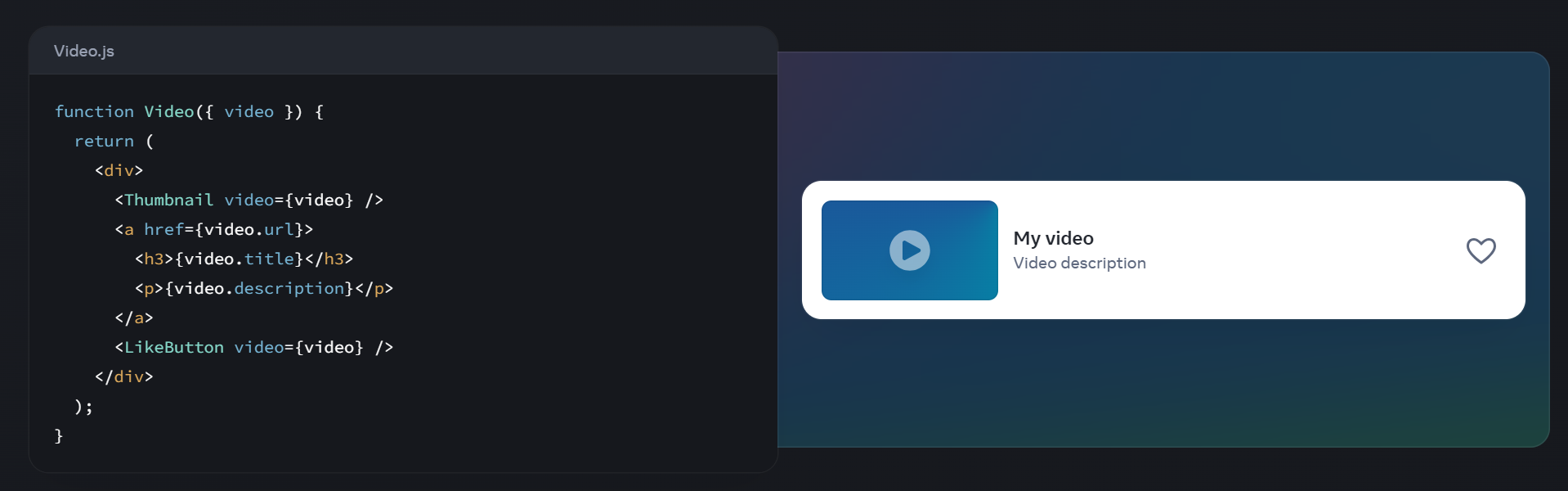


**Create user interfaces from components**

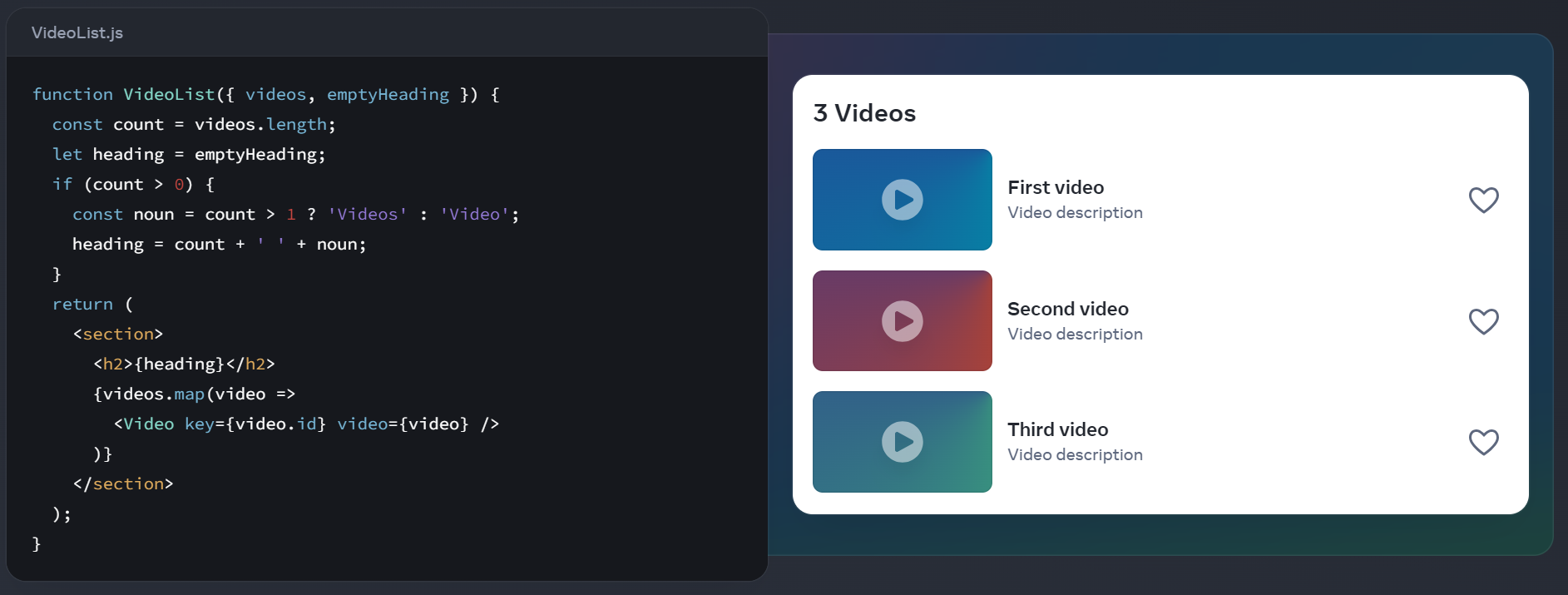
React lets you build user interfaces out of individual pieces called components. Create your own React components like Thumbnail, LikeButton, and Video. Then combine them into entire screens, pages, and apps.



