Chrome Dev Summit 2014: Polymer - State of the Union

By Rob Dodson

Rob is a contributor to Web**Fundamentals**

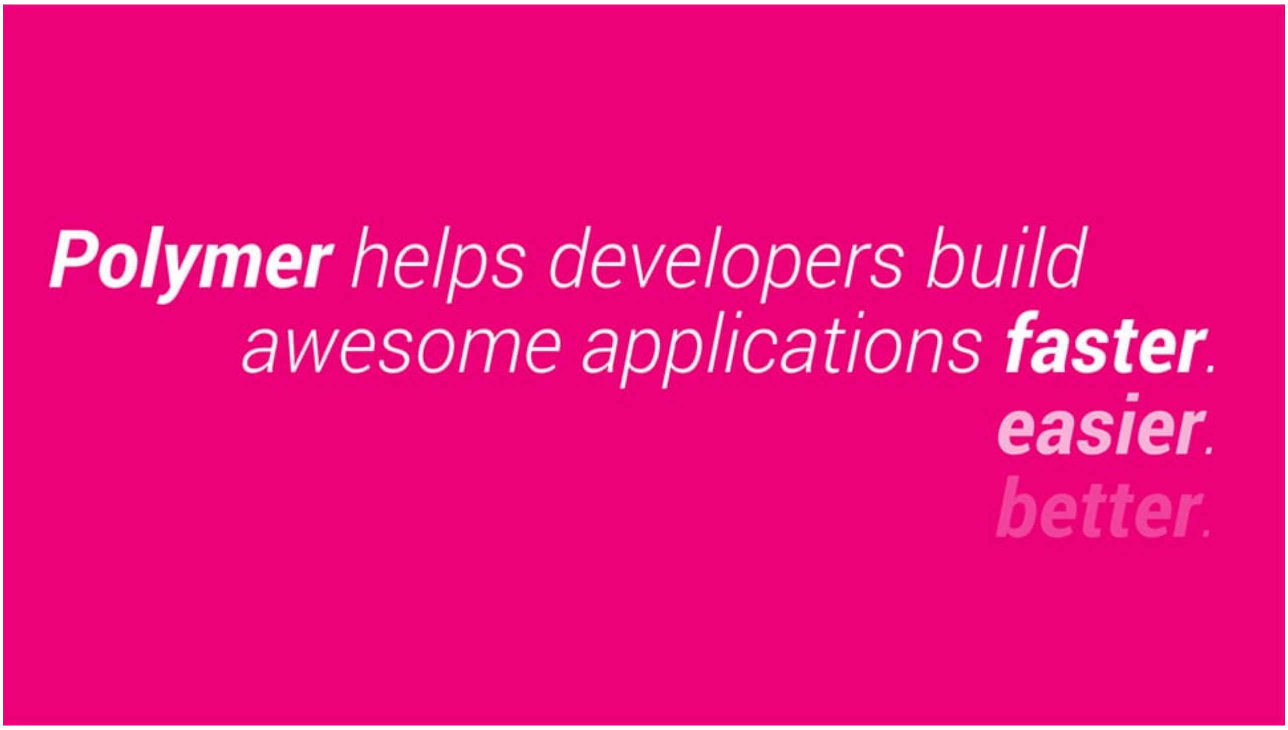
Polymer and Web Components are very hot topics as of late, and since this ecosystem is evolving rapidly, it can often be difficult for developers to stay abreast of all the latest changes.

In his talk at Chrome Dev Summit, Matt McNulty, engineering manager for the Polymer team, explained what Polymer is, and also outlined the roadmap to Polymer 1.0.

What is Polymer?

First, what is Polymer exactly?

Polymer is a library that helps you build elements and apps out of web components. Web Components are a cutting edge set of new standards that allow developers to extend the HTML vocabulary with their own custom elements.



Polymer *helps developers build awesome applications* ***faster.***
easier.
better.

Because Web Components are designed to be a new primitive for the browser, it means that they're very powerful but also very low level and working with them requires a fair bit of code.

