# Contributing

Thanks for taking the time to contribute! :tada: :confetti\_ball: :+1:

## Getting started

Looking for places to contribute to the codebase? Check out the ["help wanted"](https://github.com/palantir/blueprint/labels/help%20wanted) label.

### Installation

First, ensure you have Node v10+ and the latest [yarn](https://yarnpkg.com) installed on your machine.

As an external contributor, you will have to fork Blueprint in order to contribute code.

Clone your fork onto your machine and then run the following commands to install dependencies:

```sh

git clone git@github.com:<username>/blueprint.git # using ssh

cd blueprint

yarn

yarn verify

```

## Developing

A typical contributor workflow looks like this:

1. Create a new feature branch. We use a format like `[your-initials]/[short-name]`:

`bd/refactor-buttons`.

1. Run `yarn dev` in the root to watch all the code and run the dev app at http://localhost:9000

- You can instead watch package subsets with `yarn dev:core`, `yarn dev:select`, etc.

1. Write some code. :hammer: \*\*Refer to the wiki in this repo for detailed instructions on:\*\*

- [Development practices](https://github.com/palantir/blueprint/wiki/Development-practices)

- [Coding guidelines](https://github.com/palantir/blueprint/wiki/Coding-guidelines)

- [Editor integration](https://github.com/palantir/blueprint/wiki/Editor-integration)

1. Ensure your code is \*\*tested\*\*, \*\*linted\*\*, and \*\*formatted\*\*.

- Add unit tests as necessary when fixing bugs or adding features; run them with `yarn test`

in the relevant `packages/` directory.

- Linting is best handled by your editor for real-time feedback (see

[Editor integration](https://github.com/palantir/blueprint/wiki/Editor-integration)). Run

`yarn lint` to be 100% safe.

- TypeScript lint errors can often be automatically fixed by ESLint. Run lint fixes with `yarn lint-fix`.

- Code formatting is enforced using [Prettier](https://prettier.io/). These errors can be fixed in your editor

through ESLint (make sure you have set up the editor integrations linked above).

\_\_Formatting checks will not run\_\_ during the `yarn lint` package script.

Instead, when using the CLI or in a CI environment, you should run the `yarn format` script to fix all

formatting issues across the Blueprint monorepo.

1. Submit a Pull Request on GitHub and fill out the template.

- ?? \_\_DO NOT enable CircleCI for your fork of Blueprint.\_\_ Our build

will run on your fork when you open a PR. You can run NPM scripts locally

to validate before pushing code.

1. Team members will review your code and merge it after approvals.

- You may be asked to make modifications to code style or to fix bugs you may have not noticed.

- Please respond to comments in a timely fashion (even if to tell us you need more time).

- \_Do not\_ amend commits and `push --force` as they break the PR history. Please add more commits; we squash each PR to a single commit on merge.

1. Hooray, you contributed! :tophat: