Thanks for considering to write a Pull Request (PR) for CLI11! Here are a few guidelines to get you started:

Make sure you are comfortable with the license; all contributions are licensed under the original license.

## Adding functionality

Make sure any new functions you add are are:

\* Documented by `///` documentation for Doxygen

\* Mentioned in the instructions in the README, though brief mentions are okay

\* Explained in your PR (or previously explained in an Issue mentioned in the PR)

\* Completely covered by tests

In general, make sure the addition is well thought out and does not increase the complexity of CLI11 needlessly.

## Things you should know:

\* Once you make the PR, tests will run to make sure your code works on all supported platforms

\* The test coverage is also measured, and that should remain 100%

\* Formatting should be done with pre-commit, otherwise the format check will not pass. However, it is trivial to apply this to your PR, so don't worry about this check. If you do want to run it, see below.

\* Everything must pass clang-tidy as well, run with `-DCLI11\_CLANG\_TIDY=ON` (if you set `-DCLI11\_CLANG\_TIDY\_OPTIONS="-fix"`, make sure you use a single threaded build process, or just build one example target).

\* Your changes must also conform to most of the [Google C++ Style Guide](https://google.github.io/styleguide/cppguide.html) rules checked by [cpplint](https://github.com/cpplint/cpplint). For unused cpplint filters and justifications, see [CPPLINT.cfg](/CPPLINT.cfg).

## Pre-commit

Format is handled by pre-commit. You should install it:

```bash

python3 -m pip install pre-commit

```

Then, you can run it on the items you've added to your staging area, or all files:

```

pre-commit run

# OR

pre-commit run --all-files

```

And, if you want to always use it, you can install it as a git hook (hence the name, pre-commit):

```bash

pre-commit install

```

## For developers releasing to Conan.io

This is now done by the CI system on tagged releases. Previously, the steps to make a Conan.io release were:

```bash

conan remove '\*' # optional, I like to be clean

conan create . cliutils/stable

conan upload "\*" -r cli11 --all

```

Here I've assumed that the remote is `cli11`.

## For maintainers: remember to add contributions

In a commit to a PR, just add "`@all-contributors please add <username> for <contributions>`" or similar (see <https://allcontributors.org>). Use `code` for code, `bug` if an issue was submitted, `platform` for packaging stuff, and `doc` for documentation updates.

To run locally, do:

```bash

yarn add --dev all-contributors-cli

yarn all-contributors add username code,bug

```

## For maintainers: Making a release

Remember to replace the emoji in the readme, being careful not to replace the ones in all-contributors if any overlap.

Steps:

\* Update changelog if needed

\* Update the version in `.appveyor.yml` and `include/CLI/Version.hpp`.

\* Find and replace in README:

\* Replace " ?" and "? " with "" (ignores the description line)

\* Check for `\/\/$` (vi syntax) to catch leftover `// ?`

\* Replace "?" with "?" (manually ignore the description line)