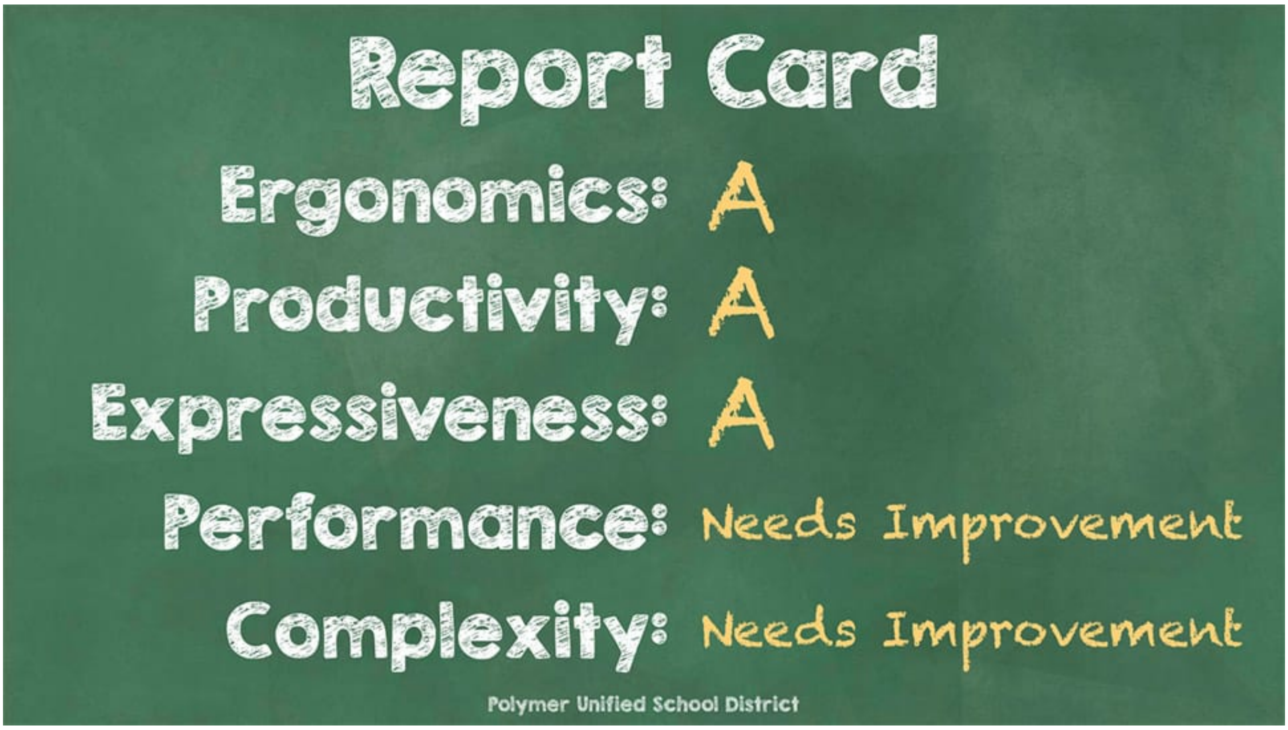
Polymer makes web components **sweeter**

Image: http://bit.ly/1ikUzs4

Polymer makes it easier to work Web Components by "sugaring" the syntax. It reduces the amount of boilerplate code you need to write, and adds a declarative style so creating Web Components is as easy as writing HTML.

The Polymer Experiment

Polymer began as an experiment to see if we could polyfill the Web Component standards and get feedback from developers before these technologies had shipped in all browsers. As more developers began using Polymer, it changed from being just polyfills, to an actual library full of productivity features (data binding, attribute change watchers, automatic node finding, etc). But all experiments have results, so how did we do?



Report Card

Ergonomics: A

Productivity: A

Expressiveness: A

Performance: Needs Improvement

Complexity: Needs Improvement

Polymer Unified School District

While many developer said they liked the expressiveness and productivity gains of working with Web Components in Polymer, they also expressed concerns around performance and overall complexity.

