Progressive Web Apps & Polymer

Wendy Ginsberg
Product Manager, Google
@wmginsberg

Who am I?

Wendy Ginsberg
Chrome Web Platform
Google

Who are you?

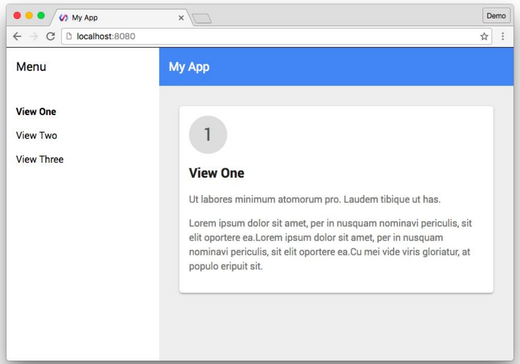
??????
????????????
???????

1. Polymer
2. PWAs
3. Polymer + PWAs

1. Polymer

Polymer == “An **opinionated library** that **sugars** Web Components APIs, to make it easy to build encapsulated, reusable custom elements.”

Polymer library



My App

View One

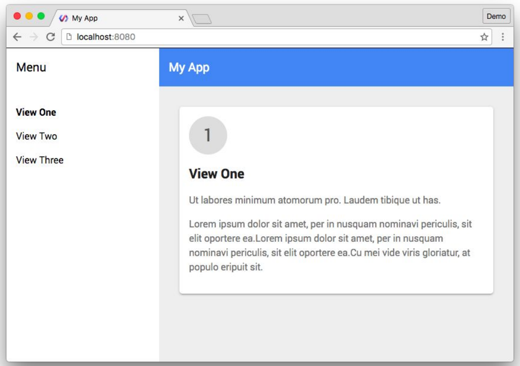
Ut labores minimum atomorum pro. Laudem tibique ut has.

Lorem ipsum dolor sit amet, per in nusquam nominavi periculis, sit elit oportere ea.Lorem ipsum dolor sit amet, per in nusquam nominavi periculis, sit elit oportere ea.Cu mei vide viris gloriatur, at populo eripuit sit.

App Toolbox

webcomponents.org

Polymer library



Menu

View One

View Two

View Three

My App

1

View One

Ut labores minimum atomorum pro. Laudem tibique ut has.

Lorem ipsum dolor sit amet, per in nusquam nominavi periculis, sit elit oportere ea.Lorem ipsum dolor sit amet, per in nusquam nominavi periculis, sit elit oportere ea.Cu mei vide viris gloriatur, at populo eripuit sit.

