

Why React? By Cory House

Email: housecor@gmail.com | Blog: bitnative.com | Twitter: [@housecor](https://twitter.com/housecor)

Pluralsight Course: pluralsight.com/author/cory-house

React Slingshot: github.com/coryhouse/react-slingshot

Please rate this talk! speakerrate.com/talks/65271

React Innovations

1. JSX = JS-Centric instead of HTML-Centric

- a. "HTML" in JavaScript (as opposed to Angular/KO/Ember who put "JS" in HTML)
- b. Compiles to JavaScript
- c. Optional – Can just write plain JS instead since that's what JSX compiles to
- d. Some minor differences like className, htmlFor
- e. Actually useful having JS and HTML in same file since there's no interface and they must stay in sync
- f. Notably simpler and clearer syntax than Angular or Knockout because it's mostly plain JavaScript.
- g. Related post: React's JSX: The Other Side of the Coin: <http://bit.ly/reactjsx>

2. Virtual DOM

- a. Updating the DOM is expensive
- b. Compares current state to new state. Updates DOM in most efficient way.
- c. Delivers exceptional performance
- d. Not just about performance. Enables React Native, synthetic events, and Isomorphic JS as well.

3. Hot Reloading

- a. Don't lose client-side state when editing code
- b. See your changes immediately without a browser refresh – Check out React Slingshot for example.

4. Isomorphic/Universal Rendering

- a. Run the same code on the client and server
- b. Better perceived performance
- c. SEO – Full HTML is sent to browser
- d. Maintainability – Use single tech and code base for server and client.

5. Unidirectional Data Flow

- a. Traditional two-way binding can create unpredictable data flows
- b. Flux utilizes a unidirectional flow so data flows are easy to reason about and debug
- c. Actions are things that happen in the UI. They're triggered by React
- d. The dispatcher tells everyone about the action that just happened
- e. Stores respond to the dispatcher and update data accordingly

6. Summary

- a. Angular 2 and Ember recently added support 2, 4, and 5. But they can't add 1 without a fundamental rewrite. And it has a HUGE impact on developer ergonomics, debugging, and conceptual simplicity. They're hampered by their HTML-first design.

Is Your Team Writing Clean Code?

I offer on-site training sessions on clean C# and JavaScript development. bitnative.com/training