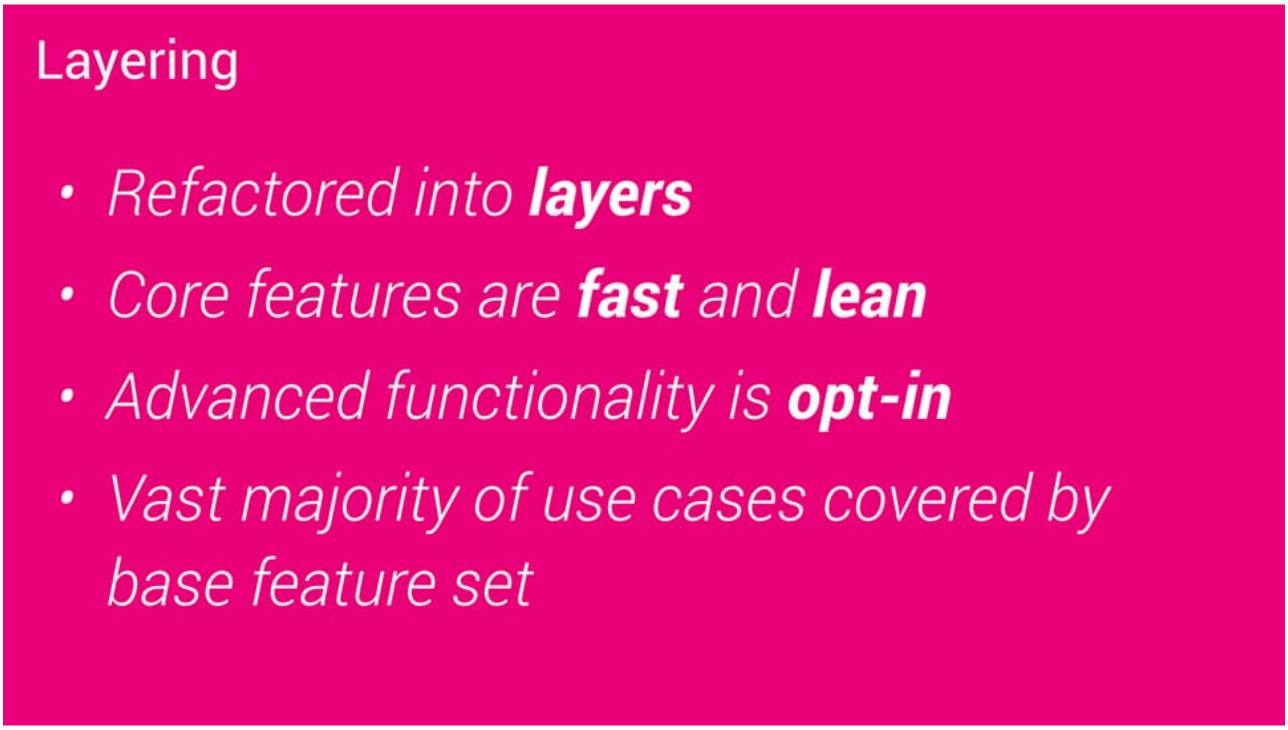
It highlights a natural tension that Polymer has had all along: be an experiment to push the web platform forward, but also create something production-worthy that developers can depend on.

Upcoming Changes

The Polymer team has taken a hard look at every feature of the library with the goal of hammering out a leaner, production-ready version that developers can feel confident about using.

Layers

Polymer has been refactored into a series of layers. The core features are fast and lean, while the more advanced features will be opt-in. For the majority of use cases, the core features should cover developers' needs.

