# Contributing to Sucrase

## Intro

Thanks for taking a look at the project and thinking about contributing!

There are two main ways to communicate about Sucrase development:

\* Join the [Gitter room](https://gitter.im/sucrasejs/Lobby) and feel free to ask

questions or discuss anything.

\* File an issue or PR in this repository. Issues don't need to be bug reports;

usage questions or suggests are fine.

All contributions and interactions are expected to follow the

[Code of Conduct](./CODE\_OF\_CONDUCT.md).

## What kinds of contributions make sense?

There are lots of ways to contribute to Sucrase! The easiest way to get started

is to just starting using Sucrase for a project and see what issues you run

into. Did you run into any bugs? Were the details confusing? Is there a way to

make it easier to use? All of those would be great issues to file in this

repository!

In addition to filing issues, there are lots of ways to contribute:

\* Fix bugs. Be sure to also include a regression test for any bug fix!

\* Add more tests or test infrastructure. Right now, Sucrase runs on a few open

source projects as part of its test suite, but it would always be great to add

more.

\* Add better performance tests.

\* Add better documentation, like a Getting Started guide.

\* Write more integrations for Sucrase, like a Browserify plugin.

Contributing features, refactors, and performance improvements to the "core" of

Sucrase (the parser and the transformers) may need some more thought and care.

See the [Project Vision](./docs/PROJECT\_VISION.md) for some thoughts on what

types of improvements make sense for the project, and make sure to start a

discussion before getting too deep in the code. Some problems may be best solved

outside of the core Sucrase code, or may be better as a fork of Sucrase.

See [this issue](https://github.com/alangpierce/sucrase/issues/161) for some

more concrete suggestions.

## Getting started

Sucrase uses `yarn` for everything.

```

git clone git@github.com:alangpierce/sucrase.git

cd sucrase

yarn # Install dependencies.

yarn test # Run tests.

yarn build # Build Sucrase.

yarn fast-build # Quickly build Sucrase for development.

yarn release 1.2.3 # Run the release script, releasing version 1.2.3.

```

I'd also recommend getting your editor set up nicely to handle TypeScript,

Prettier, and Mocha.

Contributions should follow the normal GitHub pull request process: fork the

repo, make a new branch with your change, push the branch, and create a pull

request with a meaningful title and description, including any interesting

details of the change and references to any issues closed by the change.

## Code structure

The code is organized as a monorepo (with one primary `sucrase` package and

several smaller packages/projects). Here are the most important directories:

\* The `src` directory is where the main code lives. Much of the code is in the

`parser` subdirectory, which is a heavily modified fork of Babel's parser. The

rest of the code handles transforming the code given the result of the parser.

\* The `test` directory has all tests, generally broken down by which transform

is most relevant. Nearly all tests simply give code as input and assert the

output code. Running a test in debug mode and stepping through different parts

of the code is a great way to learn how Sucrase works!

\* The `website` directory contains the full code for the

[sucrase.io website](https://sucrase.io).

\* The `integrations` directory has several small packages integrating Sucrase

with Webpack, Jest, etc.

\* The `example-runner` directory has code to clone various open source projects,

run Sucrase on them, then run the tests to make sure that the Sucrase output

was correct. You can run all example projects with `yarn run-examples`.

\* The `generator` directory contains some some utilities for generating code

used in Sucrase, and can be run using `yarn generate`.

\* The `benchmark` directory contains some some simple benchmarks, which can be

run with `yarn benchmark`.

\* The `script` directory has all build scripts (written in TypeScript and

compiled with Sucrase).