Contributing to Bootstrap

Looking to contribute something to Bootstrap? **Here's how 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 issue tracker is the preferred channel for bug reports, features requests and submitting pull requests, but please respect the following restrictions:

- Please **do not** use the issue tracker for personal support requests. Stack Overflow (`bootstrap-5` tag), Slack or IRC are better places to get help.
- Please **do not** derail or troll issues. Keep the discussion on topic and respect the opinions of others.
- Please **do not** post comments consisting solely of "+1" or ":thumbsup:". Use GitHub's "reactions" feature instead. We reserve the right to delete comments which violate this rule.
- Please **do not** open issues regarding the official themes offered on https://themes.getbootstrap.com/. Instead, please email any questions or feedback regarding those themes to `themes AT getbootstrap DOT com` .

Issues and labels

Our bug tracker utilizes several labels to help organize and identify issues. Here's what they represent and how we use them:

- `browser bug` - Issues that are reported to us, but actually are the result of a browser-specific bug. These are diagnosed with reduced test cases and result in an issue opened on that browser's own bug tracker.
- `confirmed` - Issues that have been confirmed with a reduced test case and identify a bug in Bootstrap.
- `css` - Issues stemming from our compiled CSS or source Sass files.
- `docs` - Issues for improving or updating our documentation.
- `examples` - Issues involving the example templates included in our docs.
- `feature` - Issues asking for a new feature to be added, or an existing one to be extended or modified. New features require a minor version bump (e.g., `v3.0.0` to `v3.1.0`).
- `build` - Issues with our build system, which is used to run all our tests, concatenate and compile source files, and more.
- `help wanted` - Issues we need or would love help from the community to resolve.
- `js` - Issues stemming from our compiled or source JavaScript files.
- `meta` - Issues with the project itself or our GitHub repository.

For a complete look at our labels, see the project labels page.

Bug reports

A bug is a *demonstrable problem* that is caused by the code in the repository. Good bug reports are extremely helpful, so thanks!

Guidelines for bug reports:

0. **Validate your HTML** to ensure your problem isn't caused by a simple error in your own code.
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 `v4-dev` branch if the issue is about v4) in the repository.
3. **Isolate the problem** — ideally create a reduced test case and a live example. This JS Bin is a helpful template.

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? Do other browsers show the bug differently? 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.*
>
> `<url>` - *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).*

Reporting upstream browser bugs

Sometimes bugs reported to us are actually caused by bugs in the browser(s) themselves, not bugs in Bootstrap per se.

Vendor(s)	Browser(s)	Rendering engine	Bug reporting website(s)	Notes
Mozilla	Firefox	Gecko	https://bugzilla.mozilla.org/enter_bug.cgi	"Core" is normally the right product option to choose.
Apple	Safari	WebKit	https://bugs.webkit.org/enter_bug.cgi?product=WebKit	In Apple's bug reporter, choose "Safari" as

				the product.
Google, Opera	Chrome, Chromium, Opera v15+	Blink	https://bugs.chromium.org/p/chromium/issues/list	Click the "New issue" button.
Microsoft	Edge	Blink	https://developer.microsoft.com/en-us/microsoft-edge/	Go to "Help > Send Feedback" from the browser

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. For trivial things, or things that don't require a lot of your time, you can go ahead and make a PR.

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

Do not edit `bootstrap.css` or `bootstrap.js`, and do not commit any dist files (`dist/` or `js/dist`). Those files are automatically generated by our build tools. You should edit the source files in `/bootstrap/scss/` and/or `/bootstrap/js/src/` instead.

Similarly, when contributing to Bootstrap's documentation, you should edit the documentation source files in the `/bootstrap/site/content/docs/` directory of the `main` branch. **Do not edit the `gh-pages` branch.** That branch is generated from the documentation source files and is managed separately by the Bootstrap Core Team.

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>/bootstrap.git
# Navigate to the newly cloned directory
cd bootstrap
# Assign the original repo to a remote called "upstream"
git remote add upstream https://github.com/twbs/bootstrap.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 git commit message guidelines or your code is unlikely be merged into the main project. Use Git's interactive rebase 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 against the `main` branch.

IMPORTANT: By submitting a patch, you agree to allow the project owners to license your work under the terms of the MIT License (if it includes code changes) and under the terms of the Creative Commons Attribution 3.0 Unported License (if it includes documentation changes).

Code guidelines

HTML

Adhere to the Code Guide.

- Use tags and elements appropriate for an HTML5 doctype (e.g., self-closing tags).
- Use CDNs and HTTPS for third-party JS when possible. We don't use protocol-relative URLs in this case because they break when viewing the page locally via `file://` .
- Use WAI-ARIA attributes in documentation examples to promote accessibility.

CSS

Adhere to the Code Guide.

- When feasible, default color palettes should comply with WCAG color contrast guidelines.
- Except in rare cases, don't remove default `:focus` styles (via e.g. `outline: none;`) without providing alternative styles. See this A11Y Project post for more details.

JS

- No semicolons (in client-side JS)
- 2 spaces (no tabs)
- strict mode
- "Attractive"

Checking coding style

Run `npm run test` before committing to ensure your changes follow our coding standards.

License

By contributing your code, you agree to license your contribution under the MIT License. By contributing to the documentation, you agree to license your contribution under the Creative Commons Attribution 3.0 Unported License.

Prior to v3.1.0, Bootstrap's code was released under the Apache License v2.0.