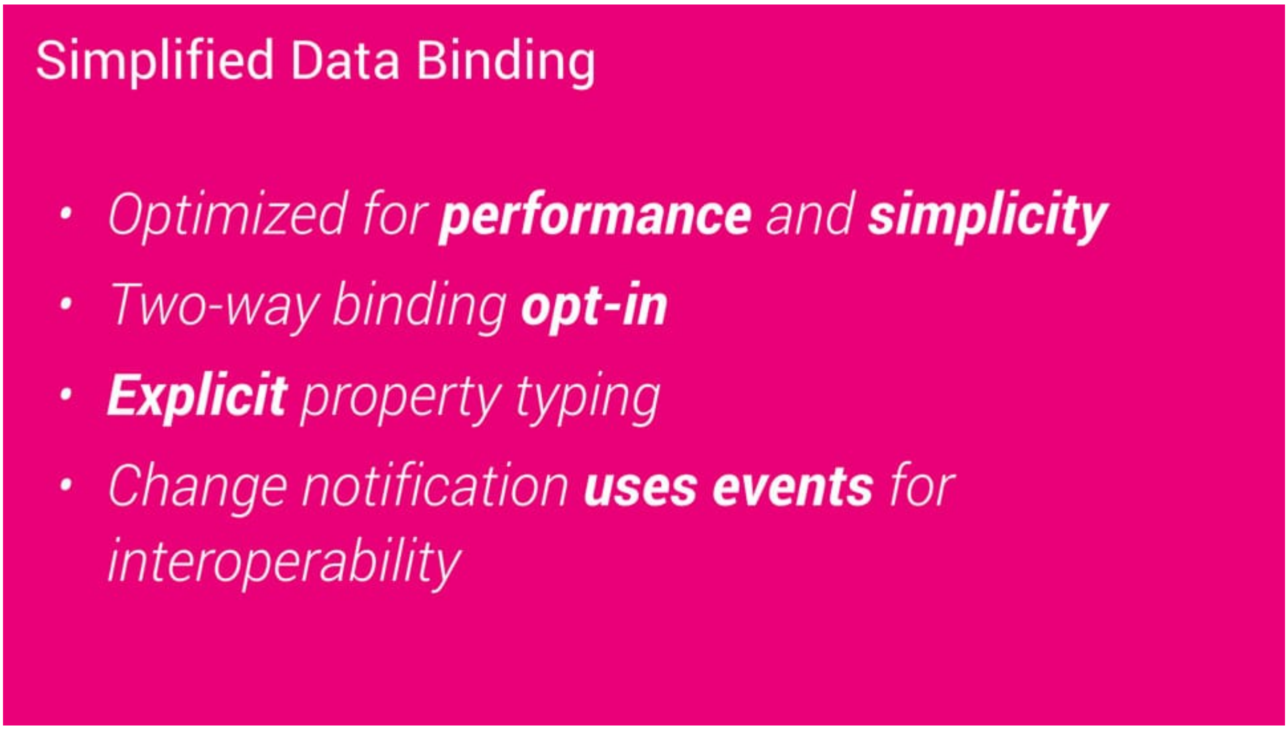


Layering

- *Refactored into **layers***
- *Core features are **fast** and **lean***
- *Advanced functionality is **opt-in***
- *Vast majority of use cases covered by base feature set*

Simplified Data Binding

Polymer's data binding system has also been significantly optimized for performance. Following the layered approach, two-way binding is now opt-in, with the default being one-way bindings. Also, published property types have been made explicit and a property change now fires an event to help elements from different libraries more easily communicate.