Whether you work on your own or with thousands of other developers, using React feels the same. It is designed to let you seamlessly combine components written by independent people, teams, and organizations.

## Write components with code and markup

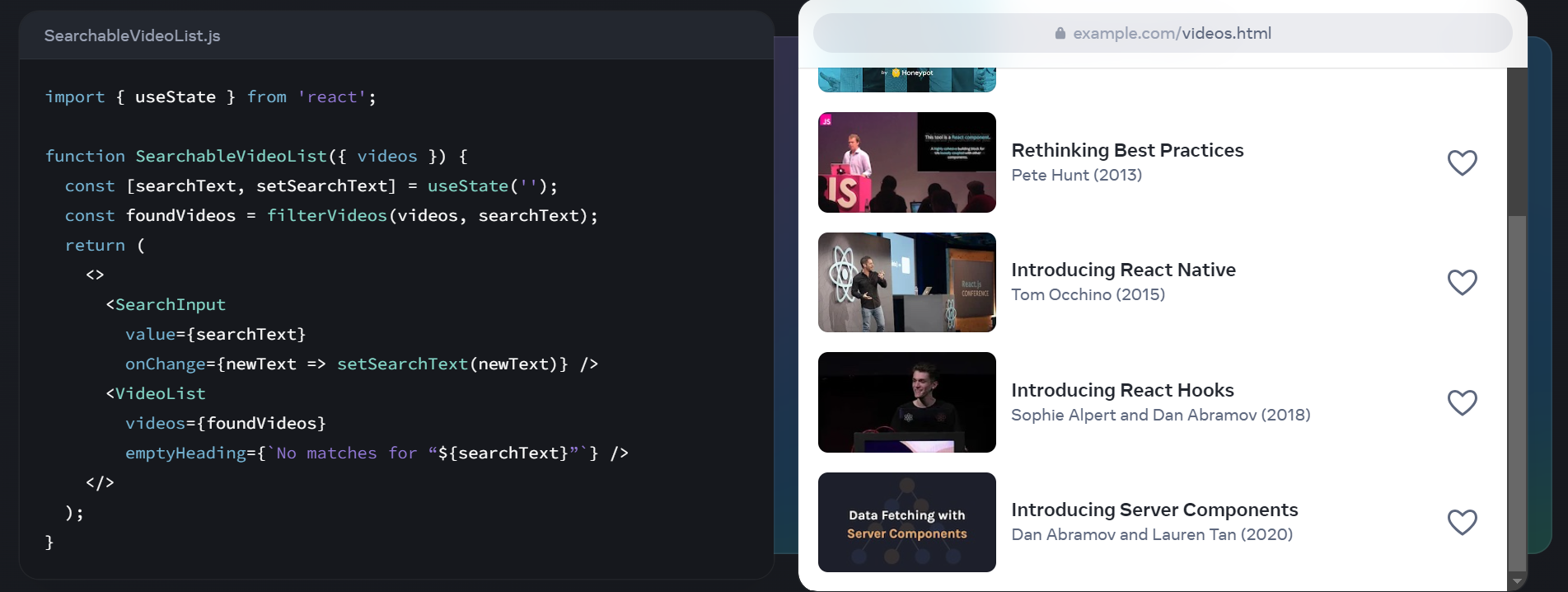
React components are JavaScript functions. Want to show some content conditionally? Use an if statement. Displaying a list? Try array map(). Learning React is learning programming.



