# Contributing to PHPWord

PHPWord is built by the crowd and for the crowd. Every contribution is welcome; either by [reporting a bug](https://github.com/PHPOffice/PHPWord/issues/new?labels=Bug+Report&template=bug\_report.md) or [suggesting improvements](https://github.com/PHPOffice/PHPWord/issues/new?labels=Change+Request&template=feature\_request.md), or in a more active form like [requesting a pull](https://github.com/PHPOffice/PHPWord/pulls).

We want to create a high quality document writer and reader library that people can use with more confidence and fewer bugs. We want to collaborate happily, code joyfully, and live merrily. Thus, below are some guidelines that we expect to be followed by each contributor:

- \*\*Be brief, but be bold\*\*. State your issues briefly. But speak out your ideas loudly, even if you can't or don't know how to implement them right away. The world will be better with limitless innovations.

- \*\*Follow PHP-FIG standards\*\*. We follow PHP Standards Recommendations (PSRs) by [PHP Framework Interoperability Group](http://www.php-fig.org/). If you're not familiar with these standards, [familiarize yourself now](https://github.com/php-fig/fig-standards). Also, please run `composer fix` to automatically fix your code to match these recommendations.

- \*\*Test your code\*\*. No one knows your code better than you, so we depend on you to test the changes you make before pull request submission. We use [PHPUnit](https://phpunit.de/) for our testing purposes and request that you use this tool too. Tests can be ran with `composer test`. [Documentation for writing tests with PHPUnit is available on Read the Docs.](https://phpunit.readthedocs.io)

- \*\*Use best practices when submitting pull requests\*\*. Create a separate branch named specifically for the issue that you are addressing. Read the [GitHub manual](https://help.github.com/articles/about-pull-requests) to learn more about pull requests and GitHub. If you are new to GitHub, read [this short manual](https://help.github.com/articles/fork-a-repo) to get yourself familiar with forks and how git works in general. [This video](http://www.youtube.com/watch?v=-zvHQXnBO6c) explains how to synchronize your fork on GitHub with the upstream branch from PHPWord.

## Getting Started

1. [Clone](https://help.github.com/en/articles/cloning-a-repository) [PHPWord](https://github.com/PHPOffice/PHPWord/)

2. [Install Composer](https://getcomposer.org/download/) if you don't already have it

3. Open your terminal and:

1. Switch to the directory PHPWord was cloned to (e.g., `cd ~/Projects/PHPWord/`)

2. Run `composer install` to install the dependencies

You're ready to start working on PHPWord! Tests belong in the `/tests/PhpWord/` directory, the source code is in `/src/PhpWord/`, and any documentation should go in `/docs/`. Familiarize yourself with the codebase and try your hand at fixing [one of our outstanding issues](https://github.com/PHPOffice/PHPWord/issues). Before you get started, check the [existing pull requests](https://github.com/PHPOffice/PHPWord/pulls) to make sure no one else is already working on it.

Once you have an issue you want to start working on, you'll need to write tests for it, and then you can start implementing the changes necessary to pass the new tests. To run the tests, you can run one of the following commands in your terminal:

- `composer test-no-coverage` to run all of the tests

- `composer test` to run all of the tests and generate test coverage reports

When you're ready to submit your new (and fully tested) feature, ensure `composer check` passes and [submit a pull request to PHPWord](https://github.com/PHPOffice/PHPWord/issues/new).

That's it. Thank you for your interest in PHPWord, and welcome!

May the Force be with you.