# Contributing to GNU Radio

Welcome! You are reading about how to contribute code to GNU Radio. First of

all, we are very happy that you're about to contribute, and welcome your

submissions! We hope many more will come.

In this document, we will explain the main things to consider when submitting

pull requests against GNU Radio. Reading this first will help a lot with

streamlining the process of getting your code merged.

There is also a [wiki-based version of this file][wikicontrib], which contains

more detail. This file is more of a checklist.

## New to GNU Radio?

If you're new, it is a good idea to first get acquainted with GNU Radio from a

user perspective. https://tutorials.gnuradio.org/ is a good place to start.

## What about non-code contributions?

Those are at least as important as code contributions: Emails to the mailing

list, answers on Stack Overflow, Wiki page edits, examples... We very much

appreciate those. However, this document is specifically about contributing

code.

## CLA filed?

The GNU Radio codebase's copyright belongs to the Free Software Foundation. Any

submission that gets merged must therefore be assigned to the FSF. Exceptions

are small or obvious changes (< 10 lines, typically) or documentation changes.

See also [What's this Copyright Assignment?][cla].

## Coding Guidelines

We have codified our coding guidelines in [GREP1][grep1]. Please read them, and

stick to them. For C++ code, use clang-format. For Python, PEP8 is your friend

(but again, check the actual coding guidelines).

## Python or C++?

Python is nice, and generally simpler. However, all performance-critical code

should be written in C++. The same goes for all blocks, since there are use

cases for C++-only GNU Radio.

## Git commit messages are very important

We follow standard git commit message guidelines, similar to many other open

source projects. See the [coding guidelines][grep1] for more details. In a

nutshell:

- Keep the lines below 72 characters

- Subject line has the component prepended (e.g., `runtime:`)

- Avoid empty git commit messages

- The git commit message explains the change, the code only explains the current

state

## Include Unit Tests

If you have an obvious test, that might speed up the time it takes to convince

reviewers that your code is correct.

## GitHub-specific Actions

GNU Radio is currently hosted at GitHub, where we've added some mechanics that

help with getting code merged. If you intend to develop code more than once,

request being accepted into the [gr-devs][gr-devs] group. Members of this group

can change labels and approve pull requests or request changes.

## The Buddy Principle: Submit One, Review One

When you've submitted a pull request, please take the time to review another

one. This helps make sure that there are always a number of reviews at least

equal to the number of pull requests, which means the maintainers don't get

overwhelmed when a lot is being contributed.

## Standard command line options

When writing programs that are executable from the command line,

please follow existing examples regarding their command line arguments, and

reuse them.

[grep1]: https://github.com/gnuradio/greps/blob/master/grep-0001-coding-guidelines.md

[cla]: https://wiki.gnuradio.org/index.php/Development#What.27s\_this\_Copyright\_Assignment.3F

[wikicontrib]: https://wiki.gnuradio.org/index.php/Development

[gr-devs]: https://github.com/orgs/gnuradio/teams/gr-devs