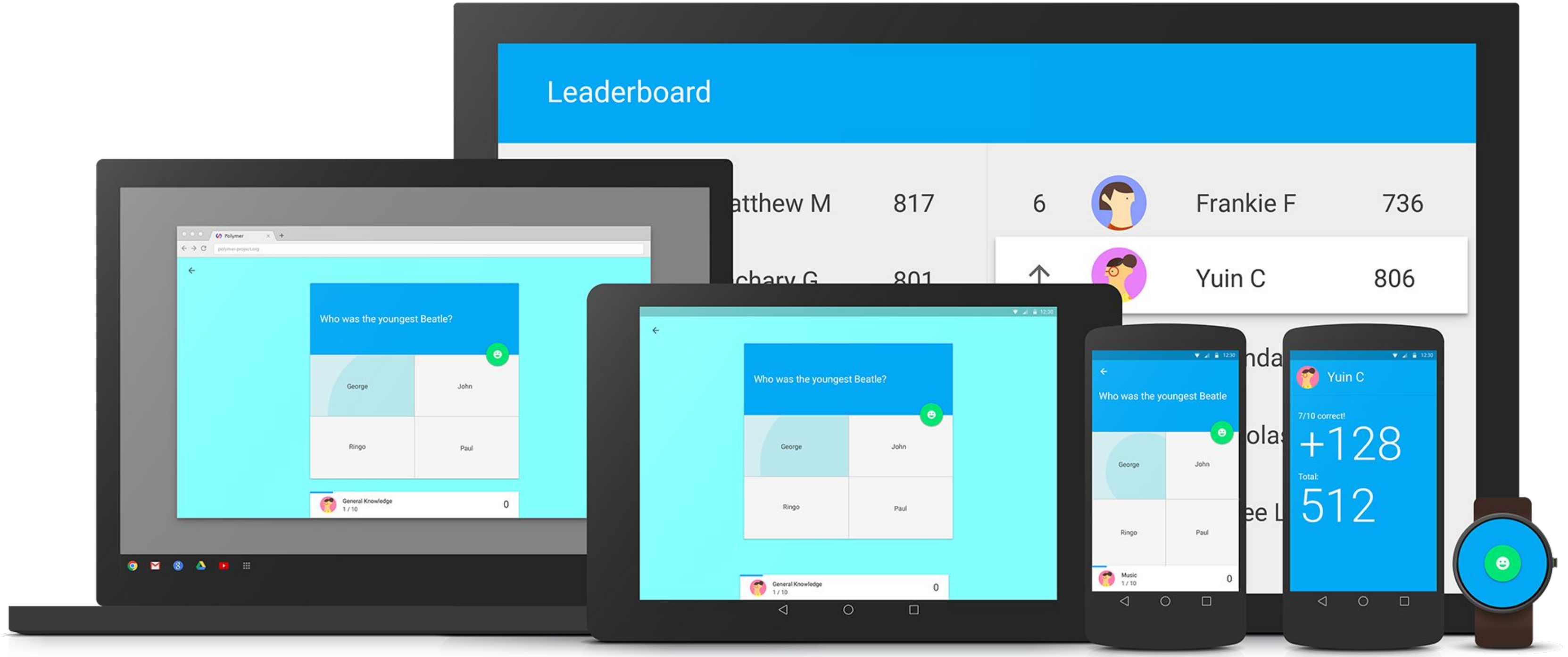
App Toolbox

webcomponents.org

Polymer CLI

Web Component Tester

What problems are we solving?

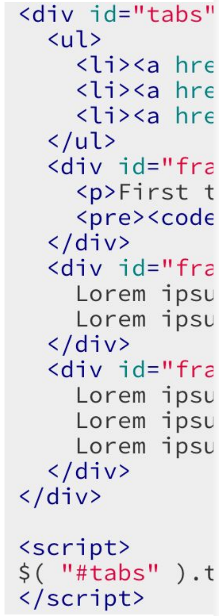

Leaderboard
atthew M 817
6 Frankie F 736
chary G 801
Yuin C 806
Who was the youngest Beatle?
George
John
Ringo
Paul
General Knowledge
1 / 10
0
Music
Yuin C
7/10 correct!
+128
Total:
512

```
<h1>
<ul>
<p>
```

