Web Components

The Four Standards, Vanilla Components, and
the Polymer Project

Ryan B. Harvey
Data Engineer @ **TED** Conferences

Thursday, October 27, 2016

Who am I?

20 years of building software through all areas of the stack

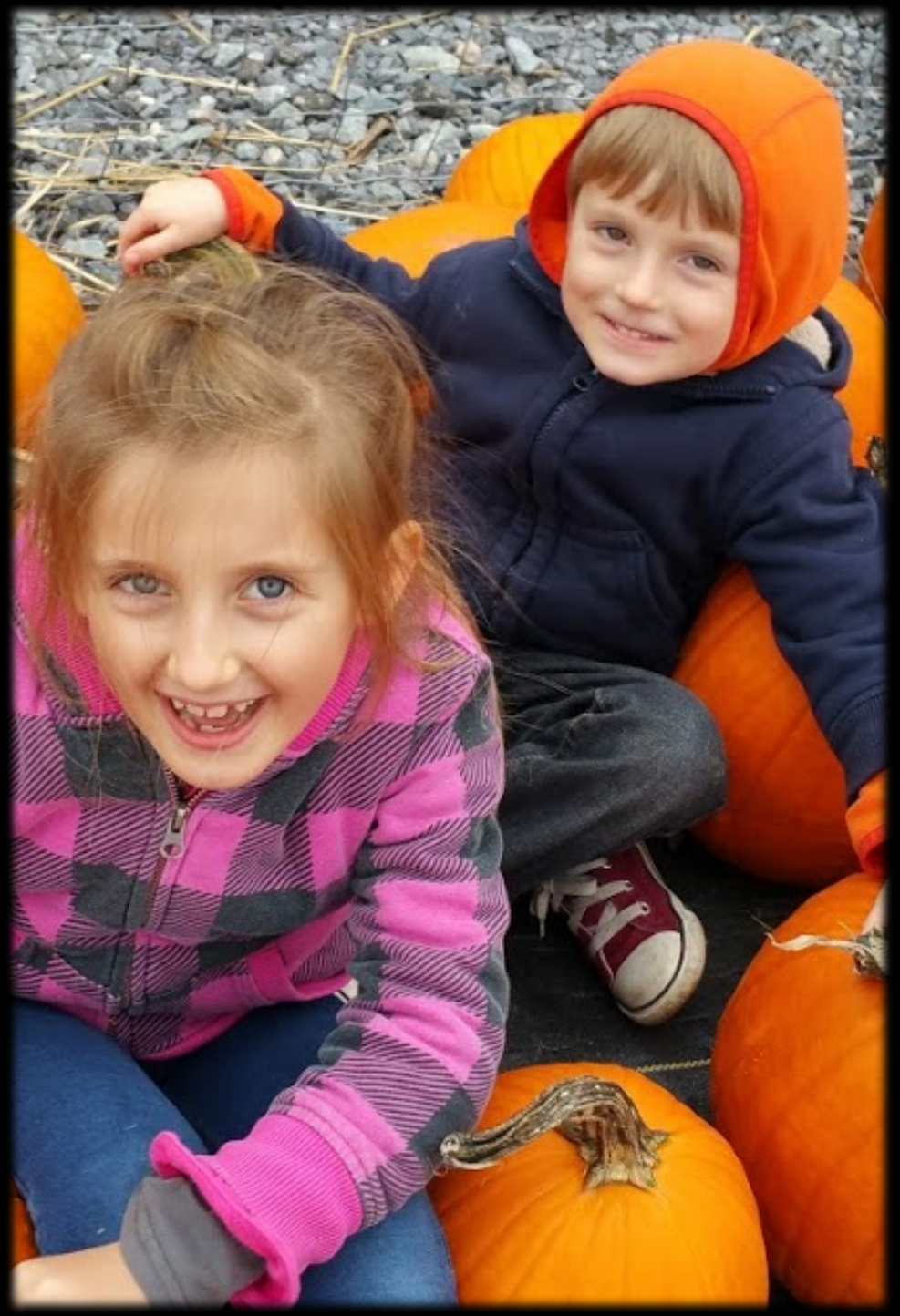
Tech lead & manager for a few years

Graduate work in computational maths

Tinkerer & civic hacker

Husband & father

Newly returned to NOLA area & working fully remote



Web Components?

