What Is React?

What React is, why it exists, the difference between a library and a framework, what an SPA is, and where React is used -- in plain language, with no code yet.

Beginner 8 min

Last updated: August 12, 2026
Read online: https://www.learnforgex.com/en/topics/what-is-react

---

What Is React?

This is the first lesson of the React course. We won't write any code here -- first we'll simply understand what React is, why it exists, and where it's used. In the next lesson we'll set up a real React project, and after that we'll start writing actual code.

What Is React?

React is a JavaScript library for building user interfaces (UI). It was created by Facebook (now Meta).

With React, you split a web page into small, reusable pieces called "components." For example, on an e-commerce site, a "product card," a "cart icon," and a "search box" could each be their own component. You combine these pieces to build the whole page.

Why React?

Before React, changing something on a page (say, incrementing a counter) usually meant touching the DOM (the page's HTML structure) by hand -- you had to write the code that finds the right element and changes it yourself. As a page grew, this became hard to keep track of.

With React, you think instead: "what should the UI look like once the data changes?" You change the data, and React automatically updates the UI to match. This keeps the code simpler and less error-prone.

History (Brief)

React was first released as open source by Facebook in 2013. Since then it has become one of the most widely used UI libraries, adopted by large companies like Facebook/Instagram

as well as countless smaller projects. The biggest change over the years came with Hooks (2019) -- we'll cover this in detail later in this course, in the "Hooks" category.

Library vs. Framework

React describes itself as a "library," not a "framework." The difference is simple:

- A **framework** (like Angular) gives you routing, form handling, HTTP requests, and much more, all built in and following its own rules. You work within its structure.
- A **library** (React) focuses on just one question: "how do you build the UI?" For other needs, like routing or form handling, you pick and add separate libraries yourself (like React Router).

This makes React more flexible, but also means you make a few more decisions up front.

React vs. Vanilla JavaScript

"Vanilla JavaScript" means plain JavaScript, with no libraries at all. For a small page, vanilla JavaScript can be enough. But as a page grows, it gets harder to manually track which element needs updating and when.

React handles that tracking for you: you say "the UI should look like this now," and React figures out what changed and updates it. We'll see this with live examples later, in the "State & Events" category.

What Is an SPA (Single Page Application)?

Apps built with React usually work as an "SPA" (Single Page Application). This means that when a user clicks a link, the browser does **not** reload the entire page from scratch. Instead, JavaScript updates only the part of the page that needs to change.

This makes the app feel faster and smoother -- you never see that brief white "reloading" flash between pages. We'll build this for real, with React Router, in the Routing lesson.

Where Is React Used?

React isn't just for web pages. The same core ideas power:

- **Web:** Regular websites and web applications (the focus of this course).
- **Mobile:** iOS/Android apps, with React Native.
- **Desktop:** Desktop apps, with tools like Electron.

This course focuses only on the web side -- regular React.

Summary and Glossary

React is a JavaScript library that lets you build UIs by splitting them into small components. It's built around the idea that the UI should automatically update whenever the data changes. It's a library, not a framework -- it focuses on the UI layer and leaves the rest (like routing) up to you.

Glossary

React — A JavaScript library for building user interfaces.

Component — A small, reusable piece of a UI (like a button or a card).

Library — A tool that helps you do a specific job (for React: building UIs), while leaving the overall structure up to you.

Framework — A more comprehensive tool that imposes its own rules and structure (like Angular).

SPA (Single Page Application) — A type of app where the browser doesn't fully reload between page transitions -- JavaScript updates only the part that needs to change.