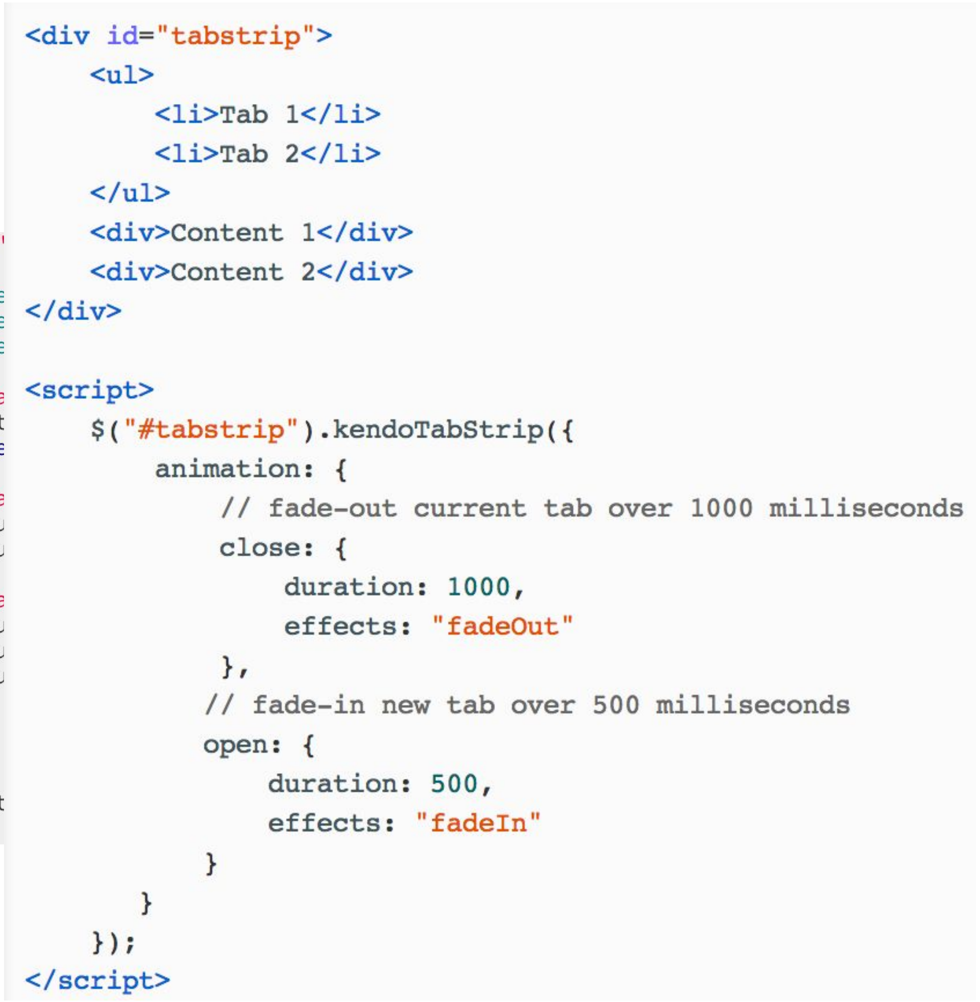
Building UI tabs should be easy!

http://drbl.in/esYL

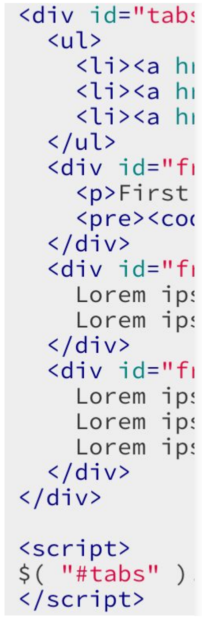
```
<div id="tabs">
  <ul>
    <li><a href="#fragment-1"><span>One</span></a></li>
    <li><a href="#fragment-2"><span>Two</span></a></li>
    <li><a href="#fragment-3"><span>Three</span></a></li>
  </ul>
  <div id="fragment-1">
    <p>First tab is active by default:</p>
    <pre><code>$( "#tabs" ).tabs(); </code></pre>
  </div>
  <div id="fragment-2">
    Lorem ipsum dolor sit amet, consectetuer adipiscing eli
    Lorem ipsum dolor sit amet, consectetuer adipiscing eli
  </div>
  <div id="fragment-3">
    Lorem ipsum dolor sit amet, consectetuer adipiscing eli
    Lorem ipsum dolor sit amet, consectetuer adipiscing eli
    Lorem ipsum dolor sit amet, consectetuer adipiscing eli
  </div>
</div>

<script>
$( "#tabs" ).tabs();
</script>
```



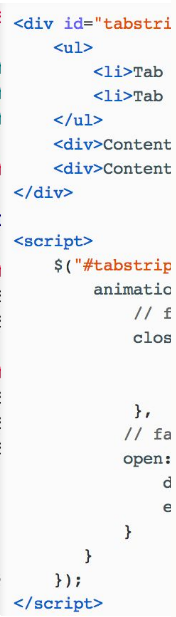
```
<div id="tabs"
  <ul>
    <li><a hre
    <li><a hre
    <li><a hre
  </ul>
  <div id="fra
    <p>First t
    <pre><code
  </div>
  <div id="fra
    Lorem ipsu
    Lorem ipsu
  </div>
  <div id="fra
    Lorem ipsu
    Lorem ipsu
    Lorem ipsu
  </div>
</div>

<script>
$( "#tabs" ).t
</script>
```