- First stop: webcomponents.org (live since 2013)
- Four standards:
  - Custom Elements
  - HTML Imports
  - Templates
  - Shadow DOM

Polyfills to the rescue!

- All your browsers are belong to webcomponents.js
  (which you can get from webcomponents.org)

- This means we can use this stuff now!
  In production!

What are the specs?

- Custom Elements (v1, https://goo.gl/XbR7ue)

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

- HTML Imports (WD, https://goo.gl/CQkfV6)

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

- Templates (v1, https://goo.gl/vzBJfl)

  This specification describes a method for declaring inert DOM subtrees in HTML and manipulating them to instantiate document fragments with identical contents.

- Shadow DOM (v1, https://goo.gl/necJRu)

  This specification describes a method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM.

Shadow DOM

§ **4.2.2. Shadow tree**

A **shadow tree** is a node tree whose root is a shadow root.

A shadow root is always attached to another node tree through its host. A shadow tree is therefore never alone. The node tree of a shadow root's host is sometimes referred to as the **light tree**.

Note *A shadow tree's corresponding light tree can be a shadow tree itself.*

An element is **connected** if its shadow-including root is a document.

Essentially, an encapsulated DOM tree within a document tree. Can include any DOM elements, including <style> and <script> tags.

Testing support:

```
const supportsShadowDOMv0 = !!HTMLElement.prototype.createShadowRoot;
const supportsShadowDOMv1 = !!HTMLElement.prototype.attachShadow;
```

The <template> element

The `template` element is used to declare fragments of HTML that can be cloned and inserted in the document by script.

In a rendering, the `template` element represents nothing.

The template contents of a `template` element are not children of the element itself. Instead, they are stored in a `DocumentFragment` associated with a different `Document` without a browsing context so as to avoid the `template` contents interfering with the main `Document`. (For example, this avoids form controls from being submitted, scripts from executing, and so forth.) The template contents have no conformance requirements.

A container for a detached DOM subtree (document fragment) that does not get rendered unless/until adopted by another DOM node (usually a custom element).

Testing support:

```
const supportsTemplateV0 = 'content' in document.createElement(‘template');
const supportsTemplateV1 = 'slot' in document.createElement('template');
```

`<template>`, continued

Templates can contain `<slot>` elements, which allow you to include light DOM content within the shadow DOM you are defining.

These look like:

```
<slot></slot>
<slot>
  Default content
  <span>possibly including DOM</span>
</slot>
<slot name="card-title">Default title</slot>
```

Custom Elements

```
class AppDrawer extends HTMLElement {...}
window.customElements.define('app-drawer', AppDrawer);

// Or use an anonymous class if you don't want a named constructor in current scope.
window.customElements.define('app-drawer', class extends HTMLElement {...});
```

Register a new element DOM object.

- Must have ‘–‘ (hyphen) in the tag name.
- You can’t define the same tag more than once; browsers will throw a DOMException on a second definition of the same tag.
- Must not be self-closing — always use with a closing tag.
- If you define a `constructor`, it must call `super()` first.
- Elements must have `HTMLElement` as an ancestor, either extending it or another component that extends it.

HTML Imports

A new option for the `<link>` tag that supports importing an HTML document into the current DOM tree.

- The browser's network stack de-dupes requests from the same full URL, so an import used in multiple requests is loaded only once.
- Scripts defined in an import executes in the scope of the importing document, not the import.
- Scripts do not block the parsing of the main page, but execute in order specified within the import (like `defer`, for free!).

What it looks like:

```
<link rel='import' href='path/to/import.html'></link>
```

