# Welcome to Clasp

Clasp is a very fast-moving project, but that speed still currently relies mostly on one person: drmeister. Clasp still has a very long way to go too, there's a plethora of [issues and missing features](https://github.com/drmeister/clasp/issues). As such there should be plenty of room for other people to join in and help us out to make Clasp the amazing project it has the potential to be.

If you would like to get acquainted with the current developers and the status of Clasp in general, you are more than welcome to check by our IRC channel #clasp on Freenode. It is also a good place to be if you just want to lurk around and watch the development happen before your very eyes.

This document is for those who would like to become an actual contributor to Clasp, meaning adding or cleaning up code, or helping out with organisation and testing.

# Submission Constraints

While we welcome submissions very much, we nevertheless need to enforce some rules and guidelines in order to keep things from looking like a plate of cold spaghetti. As Clasp does not have a big developer team at the moment, these rules are not too strict or roundabout. Submissions that do not follow them will be rejected however.

## Setting up Your Work Area

You will want to [fork](https://github.com/drmeister/clasp/fork) the Clasp repository, clone that to your machine and set up the upstream remote using `git remote add upstream https://github.com/drmeister/clasp`. While developing you should periodically synchronise your fork with the upstream as otherwise things can get too out of sync to still be mergeable. In case of updates from upstream, you should pull those in using `git pull --rebase upstream dev` while on your local `dev` or feature branch. \*\*Only create merge commits if there is a conflict\*\*; we'd like to keep a somewhat clean history.

## Code Style

Please follow the [LLVM style standard](http://llvm.org/docs/CodingStandards.html), in the very least concerning code formatting, indentation, and brace-style. There is a `.dir-locals.el` file at the Clasp root that should automatically configure the proper rules, if you happen to use Emacs. Otherwise you can run `clang-format` on your code with the settings from the `.clang-format` file, which you can find at the Clasp root as well.

## Branching and Committing

You should never, \*ever\* commit to `testing` or `preview`. These branches are solely for build testing and should not receive any commits. If you do commit to them by mistake, switch to the `dev` branch, cherry-pick the commit, and then reset the `testing`/`preview` branch to the commit before that.

You should also never commit to `master` unless you have the permission to do so. `master` is guarded for releases and only hotfixes or documentation changes should be pushed on there between releases. Not to mention that it is very likely for `master` to be heavily out of date, so developing against it is a bad idea anyway.

If you are only fixing minor things or adding small features, it is Ok to commit to `dev`. Otherwise, if you are planning something bigger like a refactoring of some files, changing the architecture, adding a component to Clasp, or something like that, then you must create a feature branch. The branch's name should be prefixed with `dev-` followed by a somewhat descriptive name that tells us what you're doing in that branch.

Please use descriptive commit messages that tell us what you did. Please also avoid committing huge amounts of changes in one go-- create multiple smaller commits where possible and applicable. This makes bisecting in case of regressions much easier. If your changes are related to an issue ticket, please reference it in your commit message.

## Pull Requests

Once you are done with your changes, make sure that all upstream changes are present on your branch and verify that you did not break anything by running a full build against a clean clone of your fork. If everything passed and Clasp launches successfully, you can [submit a pull request](https://github.com/drmeister/clasp/pull/new/dev). You should usually PR against `dev` or the corresponding feature branch if one exists. Please choose a fitting title and write a short description that summarises your changes and notes that you successfully build-tested everything to work.

When your PR has been approved and merged in, things will be pushed out to `testing`, where our CI machines will hopefully verify that everything is still just fine and dandy. If that comes to pass, you are free to celebrate and rejoice!

# Reaching Clasp Developers

In case of problems, questions, or general discussion about Clasp, the following three would be the places to go in descending order of preference.

\* [#clasp on irc.freenode.net](irc://irc.freenode.net/%23clasp) / [Archive](https://irclog.tymoon.eu/freenode/clasp)

\* [Clasp mailing list](https://mailman.common-lisp.net/listinfo/clasp-devel)

\* [Issue tracker](https://github.com/drmeister/clasp/issues)