# Contributing to greatest

Thanks for taking time to contribute to greatest!

Please send patches or pull requests against the `develop` branch. This

makes it easier to avoid interface changes until they can be reflected

in version number updates.

Sending changes via patch or pull request acknowledges that you are

willing and able to contribute it under this project's license. (Please

don't contribute code you aren't legally able to share.)

## Bug Reports

Please report bugs at [the Issues page](https://github.com/silentbicycle/greatest/issues).

If you are reporting a bug, please include:

+ Your operating system name and version.

+ Your compiler version and target platform.

+ Any details about your local setup that might be helpful in

troubleshooting.

+ Detailed steps to reproduce the bug.

## Documentation

Improvements to the documentation are welcome. So are requests for

clarification -- if the docs are unclear or misleading, that's a

potential source of bugs.

## Portability

greatest tries to assume little about its environment. It targets ANSI C

(C89) as a baseline, and features that are specific to C99 or later need

to be wrapped in a version check.

It doesn't require a particular OS, or that there is an OS at all. (It

may be running tests on an embedded system without an RTOS.) It uses

`fprintf(3)` for reporting, that's about it.

Improvements that don't fit the portability requirements can go in

`contrib/`, just not the main `greatest.h` header.

## contrib/: Extensions and Other Tools

There is a `contrib/` directory for extensions. This could include

scripts that generate test templates, add formatting to the reports, or

better integrate greatest into build / continuous integration systems.

Extension libraries that depend on dynamic allocation or

platform-specific features can also go here.

Please include a license in any standalone scripts or extensions.

## Versioning & Compatibility

The versioning format is MAJOR.MINOR.PATCH.

Improvements or minor bug fixes that do not break compatibility with

past releases lead to patch version increases. API changes that do not

break compatibility lead to minor version increases and reset the patch

version, and changes that do break compatibility lead to a major version

increase.

The version will be increased during the merge to master, as part of

the release process.