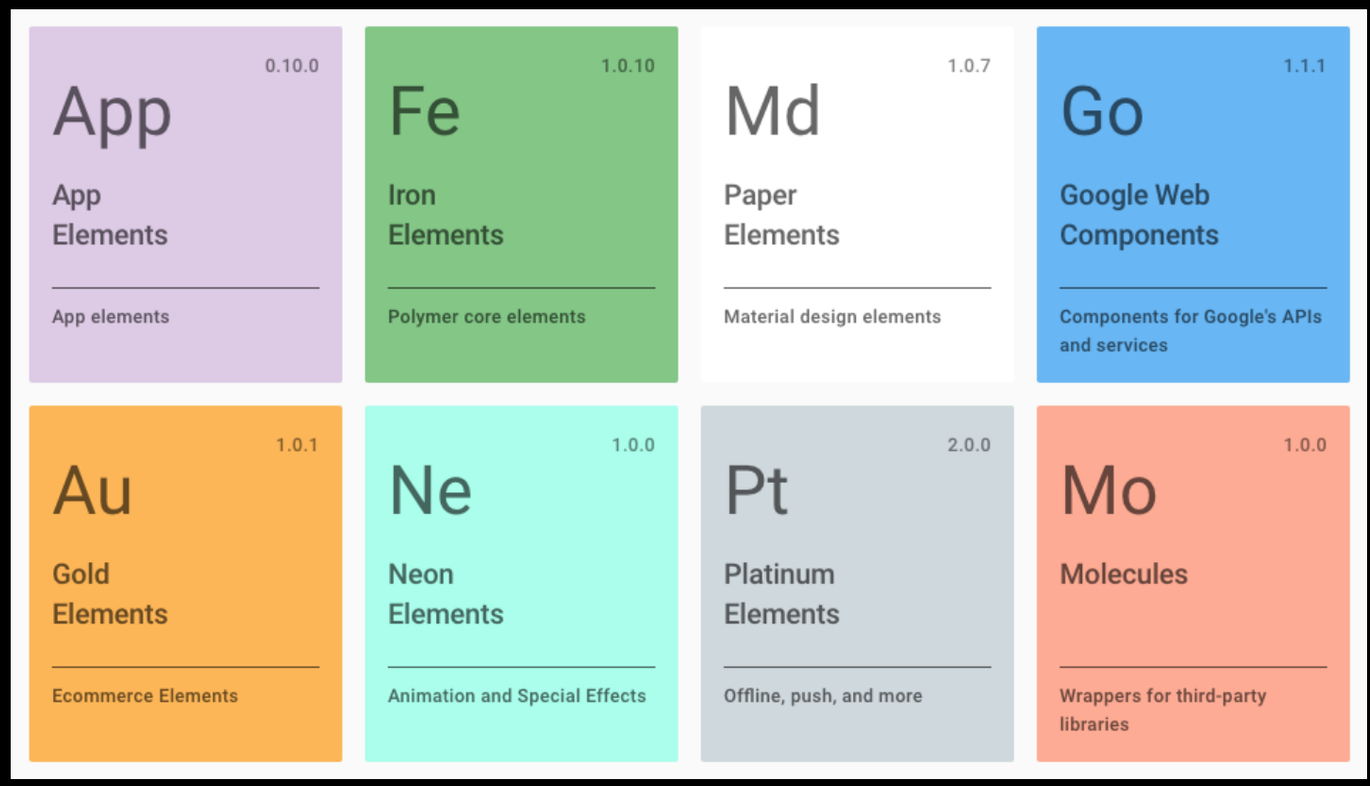
Testing for support:

```
const supportsHTMLImports = 'import' in document.createElement('link');
```

Polymer Project!

Google's Polymer Project is a massive effort to move web components forward with minimal JavaScript library support.

- Framework for easier creation of web components, and catalog of many hundreds of ready-to-import components. Current version 1.7.
- Used in production by several big companies in big and small ways: Google, USA Today, EA, Bloomberg, Comcast, GitHub.
- The catalog contains several element categories.
- Polymer Project v2.0 preview is available today!



Demo time!

To the code!!!

https://github.com/nihonjinrxs/frontendparty-oct2016

Ryan B. Harvey

http://datascientist.guru

ryan.b.harvey@gmail.com

@CodeAndData

Data Engineer
TED Conferences

Data Scientist & Software Architect
KITCHOLOGY

Previously
Sr. IT Project Manager @ White House OMB
Research Affiliate @ UMCP
Board Member @ Data Community DC
IT Fellow/Computer Scientist @ SSA
DSP Analyst & Engineer @ BAE Systems

Education
Executive Data Science Spec., JHU/Coursera
M.S. Applied Mathematics, UMCP
Cert. Innovation Management, UMCP
Cert. Computational Harmonic Analysis, UMCP
Cert. Scientific Computing, UMCP
B.S. Maths & Comp. Sci., Loyola U New Orleans