

[API REFERENCE >](#)

# React Reference Overview

This section provides detailed reference documentation for working with React. For an introduction to React, please visit the [Learn](#) section.

The React reference documentation is broken down into functional subsections:

## React

Programmatic React features:

- [Hooks](#) - Use different React features from your components.
- [Components](#) - Built-in components that you can use in your JSX.
- [APIs](#) - APIs that are useful for defining components.
- [Directives](#) - Provide instructions to bundlers compatible with React Server Components.

## React DOM

React DOM contains features that are only supported for web applications (which run in the browser DOM environment). This section is broken into the following:

- [Hooks](#) - Hooks for web applications which run in the browser DOM environment.
- [Components](#) - React supports all of the browser built-in HTML and SVG components.
- [APIs](#) - The `react-dom` package contains methods supported only in web applications.
- [Client APIs](#) - The `react-dom/client` APIs let you render React components on the client (in the browser).
- [Server APIs](#) - The `react-dom/server` APIs let you render React components to HTML on the server.
- [Static APIs](#) - The `react-dom/static` APIs let you generate static HTML for React components.

# React Compiler

The React Compiler is a build-time optimization tool that automatically memoizes your React components and values:

- [Configuration](#) - Configuration options for React Compiler.
- [Directives](#) - Function-level directives to control compilation.
- [Compiling Libraries](#) - Guide for shipping pre-compiled library code.

## ESLint Plugin React Hooks

The [ESLint plugin for React Hooks](#) helps enforce the Rules of React:

- [Lints](#) - Detailed documentation for each lint with examples.

## Rules of React

React has idioms — or rules — for how to express patterns in a way that is easy to understand and yields high-quality applications:

- [Components and Hooks must be pure](#) – Purity makes your code easier to understand, debug, and allows React to automatically optimize your components and hooks correctly.
- [React calls Components and Hooks](#) – React is responsible for rendering components and hooks when necessary to optimize the user experience.
- [Rules of Hooks](#) – Hooks are defined using JavaScript functions, but they represent a special type of reusable UI logic with restrictions on where they can be called.

## Legacy APIs

- [Legacy APIs](#) - Exported from the `react` package, but not recommended for use in newly written code.



Copyright © Meta Platforms, Inc

uwu?

## Learn React

- [Quick Start](#)
- [Installation](#)
- [Describing the UI](#)
- [Adding Interactivity](#)
- [Managing State](#)
- [Escape Hatches](#)

## API Reference

- [React APIs](#)
- [React DOM APIs](#)

## Community

- [Code of Conduct](#)
- [Meet the Team](#)
- [Docs Contributors](#)
- [Acknowledgements](#)

## More

- [Blog](#)
- [React Native](#)
- [Privacy](#)
- [Terms](#)

