# Contributing to annyang

Thank you for taking the time to get involved with annyang! :+1:

There are several ways you can help the project out:

\* [Contributing code](#contributing-code)

\* [Reporting Bugs](#reporting-bugs)

\* [Feature Requests and Ideas](#feature-requests-and-ideas)

## How Can I Contribute?

### Contributing Code

A lot of annyang's functionality came from pull requests sent over GitHub. Here is how you can contribute too:

- [x] Fork the repository from the [annyang GitHub page](https://github.com/TalAter/annyang).

- [x] Clone a copy to your local machine with `$ git clone git@github.com:YOUR-GITHUB-USER-NAME/annyang.git`

- [x] Make sure you have \*node.js\* and \*npm\* installed on your machine. You can use this [guide](https://docs.npmjs.com/getting-started/installing-node) for help.

- [x] Install all of annyang's development dependencies with npm. `$ cd annyang; npm install`

- [x] If you do not have the grunt-cli installed globally, run `$ npm install -g grunt-cli`

- [x] Run grunt to make sure everything runs smoothly `$ grunt`

- [x] Add tests for your code. [See details below](#automated-testing).

- [x] Code, code, code. Changes should be done in `/src/annyang.js`. They will be transpiled to `/dist/annyang.js` and `/dist/annyang.min.js`.

- [x] Run `$ grunt` after making changes to verify that everything still works and the tests all pass.

:bulb: A great alternative to repeatedly running `$ grunt` is to run `$ grunt watch` once, and leave this process running. It will continuously run all the tests and build the files every time you make a change to one of annyang's files. It will even \*beep\* if you make an error, and help you debug it. :+1:

- [x] You can run a local server to test your work in the browser by running `$ grunt dev`. This will also automatically run `$ grunt watch` for you.

Point your browser to `https://localhost:8443/demo/` to see the demo page.

Since it's using a self-signed certificate, you might need to click \*"Proceed Anyway"\*.

- [x] Before committing your changes, the last step must always be running `$ grunt`. This makes sure everything works, and all files are kept up to date with your changes.

- [x] Once you've made sure all your changes work correctly and have been committed, push your local changes back to github with `$ git push -u origin master`

- [x] Visit your fork on GitHub.com ([https://github.com/YOUR-USER-NAME/annyang](https://github.com/YOUR-USER-NAME/annyang)) and create a pull request for your changes.

- [x] Makes sure your pull request describes exactly what you changed and if it relates to an open issue references that issue (just include the issue number in the title like this: #49)

#### Important:

\* Make sure to run `npm install` and `grunt` and make sure all tasks completed successfully before committing.

\* Do not change the [API docs](https://github.com/TalAter/annyang/blob/master/docs/README.md) in `/docs/README.md` directly. This file is generated automatically, and your changes will be overwritten. Instead, update the relevant comments in `src/annyang.js`

\* annyang is still not completely automatically tested :disappointed:. If you make a change, please make sure to test your change thoroughly to make sure no backward functionality was broken, and that your change works as intended.

\* Do not update the version number yourself.

\* Please stick to the project's existing coding style. Coding styles don't need to have a consensus, they just need to be consistent :smile:.

\* Push your changes to a topic branch in your fork of the repository. Your branch should be based on the `master` branch.

\* When submitting [pull request](https://help.github.com/articles/using-pull-requests/), please elaborate as much as possible about the change, your motivation for the change, etc.

#### Automated Testing

annyang is tested using [Jasmine](http://jasmine.github.io/2.0/introduction.html).

Please include tests for any changes you make:

\* If you found a bug, please write a test that fails because of that bug, then fix the bug so that the test passes.

\* If you are adding a new feature, write tests that thoroughly test every possible use of your code.

\* If you are changing existing functionality, make sure to update existing tests so they pass. (This is a last resort move. Whenever possible try to maintain backward compatibility)

The tests reside in \*BasicSpec.js\*. The file contains a series of spec groups (e.g. `describe('a spec group', function() {});`) which each contain 1 or more specs (e.g. `it('should do stuff', function() {});`). Some of the spec groups also contain some code which runs before each spec (`beforeEach(function() {});`).

To simulate Speech Recognition in the testing environment, annyang uses a mock object called [Corti](https://github.com/TalAter/Corti) which mocks the browser's SpeechRecognition object. Corti also adds a number of utility functions to the SpeechRecognition object which simulate user actions (e.g. `say('Hello there')`), and allow checking the SpeechRecognition's status (e.g. `isListening() === true`).

### Reporting Bugs

Bugs are tracked as [GitHub issues](https://github.com/TalAter/annyang/issues). If you found a bug with annyang, the quickest way to get help would be to look through existing open and closed [GitHub issues](https://github.com/TalAter/annyang/issues?q=is%3Aissue). If the issue is already being discussed and hasn't been resolved yet, you can join the discussion and provide details about the problem you are having. If this is a new bug, please open a [new issue](https://github.com/TalAter/annyang/issues/new).

When you are creating a bug report, please include as many details as possible.

Explain the problem and include additional details to help maintainers reproduce the problem.

\* Use a clear and descriptive title for the issue to identify the problem.

\* Describe the exact steps which reproduce the problem. Share the relevant code to reproduce the issue if possible.

\* Try to isolate the issue as much as possible, reducing unrelated code until you get to the minimal amount of code in which the bug still reproduces. This is the most important step to help the community solve the issue.

### Feature Requests and Ideas

We track discussions of new features, proposed changes, and other ideas as [GitHub issues](https://github.com/TalAter/annyang/issues). If you would like to discuss one of those, please first look through existing open and closed [GitHub issues](https://github.com/TalAter/annyang/issues?q=is%3Aissue) and see if there is already a discussion on this topic which you can join. If there isn't, please open a [new issue](https://github.com/TalAter/annyang/issues/new).

When discussing new ideas or proposing changes, please take the time to be as descriptive as possible about the topic at hand. Please take the time to explain the issue you are facing, or the problem you propose to solve in as much detail as possible.