# How to contribute

If you want to help with the development of Otter Browser, please observe the following rules.

## Reporting bugs and feature requests

\* Make sure that you have a [GitHub account](https://github.com/join).

\* Check if the issue has already been reported (including closed tickets).

\* Check list of [known upstream issues](https://github.com/OtterBrowser/otter-browser/wiki/Upstream-Bugs).

\* If possible, try to reproduce the issue with current snapshot from relevant branch of the source code repository.

\* Create a ticket for your issue.

\* Describe the issue clearly and succinctly.

\* In case of bug reports:

\* describe the steps required to reproduce issue,

\* attach a [backtrace](http://en.wikipedia.org/wiki/Stack\_trace) if you have any,

\* post details about your setup:

\* application version (for example, 1.0.01 32 bit),

\* system version (for example, Ubuntu 18.04 64 bit),

\* Qt version (for example, Qt 5.6.0 MSVC 64 bit).

## Submitting patches

\* Make sure that you have a [GitHub account](https://github.com/join).

\* Ensure that nobody is currently working on the selected task (check if someone is assigned to the ticket or ask on \*#otter-browser\* at freenode).

\* Fork the repository on GitHub.

\* Create a branch just for that task.

\* Submit your changes as a pull request so they can be reviewed.

\* Patiently wait for the review.

## Coding rules

\* Use the [Allman coding style](http://en.wikipedia.org/wiki/Indent\_style#Allman\_style) with tabs for indentation.

\* Follow a naming scheme that is consistent with the existing code.

\* Use \*\*const\*\* as often as possible (both for local variables and method signatures).

\* Wrap text correctly using [``QString``, ``QStringLiteral`` or ``QLatin1String``](http://woboq.com/blog/qstringliteral.html). Remember to use ``tr()`` for translateable texts.

\* Avoid long lines but also try not to break up if-statements etc. Any good editor should be able to wrap long lines.

\* Try to keep a proper order of methods in the source file (check existing code in case of doubt):

\* constructor(s),

\* destructor,

\* methods returning void (reimplementations of ``\*Event`` methods and others should go first),

\* setters,

\* methods returning value (from pointers through data structures to primitive data types).