Guidelines on how this open source community works :)

Read below to see how you can join an existing project or create your own.

## One rule of this community:

We don't care if you break things. This is a playground and we encourage failing often. Use this as a practice ground and enjoy contributing to projects you create with your fellow students. Many students have gained real world experience "working in teams" by working on these projects.

## A Guide to Get Started

1. Read the wonderful [\_gitStarted Guide\_](https://github.com/zero-to-mastery/start-here-guidelines/blob/855a00243db60c71905f6e3afd95ebf2cf7459a0/gitstartedguideoptimized.pdf) by our fellow student [@wanraitelli](https://github.com/wanraitelli).

Check out [Andrei's videos on github](https://www.udemy.com/the-complete-web-developer-in-2018/learn/v4/t/lecture/8725782/) or this free how-to tutorial at http://makeapullrequest.com/.

2. On the [GitHub page for this repository](https://github.com/zero-to-mastery/start-here-guidelines), click on the Button "Fork".

![fork image](https://help.github.com/assets/images/help/repository/fork\_button.jpg)

3. Clone \_your forked repository\_ to your computer:

![code ui](https://docs.github.com/assets/images/help/repository/code-button.png)

For example, run this command inside your terminal:

```bash

git clone https://github.com/<your-github-username>/start-here-guidelines.git

```

\*\*Replace \<your-github-username\>!\*\*

Learn more about [forking](https://help.github.com/en/github/getting-started-with-github/fork-a-repo) and [cloning a repo](https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository).

4. Before you make any changes, [keep your fork in sync](https://www.freecodecamp.org/news/how-to-sync-your-fork-with-the-original-git-repository/) to avoid merge conflicts:

```bash

git remote add upstream https://github.com/zero-to-mastery/start-here-guidelines.git

git pull upstream master

```

If you run into a \*\*merge conflict\*\*, you have to resolve the conflict. There are a lot of guides online, or you can try this one by [opensource.com](https://opensource.com/article/20/4/git-merge-conflict).

5. On your computer, open your text editor, and add your name to the `CONTRIBUTORS.md` file.

6. Add the changes with `git add`, `git commit` ([write a good commit message](https://chris.beams.io/posts/git-commit/), if possible):

```bash

git add CONTRIBUTORS.md

git commit -m "Add <your-github-username>"

```

\*\*Replace \<your-github-username\>!\*\*

7. Push your changes \_to your repository\_:

```bash

git push origin master

```

8. Go to the GitHub page of \_your fork\_, and make a pull request:

![pull request image](https://help.github.com/assets/images/help/pull\_requests/choose-base-and-compare-branches.png)

Read more about pull requests on the [GitHub help pages](https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/creating-a-pull-request).

9. Wait until Zerobot or or one of the maintainers merges your pull request. If there are any conflicts, you will get a notification.

10. Go join a project and start contributing or create your own group apps. Don't be shy and enjoy creating things together (We have over 20 projects for all level of programmers)! [Check out this guide](https://github.com/zero-to-mastery/start-here-guidelines/blob/master/Get\_Started.md) for more information on selecting a project.

11. To see the Zero to Mastery Icon in your GitHub profile, [follow these steps](https://help.github.com/articles/publicizing-or-hiding-organization-membership/) (you must complete step 1 and 2 for this to work).

## Anatomy of an open source project:

Every open source community is different.

Spending years on one open source project means you’ve gotten to know \_one\_ open source project. Move to a different project, and you might find the vocabulary, norms, and communication styles are completely different.

That said, many open source projects follow a similar organizational structure. Understanding the different community roles and overall process will help you get quickly oriented to any new project.

A typical open source project has the following types of people:

\*\*Author\*\*: The person/s or organization that created the project.

\*\*Owner\*\*: The person/s who has administrative ownership over the organization or repository (not always the same as the original author).

\*\*Maintainers\*\*: Contributors who are responsible for driving the vision and managing the organizational aspects of the project (may also be authors or owners of the project).

\*\*Contributors\*\*: Everyone who has contributed something back to the project.

\*\*Community Members\*\*: People who use the project. They might be active in conversations or express their opinion on the project’s direction.

Bigger projects may also have subcommittees or working groups focused on different tasks, such as tooling, triage, community moderation, and event organizing. Look on a project’s website for a “team” page, or in the repository for governance documentation, to find this information.

A project also has documentation. These files are usually listed in the top level of a repository.

\*\*LICENSE\*\*: By definition, every open source project must have an open source license. If the project does not have a license, it is not open source.

\*\*README\*\*: The README is the instruction manual that welcomes new community members to the project. It explains why the project is useful and how to get started.

\*\*CONTRIBUTING\*\*: Whereas READMEs help people use the project, contributing docs help people contribute to the project. It explains what types of contributions are needed and how the process works. While not every project has a CONTRIBUTING file, its presence signals that this is a welcoming project to contribute to.

\*\*CODE\_OF\_CONDUCT\*\*: The code of conduct sets ground rules for participants’ behavior associated and helps to facilitate a friendly, welcoming environment. While not every project has a CODE\_OF\_CONDUCT file, its presence signals that this is a welcoming project to contribute to.

\*\*Other documentation\*\*: There might be additional documentation, such as tutorials, walkthroughs, or governance policies, especially on bigger projects.

Finally, open source projects use the following tools to organize discussion. Reading through the archives will give you a good picture of how the community thinks and works.

\*\*Issue tracker\*\*: Where people discuss issues related to the project.

\*\*Pull requests\*\*: Where people discuss and review changes that are in progress.

\*\*Discussion forums or mailing lists\*\*: Some projects may use these channels for conversational topics (for example, “How do I…“ or “What do you think about…“ instead of bug reports or feature requests). Others use the issue tracker for all conversations.

\*\*Synchronous chat channel\*\*: Some projects use chat channels (such as Discord or IRC) for casual conversation, collaboration, and quick exchanges.

\*\*Get all the ZTM Courses, for one monthly subscription\*\*: [Here](https://zerotomastery.io/courses?utm\_source=github&utm\_medium=start-here-guidelines).