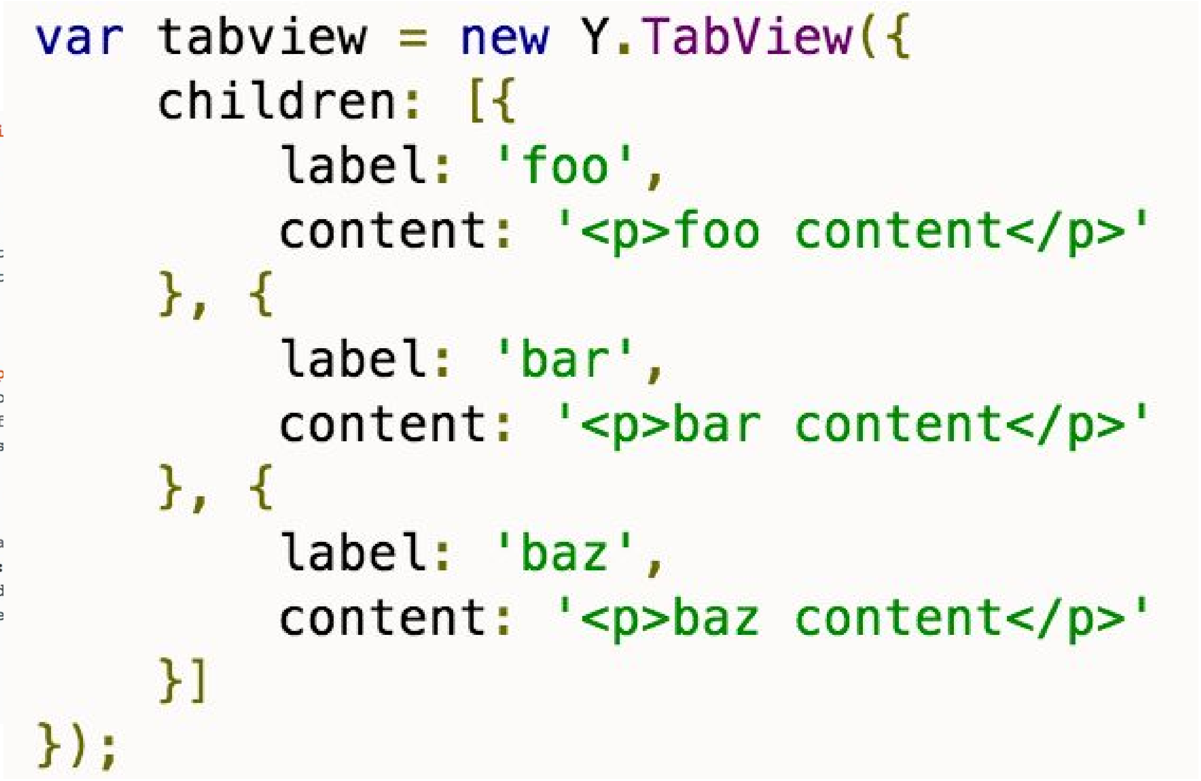
```
<div id="tabstrip">
    <ul>
        <li>Tab 1</li>
        <li>Tab 2</li>
    </ul>
    <div>Content 1</div>
    <div>Content 2</div>
</div>

<script>
    $("#tabstrip").kendoTabStrip({
        animation: {
            // fade-out current tab over 1000 milliseconds
            close: {
                duration: 1000,
                effects: "fadeOut"
            },
           // fade-in new tab over 500 milliseconds
           open: {
               duration: 500,
               effects: "fadeIn"
           }
        }
    });
</script>
```



