If you think you need help planning your contribution, please ping us on Twitter at [@docusaurus](https://twitter.com/docusaurus) and let us know you are looking for a bit of help.

### Versioned Docs

If you only want to make content changes you just need to know about versioned docs.

- `/docs` - The files in here are responsible for the "next" version at https://docusaurus.io/docs/en/next/installation.

- `website-1.x/versioned\_docs/version-X.Y.Z` - These are the docs for the X.Y.Z version at https://docusaurus.io/docs/en/X.Y.Z/installation.

To make a fix to the published versions you must edit the corresponding markdown file in both folders. If you only made changes in `docs`, be sure to be viewing the `next` version to see the updates (ensure there's `next` in the URL).

> Do not edit the auto-generated files within `versioned\_docs/` or `versioned\_sidebars/` unless you are sure it is necessary. For example, information about new features should not be documented in versioned docs. Edits made to older versions will not be propagated to newer versions of the docs.

### Join our Discord Channel

We have `#docusaurus-dev` on [Discord](https://discord.gg/docusaurus) to discuss all things Docusaurus development.

To participate in Docusaurus 2 dev, we have the [`#docusaurus-2-dev`](https://discord.gg/n8nQEAS) channel.

### Triaging Issues and Pull Requests

One great way you can contribute to the project without writing any code is to help triage issues and pull requests as they come in.

- Ask for more information if you believe the issue does not provide all the details required to solve it.

- Suggest [labels](https://github.com/facebook/docusaurus/labels) that can help categorize issues.

- Flag issues that are stale or that should be closed.

- Ask for test plans and review code.

## Our Development Process

Docusaurus uses [GitHub](https://github.com/facebook/docusaurus) as its source of truth. The core team will be working directly there. All changes will be public from the beginning.

When a change made on GitHub is approved, it will be checked by our continuous integration system, CircleCI.

### Branch Organization

Docusaurus has two primary branches: `master` and `gh-pages`.

`master` is where our code lives and development takes place. We will do our best to keep `master` in good shape, with tests passing at all times. We will also do our best not to publish updated `npm` packages that will break sites. But in order to move fast, we may make changes that could break existing sites. We will do our best to communicate these changes and version appropriately so you can lock into a specific Docusaurus version if need be.

`gh-pages` contains the [Docusaurus documentation](https://docusaurus.io). This branch is pushed to by CI and not generally managed manually.

## Bugs

We use [GitHub Issues](https://github.com/facebook/docusaurus/issues) for our public bugs. If you would like to report a problem, take a look around and see if someone already opened an issue about it. If you a are certain this is a new, unreported bug, you can submit a [bug report](#reporting-new-issues).

If you have questions about using Docusaurus, contact the Docusaurus Twitter account at [@docusaurus](https://twitter.com/docusaurus), and we will do our best to answer your questions.

You can also file issues as [feature requests or enhancements](https://github.com/facebook/docusaurus/labels/feature%20request). If you see anything you'd like to be implemented, create an issue with [feature template](https://raw.githubusercontent.com/facebook/docusaurus/master/.github/ISSUE\_TEMPLATE/feature.md)

## Reporting New Issues

When [opening a new issue](https://github.com/facebook/docusaurus/issues/new/choose), always make sure to fill out the issue template. \*\*This step is very important!\*\* Not doing so may result in your issue not managed in a timely fashion. Don't take this personally if this happens, and feel free to open a new issue once you've gathered all the information required by the template.

- \*\*One issue, one bug:\*\* Please report a single bug per issue.

- \*\*Provide reproduction steps:\*\* List all the steps necessary to reproduce the issue. The person reading your bug report should be able to follow these steps to reproduce your issue with minimal effort.

### Security Bugs

Facebook has a [bounty program](https://www.facebook.com/whitehat/) for the safe disclosure of security bugs. With that in mind, please do not file public issues; go through the process outlined on that page.

## Installation

1. Ensure you have [Yarn](https://yarnpkg.com/) installed.

1. After cloning the repository, run `yarn install` in the root of the repository.

1. To start a development server:

- For Docusaurus 1 development, run `yarn start:v1`

- For Docusaurus 2 development, run `yarn start`

## Online one-click setup for contributing

You can use Gitpod (a free, online, VS Code-like IDE) for contributing. With a single click it will launch a workspace (for Docusaurus 2) and automatically:

- clone the docusaurus repo.

- install the dependencies.

- run `yarn run start`

So that you can start contributing straight away.

[![Open in Gitpod](https://gitpod.io/button/open-in-gitpod.svg)](https://gitpod.io/#https://github.com/facebook/docusaurus)

## Pull Requests

### Your First Pull Request

So you have decided to contribute code back to upstream by opening a pull request. You've invested a good chunk of time, and we appreciate it. We will do our best to work with you and get the PR looked at.

