



# Make a README

Because no one can read your mind ([yet](#))



Need an interactive developer hub, complete with documentation, reference guides, tutorials, realtime API logs and community? [Try ReadMe.com!](#)

Make a README is proudly sponsored by ReadMe.com

---

## README 101

### What is it?

A [README](#) is a text file that introduces and explains a project. It contains information that is commonly required to understand what the project is about.

### Why should I make it?

It's an easy way to answer questions that your audience will likely have regarding how to install and use your project and also how to collaborate with you.

### Who should make it?

Anyone who is working on a programming project, especially if you want others to use it or contribute.

### When should I make it?

Definitely before you show a project to other people or make it public. You might want to get into the habit of making it the [first file you create](#) in a new project.

## Where should I put it?

In the top level directory of the project. This is where someone who is new to your project will start out. Code hosting services such as [GitHub](#), [Bitbucket](#), and [GitLab](#) will also look for your README and display it along with the list of files and directories in your project.

## How should I make it?

While READMEs can be written in any text file format, the most common one that is used nowadays is [Markdown](#). It allows you to add some lightweight formatting. You can learn more about it at the [CommonMark website](#), which also has a helpful [reference guide](#) and an [interactive tutorial](#).

Some other formats that you might see are [plain text](#), [reStructuredText](#) (common in [Python](#) projects), and [Textile](#).

You can use any text editor. There are plugins for many editors (e.g. [Atom](#), [Emacs](#), [Sublime Text](#), [Vim](#), and [Visual Studio Code](#)) that allow you to preview Markdown while you are editing it.

You can also use a dedicated Markdown editor like [Typora](#) or an online one like [StackEdit](#) or [Dillinger](#). You can even use the editable template below.

# Template

## Markdown Input (editable)

```
# Foobar
```

```
Foobar is a Python library for dealing  
with word pluralization.
```

```
## Installation
```

```
Use the package manager [pip]  
(https://pip.pypa.io/en/stable/) to  
install foobar.
```

```
```bash  
pip install foobar  
```
```

## Rendered

### Foobar

Foobar is a Python library for dealing with word pluralization.

### Installation

Use the package manager [pip](#) to install foobar.

```
pip install foobar
```

## ## Usage

```
```python
import foobar

# returns 'words'
foobar.pluralize('word')

# returns 'geese'
foobar.pluralize('goose')

# returns 'phenomenon'
foobar.singularize('phenomena')
```
```

## ## Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

Please make sure to update tests as appropriate.

## ## License

[MIT]  
(<https://choosealicense.com/licenses/mit/>)

## Usage

```
import foobar

# returns 'words'
foobar.pluralize('word')

# returns 'geese'
foobar.pluralize('goose')

# returns 'phenomenon'
foobar.singularize('phenomena')
```

## Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

Please make sure to update tests as appropriate.

## License

MIT

# Suggestions for a good README

Every project is different, so consider which of these sections apply to yours. The sections used in the template are suggestions for most open source projects. Also keep in mind that while a README can be too long and detailed, too long is better than too short. If you think your README is too long, consider utilizing [another form of documentation](#) rather than cutting out information.

## Name

Choose a self-explaining name for your project.

## Description

Let people know what your project can do specifically. Provide context and add a link to any reference visitors might be unfamiliar with. A list of **Features** or a **Background** subsection can also be added here. If there are

alternatives to your project, this is a good place to list differentiating factors.

## Badges

On some READMEs, you may see small images that convey metadata, such as whether or not all the tests are passing for the project. You can use [Shields](#) to add some to your README. Many services also have instructions for adding a badge.

## Visuals

Depending on what you are making, it can be a good idea to include screenshots or even a video (you'll frequently see GIFs rather than actual videos). Tools like [ttygif](#) can help, but check out [Asciinema](#) for a more sophisticated method.

