The Dawn of the Reusable Web: Diving into HTML5 Web Components

Cory House

Email: cory@bitnative.com | Blog: bitnative.com | Twitter: @housecor

Please rate this talk! http://speakerrate.com/talks/44941

Today, Web Developers Have 5 Problems

1. **Undescriptive Markup -** It's hard to read and navigate <div> soup. Divs are generic containers that don't convey the nature of the data inside.
2. **Style and JS Conflicts -** Have to write complex, prefixed selectors to avoid conflicts. No markup encapsulation means JS may accidentally manipulate as well.
3. **No Native Templates -** We use creative hacks to store HTML for templates. Today's approaches create confusion, lead to XSS vulnerabilities, or aren't truly inert.
4. **No Bundling -** There's no simple way to import a component with a single line of code. You have to reference the necessary JS and CSS in the right order and add proprietary markup.
5. **No Component Standard -** Switching between today's proprietary component solutions is painful. jQueryUI, KendoUI, Angular, etc. all have their own style.

The Solution: 4 New Core Technologies

1. Templates – Inert, standards-based, reusable markup within <template>
2. Custom Elements – Define your own HTML elements. Extend HTML to fit your app.
3. Shadow DOM – Encapsulated markup and styling. Hide your markup from manipulation.
4. Imports – Bundle HTML, JS, and CSS. Reference the component you need with a single line.

Core Benefits of Native HTML5 Web Components

1. Descriptive Markup
2. Don't start at square zero
3. Easily replaced and upgraded
4. Fewer integration mistakes
5. Clearer interface and API

Recommended Resources

- HTML5 Web Component Fundamentals on Pluralsight: Pluralsight.com/author/cory-house
- Webcomponents.org
- Component.kitchen
- HTML5Rocks.com