# Contributing

So you want to help? That's great!

![Image of happy people jumping with excitement](https://media.giphy.com/media/BlVnrxJgTGsUw/giphy.gif)

Here are a few things to know to get you started on the right path.

## Committing code

We make all changes via pull requests. As we have many pull requests from developers new to mermaid, the current approach is to have \*knsv, Knut Sveidqvist\* as a main reviewer of changes and merging pull requests. More precisely like this:

\* Large changes reviewed by knsv or other developer asked to review by knsv

\* Smaller low-risk changes like dependecies, documentation etc can be merged by active collaborators

\* documentation (updates to the docs folder is also allowed via direct commits)

To commit code, create a branch, let it start with the type like feature or bug followed by the issue number for reference and some describing text.

One example:

`feature/945\_state\_diagrams`

Another:

`bug/123\_nasty\_bug\_branch`

## Committing documentation

Less strict here, it is ok to commit directly in the develop branch if you are a collaborator.

## Branching

Going forward we will use a git flow inspired approach to branching. So development is done in develop, to do the development in the develop.

Once development is done we branch a release branch from develop for testing.

Once the release happens we merge the release branch to master and kill the release branch.

This means... \*\*branch off your pull request from develop\*\*

## Content of a pull request

A new feature has been born. Great! But without the steps below it might just ... fade away ...

### \*\*Add unit tests for the parsing part\*\*

This is important so that, if someone else does a change to the grammar that does not know about this great feature, gets notified early on when that change breaks the parser. Another important aspect is that without proper parsing tests refactoring is pretty much impossible.

### \*\*Add e2e tests\*\*

This tests the rendering and visual apearance of the diagram. This ensures that the rendering of that feature in the e2e will be reviewed in the release process going forward. Less chance that it breaks!

To start working with the e2e tests, run `yarn dev` to start the dev server, after that start cypress by running `cypress open` in the mermaid folder. (Make sure you have path to cypress in order, the binary is located in node\_modules/.bin).

The rendering tests are very straightforward to create. There is a function imgSnapshotTest. This function takes a diagram in text form, the mermaid options and renders that diagram in cypress.

When running in ci it will take a snapshot of the rendered diagram and compare it with the snapshot from last build and flag for review it if it differs.

This is what a rendering test looks like:

```

it('should render forks and joins', () => {

imgSnapshotTest(

`

stateDiagram

state fork\_state &lt;&lt;fork&gt;&gt;

[\*] --> fork\_state

fork\_state --> State2

fork\_state --> State3

state join\_state &lt;&lt;join&gt;&gt;

State2 --> join\_state

State3 --> join\_state

join\_state --> State4

State4 --> [\*]

`,

{ logLevel: 0 }

);

cy.get('svg');

});

```

### \*\*Add documentation for it\*\*

Finally, if it is not in the documentation, no one will know about it and then \*\*no one will use it\*\*. Wouldn't that be sad? With all the effort that was put into the feature?

The docs are located in the docs folder and are ofc written in markdown. Just pick the right section and start typing. If you want to add to the structure as in adding a new section and new file you do that via the \_navbar.md.

The changes in master is reflected in http://mermaid-js.github.io/mermaid/ once released the updates are commited to https://mermaid-js.github.io/#/

## Last words

Don't get daunted if it is hard in the beginning. We have a great community with only encouraging words. So if you get stuck, ask for help and hints in the slack forum. If you want to show off something good, show it off there.

[Join our slack community if you want closer contact!](https://join.slack.com/t/mermaid-talk/shared\_invite/enQtNzc4NDIyNzk4OTAyLWVhYjQxOTI2OTg4YmE1ZmJkY2Y4MTU3ODliYmIwOTY3NDJlYjA0YjIyZTdkMDMyZTUwOGI0NjEzYmEwODcwOTE)

![Image of superhero wishing you good luck](https://media.giphy.com/media/l49JHz7kJvl6MCj3G/giphy.gif)