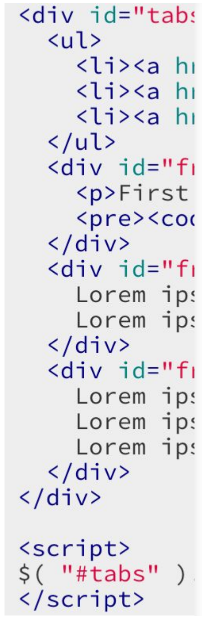
```
<div id="tabs
  <ul>
    <li><a h
    <li><a h
    <li><a h
  </ul>
  <div id="f
    <p>First
    <pre><co
  </div>
  <div id="f
    Lorem ip
    Lorem ip
  </div>
  <div id="f
    Lorem ip
    Lorem ip
    Lorem ip
  </div>
</div>

<script>
$( "#tabs" )
</script>
```



```
<div id="tabstri
    <ul>
        <li>Tab
        <li>Tab
    </ul>
    <div>Content
    <div>Content
</div>

<script>
    $("#tabstrip
        animatio
            // f
            clos


            },
            // fa
            open:
                d
                e
            }
        }
    });
</script>
```



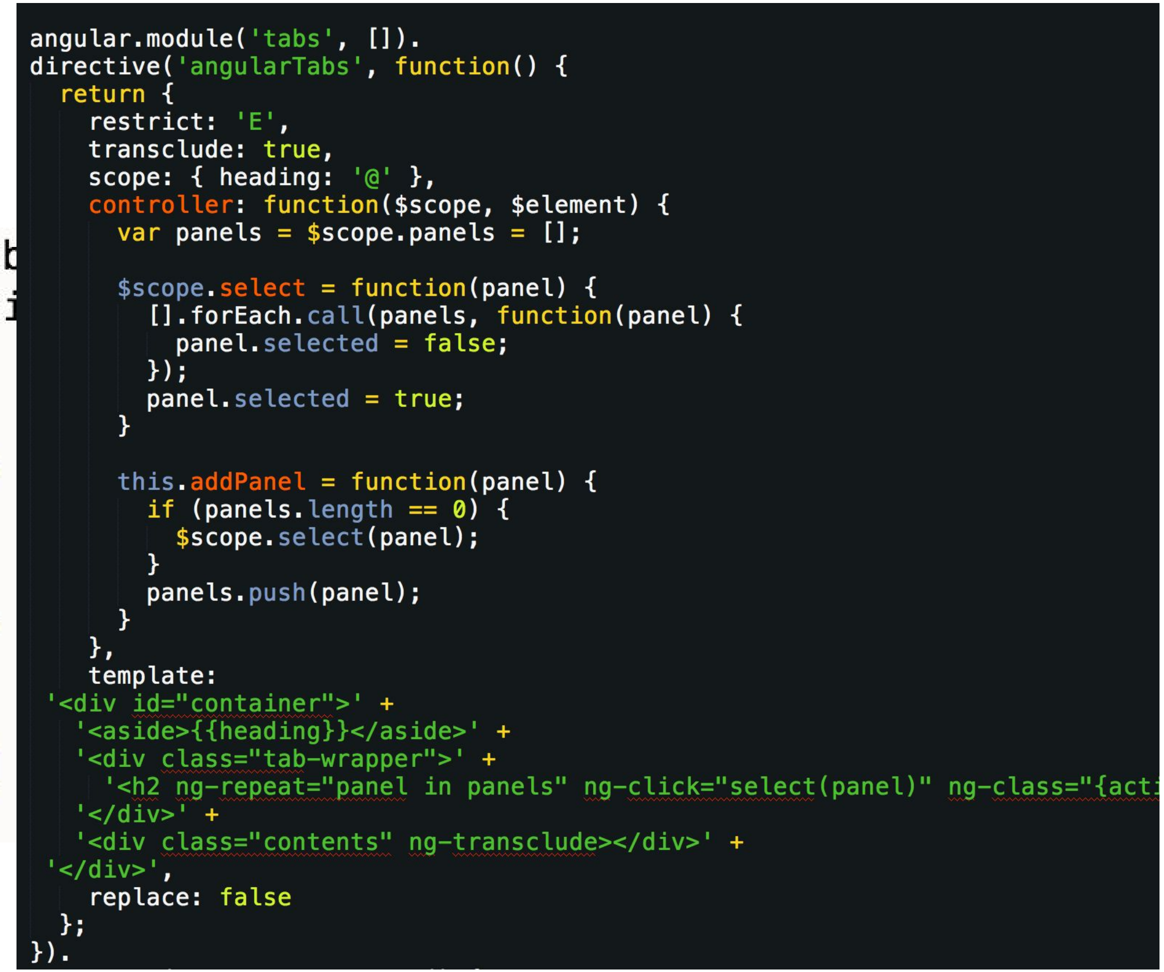
```
var tabview = new Y.TabView({
    children: [{
        label: 'foo',
        content: '<p>foo content</p>'
    }, {
        label: 'bar',
        content: '<p>bar content</p>'
    }, {
        label: 'baz',
        content: '<p>baz content</p>'
    }]
});
```



