# Contributing

Take a look through the [Wiki pages](https://github.com/gchq/CyberChef/wiki) for guides on [compiling CyberChef](https://github.com/gchq/CyberChef/wiki/Getting-started) and [adding new operations](https://github.com/gchq/CyberChef/wiki/Adding-a-new-operation).

There are lots of opportunities to contribute to CyberChef. If you want ideas, take a look at any [Issues](https://github.com/gchq/CyberChef/issues) tagged with '[help wanted](https://github.com/gchq/CyberChef/labels/help%20wanted)'.

Before your contributions can be accepted, you must:

- Sign the [GCHQ Contributor Licence Agreement](https://cla-assistant.io/gchq/CyberChef)

- Push your changes to your fork.

- Submit a pull request.

## Coding conventions

\* Indentation: Each block should consist of 4 spaces

\* Object/namespace identifiers: CamelCase

\* Function/variable names: camelCase

\* Constants: UNDERSCORE\_UPPER\_CASE

\* Source code encoding: UTF-8 (without BOM)

\* All source files must end with a newline

\* Line endings: UNIX style (\n)

## Design Principles

1. If at all possible, all operations and features should be client-side and not rely on connections to an external server. This increases the utility of CyberChef on closed networks and in virtual machines that are not connected to the Internet. Calls to external APIs may be accepted if there is no other option, but not for critical components.

2. Latency should be kept to a minimum to enhance the user experience. This means that operation code should sit on the client and be executed there. However, as a trade-off between latency and bandwidth, operation code with large dependencies can be loaded in discrete modules in order to reduce the size of the initial download. The downloading of additional modules must remain entirely transparent so that the user is not inconvenienced.

3. Large libraries should be kept in separate modules so that they are not downloaded by everyone who uses the app, just those who specifically require the relevant operations.

4. Use Vanilla JS if at all possible to reduce the number of libraries required and relied upon. Frameworks like jQuery, although included, should not be used unless absolutely necessary.

With these principles in mind, any changes or additions to CyberChef should keep it:

- Standalone

- Efficient

- As small as possible