## Reporting issues

\* Consider joining the [chat](https://wiki.postmarketos.org/wiki/Matrix\_and\_IRC) for instant help.

\* Maybe your question is answered in the [wiki](https://wiki.postmarketos.org/) somewhere. [Search](https://wiki.postmarketos.org/index.php?search=&title=Special%3ASearch&go=Go) first!

\* Otherwise, just ask what you want to know. We're happy if we can help you and glad that you're using `pmbootstrap`!

## Development

See pmbootstrap's [Development Guide](https://wiki.postmarketos.org/wiki/Development\_guide).

### Contributing code changes

\* [Fork](https://guides.github.com/activities/forking/) this repository, commit your changes and then make a [pull-request](https://guides.github.com/activities/forking/#making-a-pull-request) (PR).

\* Please test your code before submitting a PR.

\* We squash all commits from one PR into one commit. Please make multiple PRs if you feel like your changes should appear as multiple commits in the git log ([more information](https://wiki.postmarketos.org/wiki/FAQ#Development:\_Why\_do\_we\_use\_squash\_.27n.27\_merge\_for\_pull\_requests.3F)).

### Shell scripting

\* We don't write scripts for `bash`, but for `busybox`'s `ash` shell, which is POSIX compliant (plus very few features from `bash`).

\* Use `shellcheck` to test your changes for issues before submitting. There is even an [online](https://www.shellcheck.net) version.

\* We're looking into automatizing this more, some files already get checked automatically by the [static code analysis script](https://github.com/postmarketOS/pmbootstrap/blob/master/test/static\_code\_analysis.sh).

### Python

\* We use the [PEP8](https://www.python.org/dev/peps/pep-0008/) standard for Python code. Don't worry, you don't need to read all that, just run the `autopep8` program on your changed code, and confirm with the [static code analyis script](https://github.com/postmarketOS/pmbootstrap/blob/master/test/static\_code\_analysis.sh) that everything is PEP8 compliant. \*This script will run automatically on Travis CI when you make a change request, and it must pass for your code to get accepted.\*

\* We use the `reST` style for `docstrings` below functions (to comment what individual functions are doing, you'll see those when browsing through the code). Please stick to this format, and try to describe the important parameters and return values at least. Example from [here](https://stackoverflow.com/a/24385103):

```Python

"""

This is a reST style.

:param param1: this is a first param

:param param2: this is a second param

:returns: this is a description of what is returned

:raises keyError: raises an exception

"""

```

\* If it is feasible for you, try to run the testsuite on code that you have changed. The `test/test\_build.py` case will build full cross-compilers for `aarch64` and `armhf`, so it may take a long time. Testcases can be started with `pytest` and it's planned to run that automatically when making a new PR (see [#64](https://github.com/postmarketOS/pmbootstrap/issues/64)).

\*\*If you need any help, don't hesitate to open an [issue](https://github.com/postmarketOS/pmbootstrap/issues) and ask!\*\*