# How to contribute

It's important to us that you feel you can contribute towards the evolution of Phaser. This can take many forms: from helping to fix bugs or improve the docs, to adding in new features to the source. This guide should help you in making that process as smooth as possible.

Before contributing, please read the [code of conduct](https://github.com/photonstorm/phaser/blob/master/.github/CODE\_OF\_CONDUCT.md).

## Reporting issues

[GitHub Issues][0] is the place to report bugs you may have found. When submitting a bug please do the following:

\*\*1. Search for existing issues.\*\* Your bug may have already been fixed, or cannot, or will not, be fixed. So be sure to search the issues first before putting in a duplicate issue.

\*\*2. Not sure if it's a bug?.\*\* Please ask on the [forum][4]. If something is blatantly wrong then post it to GitHub. But if you feel it might just be because you're not sure of expected behavior, then it might save us time, and get you a response faster, if you post it to the Phaser forum instead.

\*\*3. Create an isolated and reproducible test case.\*\* If you are reporting a bug, make sure you also have a minimal, runnable, code example that reproduces the problem you have.

\*\*4. Include a live example.\*\* After narrowing your code down to only the problem areas, make use of [jsFiddle][1], [jsBin][2], [CodePen][5], or a link to your live site so that we can view a live example of the problem.

\*\*5. Share as much information as possible.\*\* Include browser version affected, your OS, version of the library, steps to reproduce, etc. "X isn't working!!!1!" will probably just be closed.

## Support Forum

We have a very active [Phaser Support Forum][4]. If you need general support, or are struggling to understand how to do something or need your code checked over, then we would urge you to post it to our forum. There are a lot of friendly devs in there who can help, as well as the core Phaser team, so it's a great place to get support. You're welcome to report bugs directly on GitHub, but for general support we'd always recommend using the forum first.

## Contribute with online one-click setup

You can use Gitpod (a free online VS Code-like IDE) for contributing. With a single click, it will launch a workspace and automatically:

- clone the `phaser` repo.

- install the dependencies.

- run `npm run build`.

- run `npm run watch`.

- clone `phaser-3-examples` and afterwards install dependencies and run `npm start` in there.

So that anyone interested in contributing can start straight away.

[![Open in Gitpod](https://gitpod.io/button/open-in-gitpod.svg)](https://gitpod.io/#https://github.com/photonstorm/phaser)

## Making Changes

I'm assuming you already have a recent version of [Node](https://nodejs.org) installed locally and can run `npm`. This guide is tested and works on both Windows 10 and OS X.

### 1. Checkout the repos

Check-out both the [Phaser repo](https://github.com/photonstorm/phaser) and the [Phaser 3 Examples Repo](https://github.com/photonstorm/phaser3-examples). Make sure the Phaser 3 Examples repo is saved locally in a folder called `phaser3-examples`, which will be the default for most Git clients.

### 2. Matching Directory Levels

Ensure that both repos live at the same depth in your directory structure. For example: `/usr/home/web/phaser` and `/usr/home/web/phaser3-examples`. This is so the dev build scripts in the Phaser repo can safely copy files to `../phaser3-examples` and have them end up in the correct place.

### 3. Install dependencies

Using your console, run `npm install` or `yarn install` as we've configs for both. This process will install a local copy of webpack and a handful of small support scripts. Note that Yarn on Windows seems to have issues making some packages global, so stick with npm if this is the case.

### 4. Webpack

Making sure you've got both repos checked out, and at the same directory level in your filesystem, issue the command `webpack`. If you can't issue the command then webpack may need [installing globally](https://webpack.js.org/guides/installation/). Webpack will build Phaser and if there are any path errors in the code they'll be flagged during the build process.

What you need is the ability to issue the command `webpack` within the v3 folder and have it work.

### 5. ESLint

There is an ESLint configuration and an Editor Configuration in the v3 folder. \*\*Please adhere to them!\*\* Although not enforced in the build process yet, I will be adding that at a later point. There are lots of tools you can install so your editor of choice will check the ESLint config during development.

To test if your code passes our lint config issue the command `npm run lint`.

## Coding style preferences are not contributions

If your PR is doing little more than changing the Phaser source code into a format / coding style that you prefer then we will automatically close it. All PRs must adhere to the coding style already set-out across the thousands of lines of code in Phaser. Your personal preferences for how things should "look" or be structured do not apply here, sorry. PRs should fix bugs, fix documentation or add features. No changes for the sake of change.

## I don't really like git / node.js, but I can fix this bug

That is fine too. While Pull Requests are the best thing in the world for us, they are not the only way to help. You're welcome to post fixes to our forum or even just email them to us. All we ask is that you still adhere to the guidelines presented here re: ESLint, etc.

## Code Style Guide

We provide an .editorconfig and eslint config for you to use, but generally:

- Use 4 spaces for tabs, never tab characters.

- No trailing whitespace, blank lines should have no whitespace.

- Always favor strict equals `===` unless you \*need\* to use type coercion.

- Follow conventions already in the code, and listen to eslint. Our config is set-up for a reason.

Thanks to Chad for creating the original Pixi.js Contributing file which we adapted for Phaser.

[0]: https://github.com/photonstorm/phaser/issues

[1]: http://jsfiddle.net

[2]: http://jsbin.com/

[3]: http://nodejs.org

[4]: https://phaser.discourse.group/

[5]: https://codepen.io/pen?template=YeEWom "Phaser 3 game template"