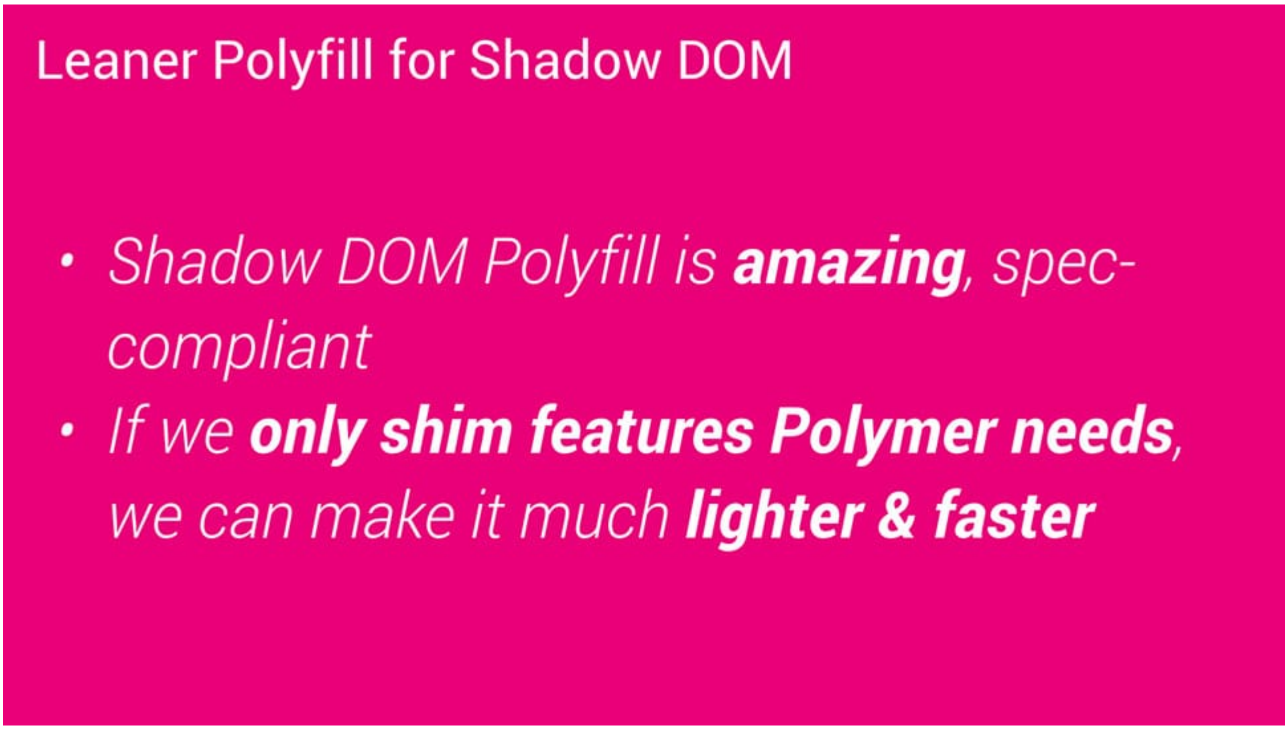
Simplified Data Binding

- *Optimized for **performance** and **simplicity***
- *Two-way binding **opt-in***
- ***Explicit** property typing*
- *Change notification **uses events** for interoperability*

Leaner Shadow DOM

The Shadow DOM polyfill is an amazing feat of engineering. It was designed to be comprehensive and spec compliant, which is important for thoroughly testing the platform primitive, but unfortunately introduces a number of performance bottlenecks for features that Polymer is not using.

The next release of Polymer will take a different approach, employing a shim-style layer that only polyfills what Polymer needs.



Leaner Polyfill for Shadow DOM

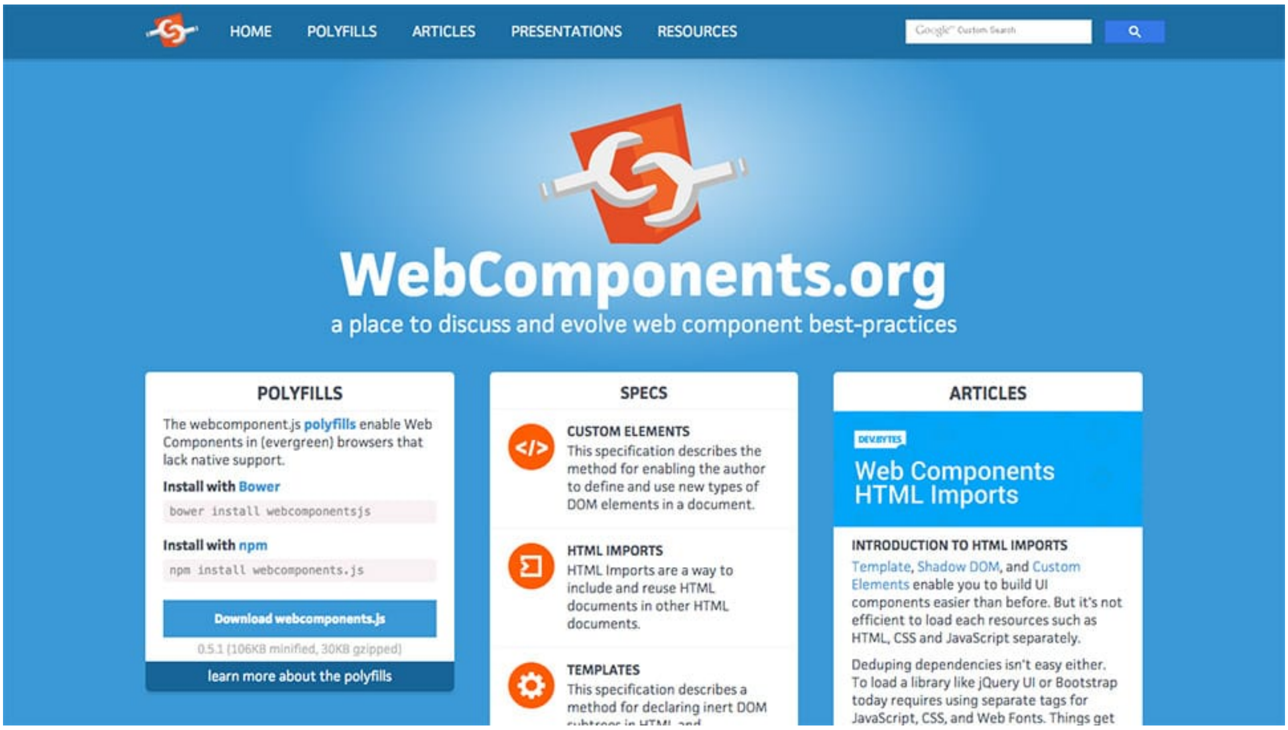
- *Shadow DOM Polyfill is* ***amazing****, spec-compliant*
- *If we* ***only shim features Polymer needs****, we can make it much* ***lighter & faster***

The existing polyfill will live on for generic, non-Polymer web components.

