

Overview

Your editor

Recommended text editor features

Linting

Formatting

Editor Setup

A properly configured editor can make code clearer to read and faster to write. It can even help you catch bugs as you write them! If this is your first time setting up an editor or you're looking to tune up your current editor, we have a few recommendations.

You will learn

- What the most popular editors are
- How to format your code automatically

Your editor

[VS Code](#) is one of the most popular editors in use today. It has a large marketplace of extensions and integrates well with popular services like GitHub. Most of the features listed below can be added to VS Code as extensions as well, making it highly configurable!

Other popular text editors used in the React community include:

- [WebStorm](#) is an integrated development environment designed specifically for JavaScript.
- [Sublime Text](#) has support for JSX and TypeScript, [syntax highlighting](#) and autocomplete built in.
- [Vim](#) is a highly configurable text editor built to make creating and changing any kind of text very efficient. It is included as "vi" with most UNIX systems and with Apple OS X.

Recommended text editor features

Some editors come with these features built in, but others might require adding an extension. Check to see what support your editor of choice provides to be sure!

Linting

Code linters find problems in your code as you write, helping you fix them early. [ESLint](#) is a popular, open source linter for JavaScript.

- [Install ESLint with the recommended configuration for React](#) (be sure you have [Node](#) installed!)
- [Integrate ESLint in VSCode with the official extension](#)

Make sure that you've enabled all the [eslint-plugin-react-hooks](#) rules for your project. They are essential and catch the most severe bugs early. The recommended [eslint-config-react-app](#) preset already includes them.

Formatting

The last thing you want to do when sharing your code with another contributor is get into a discussion about [tabs vs spaces](#)! Fortunately, [Prettier](#) will clean up your code by reformatting it to conform to preset, configurable rules. Run Prettier, and all your tabs will be converted to spaces—and your indentation, quotes, etc will also all be changed to conform to the configuration. In the ideal setup, Prettier will run when you save your file, quickly making these edits for you.

You can install the [Prettier extension in VSCode](#) by following these steps:

1. Launch VS Code
2. Use Quick Open (press Ctrl/Cmd+P)
3. Paste in `ext install esbenp.prettier-vscode`
4. Press Enter

Formatting on save

Ideally, you should format your code on every save. VS Code has settings for this!

1. In VS Code, press `CTRL/CMD + SHIFT + P`.
2. Type "settings"
3. Hit Enter
4. In the search bar, type "format on save"
5. Be sure the "format on save" option is ticked!

If your ESLint preset has formatting rules, they may conflict with Prettier. We recommend disabling all formatting rules in your ESLint preset using `eslint-config-prettier` so that ESLint is *only* used for catching logical mistakes. If you want to enforce that files are formatted before a pull request is merged, use `prettier --check` for your continuous integration.