## Installation

Within a particular ecosystem, there may be a common way of installing things, such as using [Yarn](#), [NuGet](#), or [Homebrew](#). However, consider the possibility that whoever is reading your README is a novice and would like more guidance. Listing specific steps helps remove ambiguity and gets people to using your project as quickly as possible. If it only runs in a specific context like a particular programming language version or operating system or has dependencies that have to be installed manually, also add a **Requirements** subsection.

## Usage

Use examples liberally, and show the expected output if you can. It's helpful to have inline the smallest example of usage that you can demonstrate, while providing links to more sophisticated examples if they are too long to reasonably include in the README.

## Support

Tell people where they can go to for help. It can be any combination of an issue tracker, a chat room, an email address, etc.

## Roadmap

If you have ideas for releases in the future, it is a good idea to list them in the README.

## Contributing

State if you are open to contributions and what your requirements are for accepting them.

For people who want to make changes to your project, it's helpful to have some documentation on how to get started. Perhaps there is a script that they should run or some environment variables that they need to set. Make these steps explicit. These instructions could also be useful to your future self.

You can also document commands to [lint the code](#) or [run tests](#). These steps help to ensure high code quality and reduce the likelihood that the changes inadvertently break something. Having instructions for running tests is especially helpful if it requires external setup, such as starting a [Selenium](#) server for testing in a browser.

## Authors and acknowledgment

Show your appreciation to those who have contributed to the project.

## License

For open source projects, say how it is [licensed](#).

## Project status

If you have run out of energy or time for your project, put a note at the top of the README saying that development has slowed down or stopped completely. Someone may choose to fork your project or volunteer to step in as a maintainer or owner, allowing your project to keep going. You can also make an explicit request for maintainers.

## FAQ

### Is there a standard README format?

Not all of the suggestions here will make sense for every project, so it's really up to the developers what information should be included in the README.

### What are some other thoughts on writing READMEs?

Check out [Awesome README](#) for a list of more resources.

### What should the README file be named?

`README.md` (or a different file extension if you choose to use a non-Markdown file format). It is [traditionally uppercase](#) so that it is more prominent, but it's not a big deal if you think it [looks better lowercase](#).

### What if I disagree with something, want to make a change, or have any other feedback?

Please don't hesitate to open an [issue](#) or [pull request](#). You can also send me a message on [Twitter](#).

### Mind reading?

[Scientists](#) and companies like [Facebook](#) and [Neuralink](#) (presumably) are working on it. Perhaps in the future, you'll be able to attach a copy of your thoughts and/or consciousness to your projects. In the meantime, please make READMEs.

## What's next?

### License

If your project is [open source](#), it's [important](#) to include a [license](#). You can use [this website](#) to help you pick one.

### Changelog

Make a README is inspired by [Keep a Changelog](#). A [changelog](#) is another file that is very useful for programming projects.

### More Documentation

A README is a crucial but basic way of documenting your project. While every project should at least have a README, more involved ones can also benefit from a [wiki](#) or a dedicated documentation website. [GitHub](#), [Bitbucket](#),

and [GitLab](#) all support maintaining a wiki alongside your project, and here are some tools and services that can help you generate a documentation website:

- [Daux.io](#)
- [Docusaurus](#)
- [GitBook](#)
- [MkDocs](#)
- [Read the Docs](#)
- [ReadMe](#)
- [Slate](#)
- [Docsify](#)

## Contributing

Just having a "Contributing" section in your README is a good start. Another approach is to split off your guidelines into their own file ( `CONTRIBUTING.md` ). If you use GitHub and have this file, then anyone who creates an issue or opens a pull request [will get a link](#) to it.

You can also create an [issue template](#) and a [pull request template](#). These files give your users and collaborators templates to fill in with the information that you'll need to properly respond. This helps to avoid situations like getting very vague issues. Both GitHub and [GitLab](#) will use the templates automatically.

Make a README is maintained by [Danny Guo](#), hosted on [GitHub](#) with a [MIT license](#), and powered by [Netlify](#).