Contributing to HTML5 Boilerplate

• <u>HTML5 Boilerplate</u> and want to get involved? Thanks! We're actively looking for folks interested in helping out and there are plenty of ways you can help!

Please take a moment to review this document in order to make the contribution process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project. In return, they should reciprocate that respect in addressing your issue or assessing patches and features.

Using the issue tracker

The <u>issue tracker</u> is the preferred channel for <u>bug reports</u>, <u>features requests</u> and <u>submitting pull requests</u>, but please respect the following restrictions:

- Please **do not** use the issue tracker for personal support requests (use <u>Stack Overflow</u>).
- Please do not derail or troll issues. Keep the discussion on topic and respect the opinions of others.
- Please do not open issues or pull requests regarding the code in https://www.html.ncess.not.ncess.no

Bug reports

A bug is a *demonstrable problem* that is caused by the code in the repository. Good bug reports are extremely helpful - thank you!

Guidelines for bug reports:

- 1. **Use the GitHub issue search** check if the issue has already been reported.
- 2. **Check if the issue has been fixed** try to reproduce it using the latest main or development branch in the repository.
- 3. **Isolate the problem** ideally create a <u>reduced test case</u> and a live example.

A good bug report shouldn't leave others needing to chase you up for more information. Please try to be as detailed as possible in your report. What is your environment? What steps will reproduce the issue? What browser(s) and OS experience the problem? What would you expect to be the outcome? All these details will help people to fix any potential bugs.

Example:

Short and descriptive example bug report title

A summary of the issue and the browser/OS environment in which it occurs. If suitable, include the steps required to reproduce the bug.

- 1. This is the first step
- 2. This is the second step
- 3. Further steps, etc.

```
<ur><ur1> - a link to the reduced test case
```

Any other information you want to share that is relevant to the issue being reported. This might include the lines of code that you have identified as causing the bug, and potential solutions (and your opinions on their merits).

Feature requests

Feature requests are welcome. But take a moment to find out whether your idea fits with the scope and aims of the project. It's up to *you* to make a strong case to convince the project's developers of the merits of this feature. Please provide as much detail and context as possible.

Pull requests

Good pull requests - patches, improvements, new features - are a fantastic help. They should remain focused in scope and avoid containing unrelated commits.

Please ask first before embarking on any significant pull request (e.g. implementing features, refactoring code, porting to a different language), otherwise you risk spending a lot of time working on something that the project's developers might not want to merge into the project.

Please adhere to the coding conventions used throughout a project (indentation, accurate comments, etc.) and any other requirements (such as test coverage).

Adhering to the following process is the best way to get your work included in the project:

1. Fork the project, clone your fork, and configure the remotes:

```
# Clone your fork of the repo into the current directory
git clone https://github.com/<your-username>/html5-boilerplate.git
# Navigate to the newly cloned directory
cd html5-boilerplate
# Assign the original repo to a remote called "upstream"
git remote add upstream https://github.com/h5bp/html5-boilerplate.git
```

2. If you cloned a while ago, get the latest changes from upstream:

```
git checkout main
git pull upstream main
```

3. Create a new topic branch (off the main project development branch) to contain your feature, change, or fix:

```
git checkout -b <topic-branch-name>
```

- 4. Commit your changes in logical chunks. Please adhere to these <u>git commit message guidelines</u> or your code is unlikely be merged into the main project. Use Git's <u>interactive rebase</u> feature to tidy up your commits before making them public.
- 5. Locally merge (or rebase) the upstream development branch into your topic branch:

```
git pull [--rebase] upstream main
```

6. Push your topic branch up to your fork:

```
git push origin <topic-branch-name>
```

7. Open a Pull Request with a clear title and description.

IMPORTANT: By submitting a patch, you agree to allow the project owners to license your work under the terms of the <u>MIT License</u>.