This markup syntax is called JSX. It is a JavaScript syntax extension popularized by React. Putting JSX markup close to related rendering logic makes React components easy to create, maintain, and delete.

## Add interactivity wherever you need it

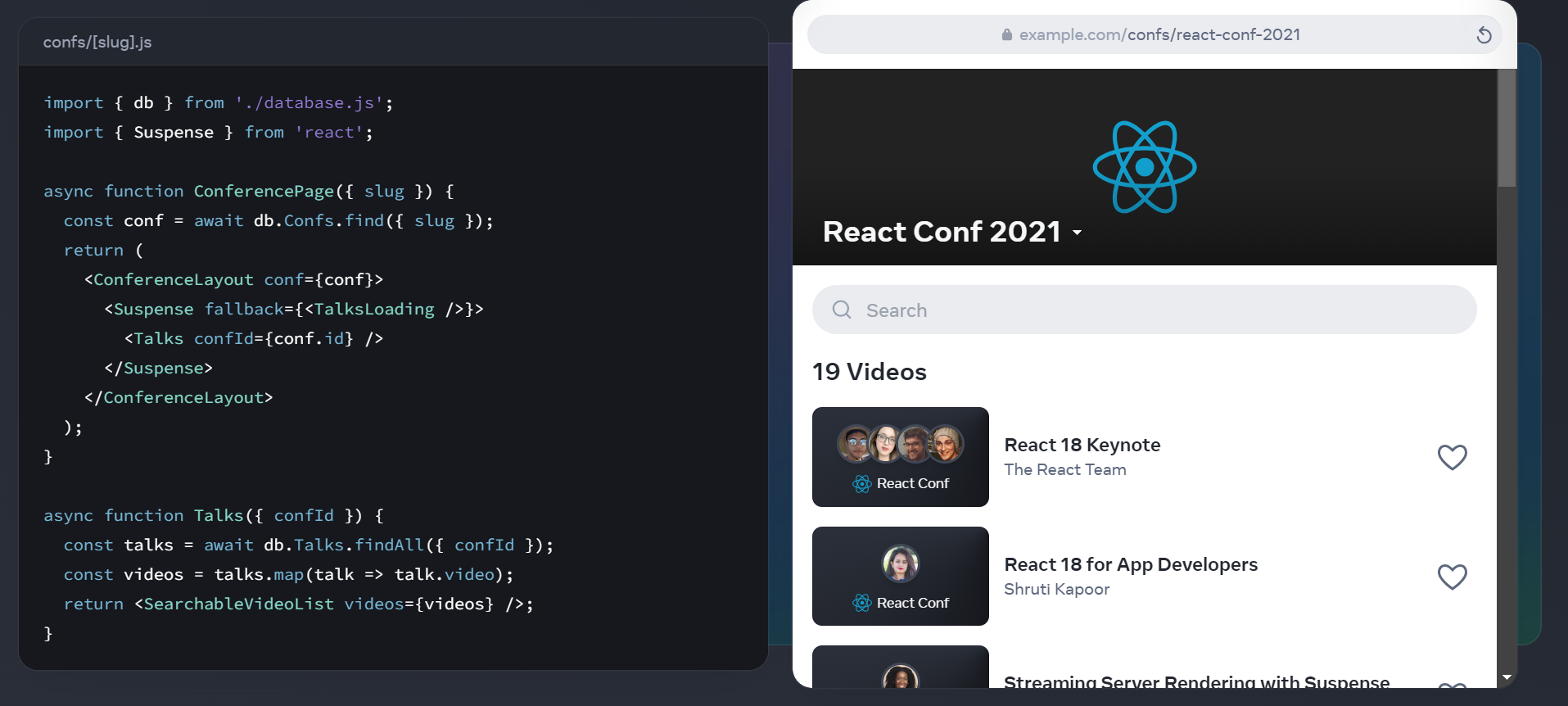
React components receive data and return what should appear on the screen. You can pass them new data in response to an interaction, like when the user types into an input. React will then update the screen to match the new data.