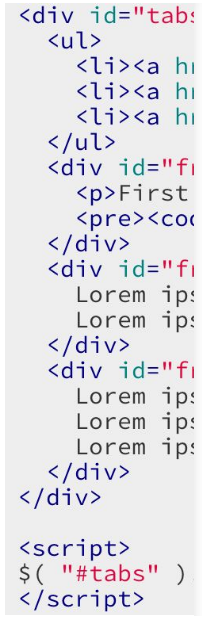
```
<div id="tabs
  <ul>
    <li><a hı
    <li><a hı
    <li><a hı
  </ul>
  <div id="fı
    <p>First
    <pre><coı
  </div>
  <div id="fı
    Lorem ips
    Lorem ips
  </div>
  <div id="fı
    Lorem ips
    Lorem ips
    Lorem ips
  </div>
</div>

<script>
$( "#tabs" )
</script>
```

```
<div id="tabstri
    <ul>
        <li>Tab
        <li>Tab
    </ul>
    <div>Content
    <div>Content
</div>

<script>
    $("#tabstrip
        animatio
            // f
            clos


            },
           // fa
           open:
               d
               e
           }
       }
   });
</script>
```

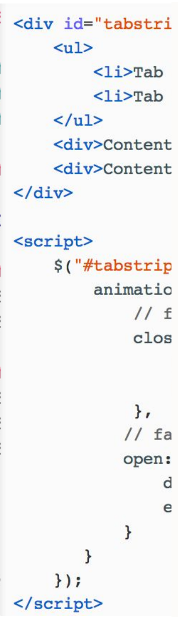
```
var tab
    chi



    },




    },



    }]
});
```



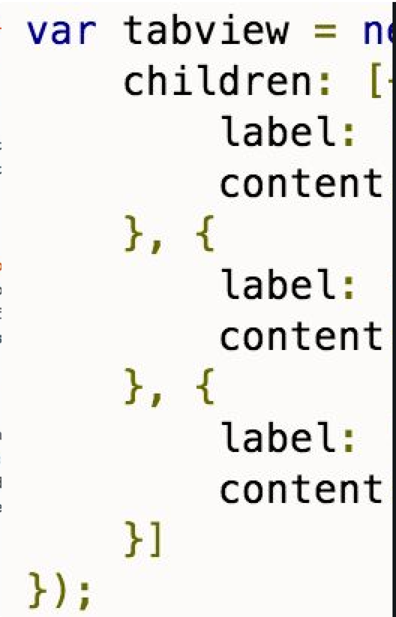
```
angular.module('tabs', []).
directive('angularTabs', function() {
  return {
    restrict: 'E',
    transclude: true,
    scope: { heading: '@' },
    controller: function($scope, $element) {
      var panels = $scope.panels = [];

      $scope.select = function(panel) {
        [].forEach.call(panels, function(panel) {
          panel.selected = false;
        });
        panel.selected = true;
      }

      this.addPanel = function(panel) {
        if (panels.length == 0) {
          $scope.select(panel);
        }
        panels.push(panel);
      }
    },
    template:
   '<div id="container">' +
     '<aside>{{heading}}</aside>' +
     '<div class="tab-wrapper">' +
       '<h2 ng-repeat="panel in panels" ng-click="select(panel)" ng-class="{acti
     '</div>' +
     '<div class="contents" ng-transclude></div>' +
   '</div>',
    replace: false
  };
}).
```