Moving to webcomponents.org

Speaking of polyfills, those are getting a new home as well. At present many developers are confused about the relationship between Polymer and Web Components. Some think that you must use all of Polymer to use Web Components, when in fact, you only need the polyfills.

To make this distinction more clear, the polyfills are being moved to the webcomponents.org and they have now been renamed to `webcomponents.js`.



WebComponents.org

a place to discuss and evolve web component best-practices

POLYFILLS

The webcomponent.js polyfills enable Web Components in (evergreen) browsers that lack native support.

Install with Bower

```
bower install webcomponentsjs
```

Install with npm

```
npm install webcomponents.js
```

0.5.1 (106KB minified, 30KB gzipped)

SPECS

CUSTOM ELEMENTS

This specification describes the method for enabling the author to define and use new types of DOM elements in a document.

HTML IMPORTS

HTML Imports are a way to include and reuse HTML documents in other HTML documents.

TEMPLATES

This specification describes a method for declaring inert DOM subtrees in HTML and

ARTICLES

INTRODUCTION TO HTML IMPORTS

Template, Shadow DOM, and Custom Elements enable you to build UI components easier than before. But it's not efficient to load each resources such as HTML, CSS and JavaScript separately.

Deduping dependencies isn't easy either. To load a library like jQuery UI or Bootstrap today requires using separate tags for JavaScript, CSS, and Web Fonts. Things get

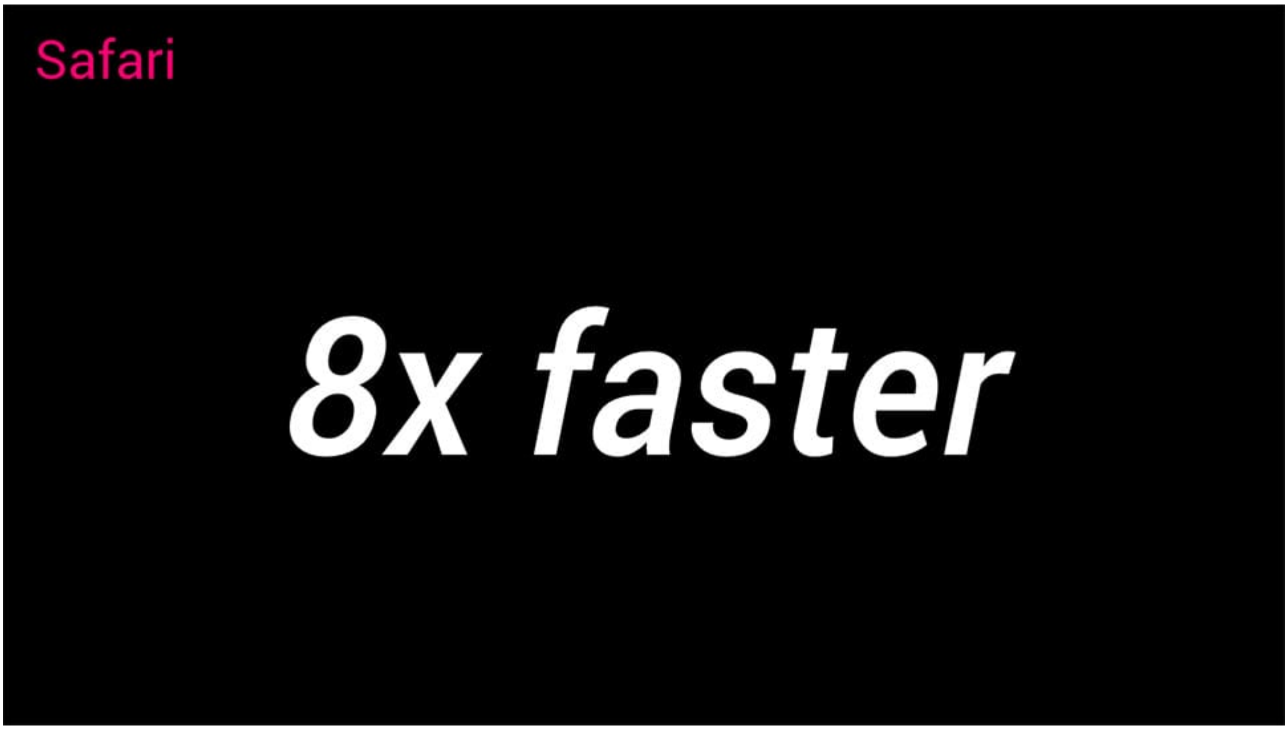
This move is designed to help other library authors take advantage of the polyfills without any confusion. The Polymer team will continue to contribute to the polyfills, but the hope is that this change makes them into more of a shared resource for the community.

