



## **React: The Future of Web App Development**

November 22, 2016

Interactive websites are engaging websites - that's why most web pages aren't just static documents anymore, but web applications running in browsers. Of course, for applications to run well, they need solid frameworks, and when it comes to creating and updating these foundations, a small learning curve and only having to write web components once are simple features with big advantages. These are the core features of React, an easy-to-download web framework for easily engineering interactive web applications.

React is a lightweight program solving heavy issues by increasing application interactivity for site visitors. It makes processes like data binding (maintaining front end components when back end changes are made) simple for web developers.

Once web components have been written, React allows for their use in various contexts. For instance, while writing an application in a web page utilizing React components, React's Native software can simultaneously be used with the exact same components. Developers are able to write their base components knowing they can add additional elements on top of these in order to tailor them to different environments. This makes it possible to generate an absolutely Native IOS or Android app using Native UI components, and not just web-based UI-type components.

Another core feature of React is the internal state associated with each component. The components can track either the value of a field or certain other data the developer may associate to them, and they can be used from page to page, on and on to a developer's satisfaction.

A common use of React involves back end notifications. If a user has five notifications which they choose to view, each update will highlight when clicked on, and the number on the notification badge will decrease as each is viewed - with all of these actions being controlled by React.

In addition, some innovators are creating React Hardware and expanding the uses of React components. For example, an R Domino can turn LEDs on and off based on button presses, or a virtual reality environment can be created to render web components in 3D (while updating based on the state of the underlying application). These are just some of the new ideas React is spawning - all of which keep a more interactive end product for site visitors in mind.

React doesn't try to be all things to all people, and that's a positive. While some developers may not prefer a program that's unable to provide control over a whole page, React provides a lightweight program, good documentation, and an easy starting point. For these reasons, we here at Six Feet Up are keeping an eye on React as a major player in the future of interactive and engaging web applications.

<http://www.sixfeetup.com/blog/interactive-web-apps-with-react>