Working on your first Pull Request? You can learn how from this free video series:

[\*\*How to Contribute to an Open Source Project on GitHub\*\*](https://egghead.io/courses/how-to-contribute-to-an-open-source-project-on-github)

We have a list of [beginner friendly issues](https://github.com/facebook/docusaurus/labels/good%20first%20issue) to help you get your feet wet in the Docusaurus codebase and familiar with our contribution process. This is a great place to get started.

### Proposing a Change

If you would like to request a new feature or enhancement but are not yet thinking about opening a pull request, you can also file an issue with [feature template](https://github.com/facebook/docusaurus/issues/new?template=feature.md).

If you intend to change the public API (e.g., something in `siteConfig.js`), or make any non-trivial changes to the implementation, we recommend filing an issue with [proposal template](https://github.com/facebook/docusaurus/issues/new?template=proposal.md) and including `[Proposal]` in the title. This lets us reach an agreement on your proposal before you put significant effort into it. These types of issues should be rare.

If you're only fixing a bug, it's fine to submit a pull request right away but we still recommend to file an issue detailing what you're fixing. This is helpful in case we don't accept that specific fix but want to keep track of the issue.

### Sending a Pull Request

Small pull requests are much easier to review and more likely to get merged. Make sure the PR does only one thing, otherwise please split it. It is recommended to follow this [commit message style](#semantic-commit-messages).

Please make sure the following is done when submitting a pull request:

1. Fork [the repository](https://github.com/facebook/docusaurus) and create your branch from `master`.

1. Add the copyright notice to the top of any code new files you've added.

1. Describe your [\*\*test plan\*\*](#test-plan) in your pull request description. Make sure to [test your changes](https://github.com/facebook/docusaurus/blob/master/admin/testing-changes-on-Docusaurus-itself.md)!

1. Make sure your code lints (`yarn prettier && yarn lint`).

1. Make sure your Jest tests pass (`yarn test`).

1. If you haven't already, [sign the CLA](https://code.facebook.com/cla).

All pull requests should be opened against the `master` branch.

#### Test Plan

A good test plan has the exact commands you ran and their output, provides screenshots or videos if the pull request changes UI.

- If you've changed APIs, update the documentation.

If you need help testing your changes locally, you can check out the doc on doing [local third party testing](./admin/local-third-party-project-testing.md).

#### Breaking Changes

When adding a new breaking change, follow this template in your pull request:

```md

### New breaking change here

- \*\*Who does this affect\*\*:

- \*\*How to migrate\*\*:

- \*\*Why make this breaking change\*\*:

- \*\*Severity (number of people affected x effort)\*\*:

```

#### Copyright Header for Source Code

Copy and paste this to the top of your new file(s):

```js

/\*\*

\* Copyright (c) Facebook, Inc. and its affiliates.

\*

\* This source code is licensed under the MIT license found in the

\* LICENSE file in the root directory of this source tree.

\*/

```

#### Contributor License Agreement (CLA)

In order to accept your pull request, we need you to submit a CLA. You only need to do this once, so if you've done this for another Facebook open source project, you're good to go. If you are submitting a pull request for the first time, the Facebook GitHub Bot will reply with a link to the CLA form. You may also [complete your CLA here](https://code.facebook.com/cla).

### What Happens Next?

The core Docusaurus team will be monitoring for pull requests. Do help us by keeping pull requests consistent by following the guidelines above.

## Style Guide

[Prettier](https://prettier.io) will catch most styling issues that may exist in your code. You can check the status of your code styling by simply running `yarn prettier`.

However, there are still some styles that Prettier cannot pick up.

## Semantic Commit Messages

See how a minor change to your commit message style can make you a better programmer.

Format: `<type>(<scope>): <subject>`

`<scope>` is optional

## Example

```

feat: allow overriding of webpack config

^--^ ^------------^

| |

| +-> Summary in present tense.

|

+-------> Type: chore, docs, feat, fix, refactor, style, or test.

```

The various types of commits:

- `feat`: (new feature for the user, not a new feature for build script)

- `fix`: (bug fix for the user, not a fix to a build script)

- `docs`: (changes to the documentation)

- `style`: (formatting, missing semi colons, etc; no production code change)

- `refactor`: (refactoring production code, eg. renaming a variable)

- `test`: (adding missing tests, refactoring tests; no production code change)

- `chore`: (updating grunt tasks etc; no production code change)

Use lower case not title case!

### Code Conventions

#### General

- \*\*Most important: Look around.\*\* Match the style you see used in the rest of the project. This includes formatting, naming files, naming things in code, naming things in documentation.

- "Attractive"

### Documentation

- Do not wrap lines at 80 characters - configure your editor to soft-wrap when editing documentation.

## License

By contributing to Docusaurus, you agree that your contributions will be licensed under its MIT license.