Results

So what are the results from all these changes?

Speed

On Chrome, Polymer is now **5x faster**, and on Safari there has been an **8x speed up**.

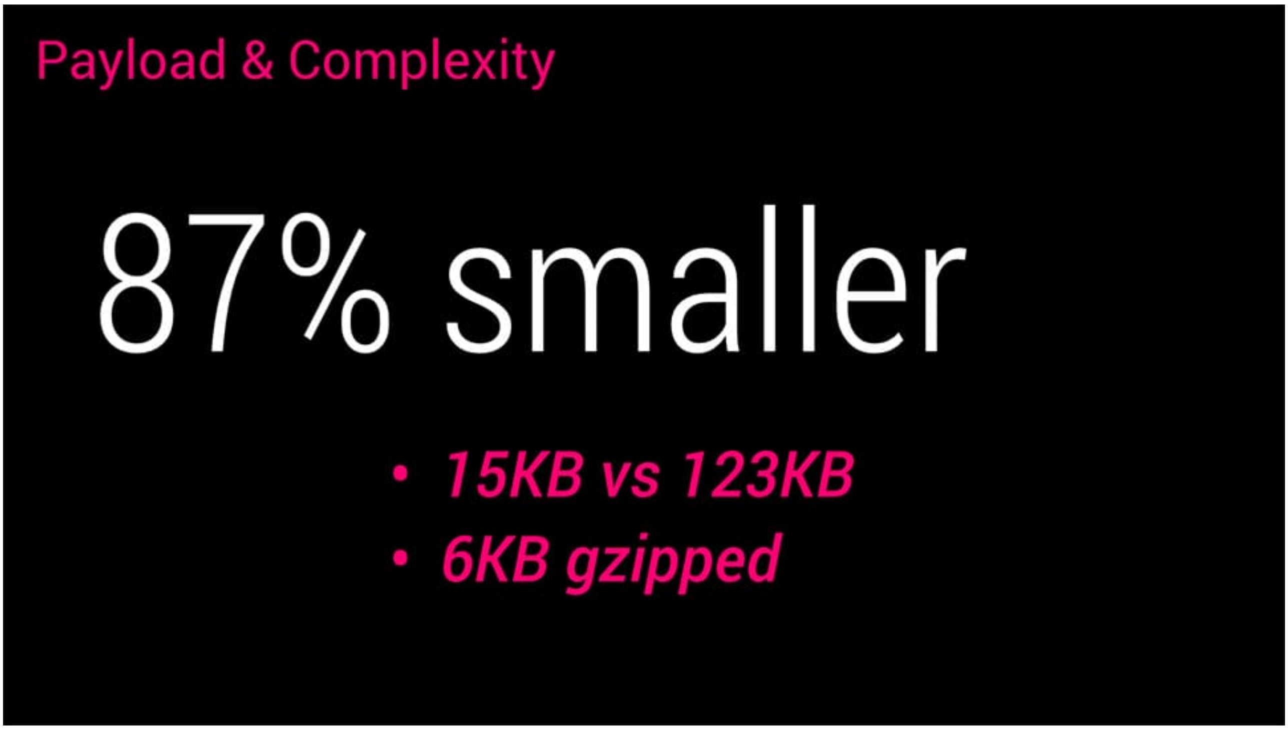


Safari

8x faster

File Size

The file size has also been reduced by 87%, from 123KB down to 15KB (6KB gzipped).