You don’t have to build your whole page in React. Add React to your existing HTML page, and render interactive React components anywhere on it.

## Go full-stack with a framework

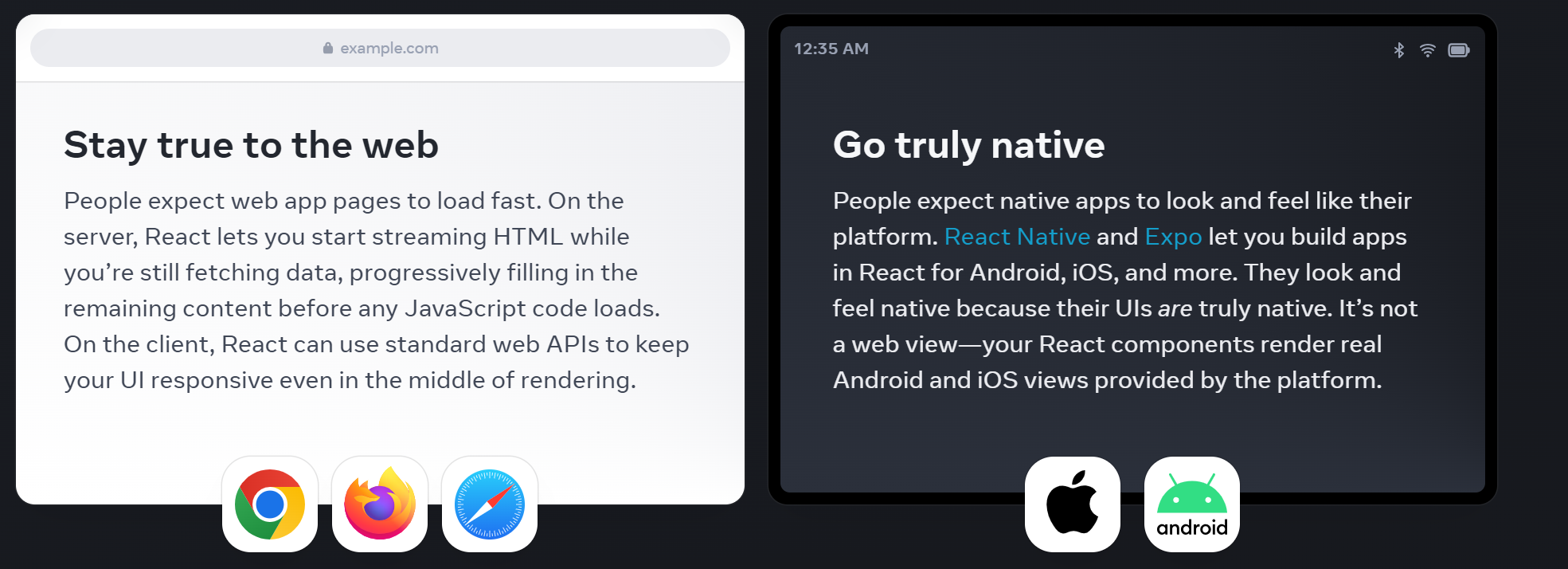
React is a library. It lets you put components together, but it doesn’t prescribe how to do routing and data fetching. To build an entire app with React, we can use full-stack React framework like Next.js or Remix.



React is also an architecture. Frameworks that implement it let you fetch data in asynchronous components that run on the server or even during the build. Read data from a file or a database, and pass it down to your interactive components.

## Use the best from every platform

People love web and native apps for different reasons. React lets you build both web apps and native apps using the same skills. It leans upon each platform’s unique strengths to let your interfaces feel just right on every platform.



With React, you can be a web and a native developer. Your team can ship to many platforms without sacrificing the user experience. Your organization can bridge the platform silos, and form teams that own entire features end-to-end.