# Contributing to Calculator

The Calculator team encourages community feedback and contributions. Thank you for your interest in

making Calculator better! There are several ways you can get involved.

## Reporting issues and suggesting new features

If Calculator is not working properly, please file a report in the

[Feedback Hub](https://insider.windows.com/en-us/fb/?contextid=130&newFeedback=True).

Feedback Hub reports automatically include diagnostic data, such as the version of Calculator

you're using.

We are happy to hear your ideas for the future of Calculator. Check the

[Feedback Hub](https://insider.windows.com/en-us/fb/?contextid=130) and see if others have

submitted similar feedback. You can upvote existing feedback or submit a new suggestion.

We always look at upvoted items in Feedback Hub when we decide what to work on next. We read the

comments in both Feedback Hub and GitHub, and we look forward to hearing your input. Remember that

all community interactions must abide by the [Code of Conduct](CODE\_OF\_CONDUCT.md).

## Finding issues you can help with

Looking for something to work on?

Issues marked [``good first issue``](https://github.com/Microsoft/calculator/labels/good%20first%20issue)

are a good place to start.

You can also check the [``help wanted``](https://github.com/Microsoft/calculator/labels/help%20wanted) tag to find

other issues to help with. If you're interested in working on a fix, leave a comment to let everyone know and to help

avoid duplicated effort from others.

## Contributions we accept

We welcome your contributions to the Calculator project, especially to fix bugs and to make

improvements which address the top issues reported by Calculator users. Some general guidelines:

\* \*\*DO\*\* create one pull request per Issue, and ensure that the Issue is linked in the pull request.

\* \*\*DO\*\* follow our [Coding and Style](#style-guidelines) guidelines, and keep code changes as small as possible.

\* \*\*DO\*\* include corresponding tests whenever possible.

\* \*\*DO\*\* check for additional occurrences of the same problem in other parts of the codebase before submitting your PR.

\* \*\*DO\*\* [link the issue](https://github.com/blog/957-introducing-issue-mentions) you are addressing in the

pull request.

\* \*\*DO\*\* write a good description for your pull request. More detail is better. Describe \*why\* the change is being

made and \*why\* you have chosen a particular solution. Describe any manual testing you performed to validate your change.

\* \*\*DO NOT\*\* submit a PR unless it is linked to an Issue marked

[`triage approved`](https://github.com/Microsoft/calculator/issues?q=is%3Aissue+is%3Aopen+label%3A%22Triage%3A+Approved%22).

This enables us to have a discussion on the idea before anyone invests time in an implementation.

\* \*\*DO NOT\*\* merge multiple changes into one PR unless they have the same root cause.

\* \*\*DO NOT\*\* submit pure formatting/typo changes to code that has not been modified otherwise.

We follow a [user-centered process for developing features](docs/NewFeatureProcess.md). New features

need sponsorship from the Calculator team, but we welcome community contributions at many stages of

the process.

> Submitting a pull request for an approved Issue is not a guarantee it will be approved.

> The change must meet our high bar for code quality, architecture, and performance, as well as

> [other requirements](#docs/NewFeatureProcess.md#technical-review).

## Making changes to the code

### Preparing your development environment

To learn how to build the code and run tests, follow the instructions in the [README](README.md).

### Style guidelines

The code in this project uses several different coding styles, depending on the age and history of

the code. Please attempt to match the style of surrounding code as much as possible. In new

components, prefer the patterns described in the

[C++ core guidelines](https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines)

and the [modern C++/WinRT language projections](https://docs.microsoft.com/en-us/windows/uwp/cpp-and-winrt-apis/).

### Testing

Your change should include tests to verify new functionality wherever possible. Code should be

structured so that it can be unit tested independently of the UI. [Manual test cases](docs/ManualTests.md)

should be used where automated testing is not feasible.

### Git workflow

Calculator uses the [GitHub flow](https://guides.github.com/introduction/flow/) where most

development happens directly on the `master` branch. The `master` branch should always be in a

healthy state which is ready for release.

If your change is complex, please clean up the branch history before submitting a pull request.

You can use [git rebase](https://docs.microsoft.com/en-us/azure/devops/repos/git/rebase#squash-local-commits)

to group your changes into a small number of commits which we can review one at a time.

When completing a pull request, we will generally squash your changes into a single commit. Please

let us know if your pull request needs to be merged as separate commits.

## Review Process

After submitting a pull request, members of the calculator team will review your code. We will

assign the request to an appropriate reviewer. Any member of the community may

participate in the review, but at least one member of the Calculator team will ultimately approve

the request.

Often, multiple iterations will be needed to responding to feedback from reviewers. Try looking at

[past pull requests](https://github.com/Microsoft/calculator/pulls?q=is%3Apr+is%3Aclosed) to see

what the experience might be like.

## Contributor License Agreement

Before we can review and accept a pull request from you, you'll need to sign a Contributor License

Agreement (CLA). The CLA ensures that the community is free to use your contributions. More

information about the agreement is available on the [Microsoft web site](https://cla.opensource.microsoft.com/).

Signing the CLA is an automated process, and you only need to do it once for Microsoft projects on

GitHub.

You don't need to sign a CLA until you're ready to create a pull request. When your pull request is

created, it is classified by a bot. If the change is trivial (i.e. you just fixed a typo) then the

bot will label the PR `cla-not-required`. Otherwise, it's classified as `cla-required`. In that

case, the system will also tell you how you can sign the CLA. Once you have signed a CLA, the

current and all future pull requests will be labeled as `cla-signed`.