Payload & Complexity

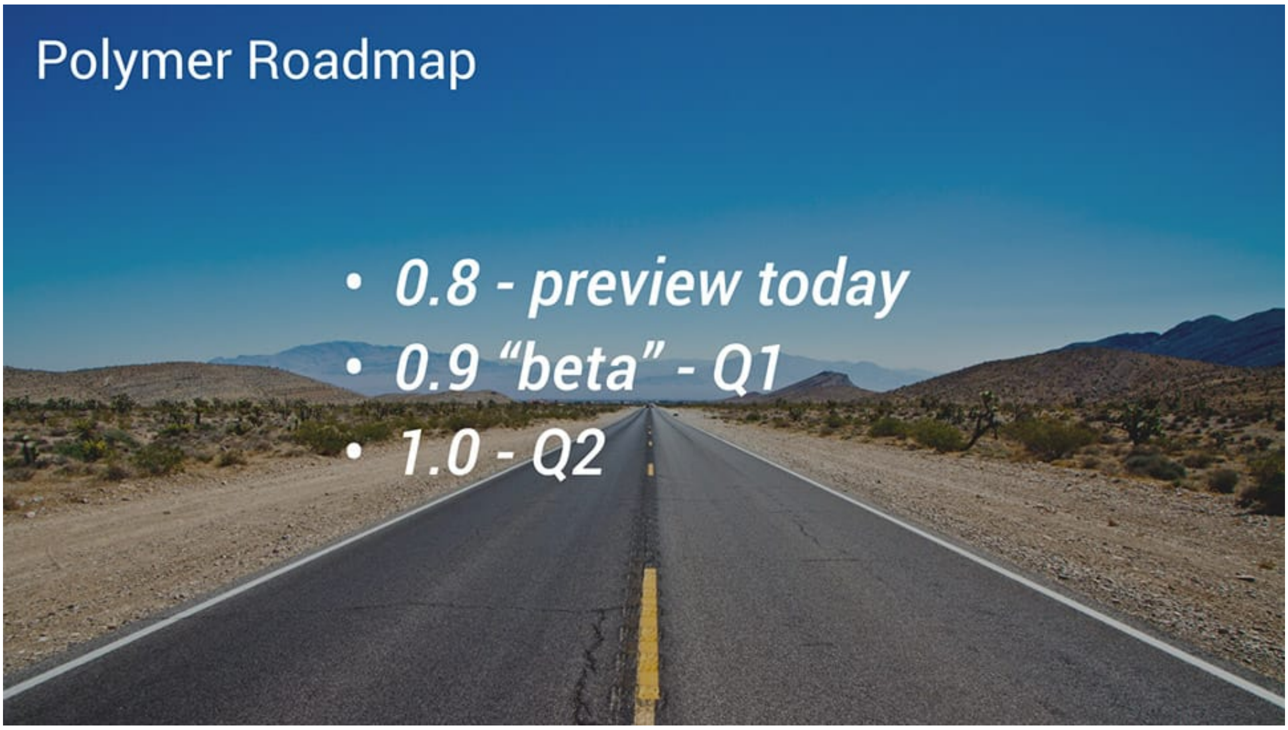
87% smaller

- *15KB vs 123KB*
- *6KB gzipped*

Roadmap

There will be some API breaking changes in the next release, signified by the new version number (0.8) but the team wants to make it clear that this is not a rewrite. Moving your current project from Polymer 0.5 to 0.8 should be fairly trivial.

The Polymer team has also outlined a roadmap to give developers more clarity around upcoming releases.



Polymer Roadmap

- 0.8 - preview today
- 0.9 “beta” - Q1
- 1.0 - Q2

The 0.8 preview is available now as a branch on GitHub (though it's still being very actively developed and is lacking in documentation). The 0.9 official beta is planned for Q1 2015, with 1.0 happening sometime in Q2.

The Experiment is Over

With all the recent changes in Polymer, the team behind it is laying the groundwork for Web Components to become an integral part of every developer's stack. If you're new to Web Components, now is a great time to take a look and familiarize yourself with these transformative technologies. If you're already working with components (and Polymer), the future looks bright indeed. Keep an eye on the Polymer Blog for all the latest updates, and sign up for the Polymer mailing list for questions or comments. Happy hacking!

Last updated July 2, 2018.