

30 DAYS OF REACT

AN INTRODUCTION TO REACT
IN 30 BITE-SIZE MORSELS

are *composable*. A component might include one or more other components in its output.

Broadly speaking, to write React apps we write React components that correspond to various interface elements. We then organize these components inside higher-level components which define the structure of our application.

For example, take a form. A form might consist of many interface elements, like input fields, labels, or buttons. Each element inside the form can be written as a React component. We'd then write a higher-level component, the form component itself. The form component would specify the structure of the form and include each of these interface elements inside of it.

Importantly, each component in a React app abides by strict data management principles. Complex, interactive user interfaces often involve complex data and application state. The surface area of React is limited and aimed at giving us the tools to be able to anticipate how our application will look with a given set of circumstances. We dig into these principles later in the course.

Okay, so how do we use it?

React is a JavaScript framework. Using the framework is as simple as including a JavaScript file in our HTML and using the `React` exports in our application's JavaScript.

For instance, the *Hello world* example of a React website can be as simple as: