Thanks for showing interest to contribute to Chakra UI ?, you rock!

When it comes to open source, there are different ways you can contribute, all

of which are valuable. Here's few guidelines that should help you as you prepare

your contribution.

- [Setup](#setup)

- [Development](#development)

- [Tooling](#tooling)

- [Commands](#commands)

- [Package Aliasing and Yarn Workspace](#package-aliasing-and-yarn-workspace)

- [Documentation](#documentation)

- [Storybook](#storybook)

- [Think you found a bug?](#think-you-found-a-bug)

- [Proposing new or changed API?](#proposing-new-or-changed-api)

- [Making a Pull Request?](#making-a-pull-request)

- [Commit Convention](#commit-convention)

- [Steps to PR](#steps-to-pr)

- [Tests](#tests)

- [Want to write a blog post or tutorial](#want-to-write-a-blog-post-or-tutorial)

- [Want to help improve the docs?](#want-to-help-improve-the-docs)

- [License](#license)

## Setup

The following steps will get you setup to contribute changes to this repo:

1. Fork the repo (click the <kbd>Fork</kbd> button at the top right of

[this page](https://github.com/chakra-ui/chakra-ui))

2. Clone your fork locally

```sh

# in a terminal, cd to parent directory where you want your clone to be, then

git clone https://github.com/<your\_github\_username>/chakra-ui.git

cd chakra-ui

```

3. Setup all dependencies and build. Chakra UI uses `yarn` and `lerna`, so run

`yarn bootstrap`. This command will install dependencies, bootstrap the repo

using `lerna` and build all packages.

> If you run into any issues during this step, kindly reach out to the Chakra UI

> React team here: https://discord.gg/dQHfcWF

## Development

To improve our development process, we've setup a couple of systems. Chakra UI

uses a monorepo structure, this means each component is it's own package and can

use consumed independently.

### Tooling

- [Lerna](https://lerna.js.org/) to manage installation of dependencies and

running various scripts. We also have yarn workspaces enabled by default.

- [Storybook](https://storybook.js.org/) for rapid UI component development and

testing

- [Testing Library](https://testing-library.com/) for testing components and

hooks

- [Gatsby](https://www.gatsbyjs.org/) for a blazing fast documentation website.

versioning and changelogs

### Commands

\*\*`yarn boot`\*\*: bootstraps the entire project, symlinks all dependencies for

cross-component development and builds all components.

\*\*`yarn bootstrap`\*\*: bootstraps the entire project and symlinks all

dependencies for cross-component development

\*\*`yarn storybook`\*\*: starts storybook server and loads stories in files that

ends with `.stories.tsx`

\*\*`yarn docs:start`\*\*: run the documentation site locally

\*\*`yarn build`\*\*: run build for all component packages

\*\*`yarn test`\*\*: run test for all component packages

\*\*`yarn release`\*\*: publish changed packages

\*\*`yarn [package] <cmd>`\*\*: Run a command on the specific package you're working

on. You can run `build`, `test`, `lint` commands

#### Package Aliasing and Yarn Workspace

Since we're using lerna monorepo + yarn workspaces by default, this enables us

to run commands within component package directly from the root.

Each component is named this way `@chakra-ui/[component]`. Let's say we want to

build the checkbox component. Here's how to do it:

```bash

yarn workspace @chakra-ui/button build

# or

lerna run build --scope @chakra-ui/button

```

\*\*Shortcut:\*\* To make this shorter and more convenient, we've added an alias for

each component in the root `package.json`. Now you can simple do:

```bash

# to build

yarn button build

# to test

yarn button test

yarn button test --watch

# to lint

yarn button lint

```

This alias is particularly useful when you're working on a specific component

and want to avoid running the command for all components.

### Documentation

The documentation site is built with Gatsby. If you'd like to contribute to the

docs, simply run `yarn docs:start`.

### Storybook

Build components in isolation with Storybook using `yarn storybook`.

## Think you found a bug?

Please conform to the issue template and provide a clear path to reproduction

with a code example. The best way to show a bug is by sending a CodeSandbox link

You may wish to use our starters to help you get going:

- JavaScript Starter: https://codesandbox.io/s/chakra-ui-javascript-lzzg9

- TypeScript Starter: https://codesandbox.io/s/chakra-ui-typescript-pomi8

## Proposing new or changed API?

Please provide thoughtful comments and some sample API code. Proposals that

don't line up with our roadmap or doesn't have a thoughtful explanation will be

closed.

## Making a Pull Request?

Pull requests need only the :+1: of two or more collaborators to be merged; when

the PR author is a collaborator, that counts as one.

### Commit Convention

Before you create a Pull Request, please check whether your commits comply with

the commit conventions used in this repository.

When you create a commit we kindly ask you to follow the convention

`category(scope or module): message` in your commit message while using one of

the following categories:

- `feat / feature`: all changes that introduce completely new code or new

features

- `fix`: changes that fix a bug (ideally you will addtionally reference an issue

if present)

- `refactor`: any code related change that is not a fix nor a feature

- `docs`: changing existing or creating new documentation (i.e. README, docs for

usage of a lib or cli usage)

- `build`: all changes regarding the build of the software, changes to

dependencies or the addition of new dependencies

- `test`: all changes regarding tests (adding new tests or changing existing

ones)

- `ci`: all changes regarding the configuration of continous integration (i.e.

github actions, ci system)

- `chore`: all changes to the repository that do not fit into any of the above

categories

If you are interested in the detailed specification you can visit

https://www.conventionalcommits.org/ or check out the

[Angular Commit Message Guidelines](https://github.com/angular/angular/blob/22b96b9/CONTRIBUTING.md#-commit-message-guidelines).

### Steps to PR

- Fork of the chakra-ui repository and clone your fork

- Create a new branch out of the `develop` branch. We follow the convention

`[type/scope]`. For example `fix/accordion-hook`, `docs/menu-typo`

- `type` can be either `docs`, `fix`, `feat`, `build`, or any other

conventional commit type

- `scope` is just a short id that describes the scope of work.

### Tests

All commits that fix bugs or add features need a test.

> \*\*Dear Chakra team:\*\* Please do not merge code without tests

## Want to write a blog post or tutorial

That would be amazing! Reach out to the core team here:

https://discord.gg/dQHfcWF. We would love to support you anyway we can.

## Want to help improve the docs?

TODO

## License

By contributing your code to the chakra-ui GitHub repository, you agree to

license your contribution under the MIT license.