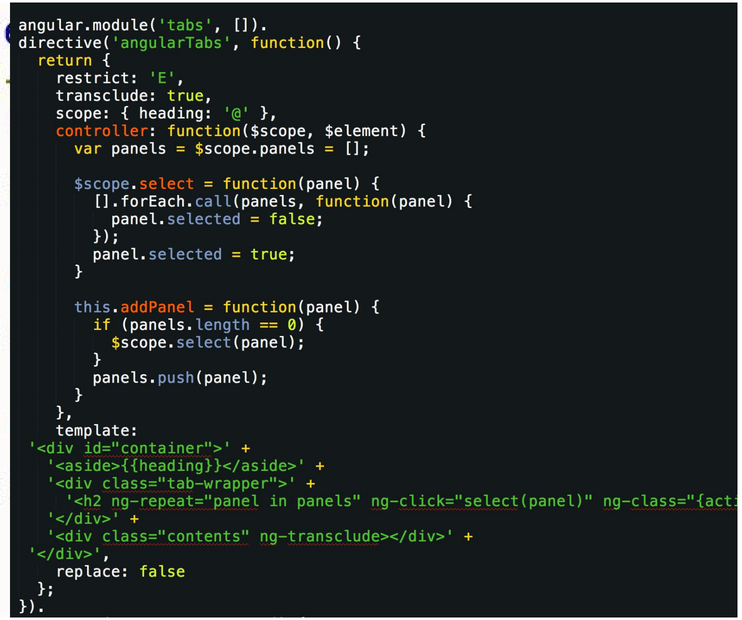
```
<div id="tabs
  <ul>
    <li><a hı
    <li><a hı
    <li><a hı
  </ul>
  <div id="fı
    <p>First
    <pre><coc
  </div>
  <div id="fı
    Lorem ips
    Lorem ips
  </div>
  <div id="fı
    Lorem ips
    Lorem ips
    Lorem ips
  </div>
</div>

<script>
$( "#tabs" )
</script>
```



```
<div id="tabstri
    <ul>
        <li>Tab
        <li>Tab
    </ul>
    <div>Content
    <div>Content
</div>

<script>
    $("#tabstrip
        animatio
            // f
            clos


            },
           // fa
           open:
               d
               e
           }
       }
    });
</script>
```



```
var tabview = ne
    children: [
        label:
        content
    }, {
        label:
        content
    }, {
        label:
        content
    }]
});
```



```
angular.module('tabs', []).
directive('angularTabs', function() {
  return {
    restrict: 'E',
    transclude: true,
    scope: { heading: '@' },
    controller: function($scope, $element) {
      var panels = $scope.panels = [];

      $scope.select = function(panel) {
        [].forEach.call(panels, function(panel) {
          panel.selected = false;
        });
        panel.selected = true;
      }

      this.addPanel = function(panel) {
        if (panels.length == 0) {
          $scope.select(panel);
        }
        panels.push(panel);
      }
    },
    template:
   '<div id="container">' +
     '<aside>{{heading}}</aside>' +
     '<div class="tab-wrapper">' +
       '<h2 ng-repeat="panel in panels" ng-click="select(panel)" ng-class="{acti
     '</div>' +
     '<div class="contents" ng-transclude></div>' +
   '</div>',
    replace: false
  };
}).
```

NOT INTEROPERABLE

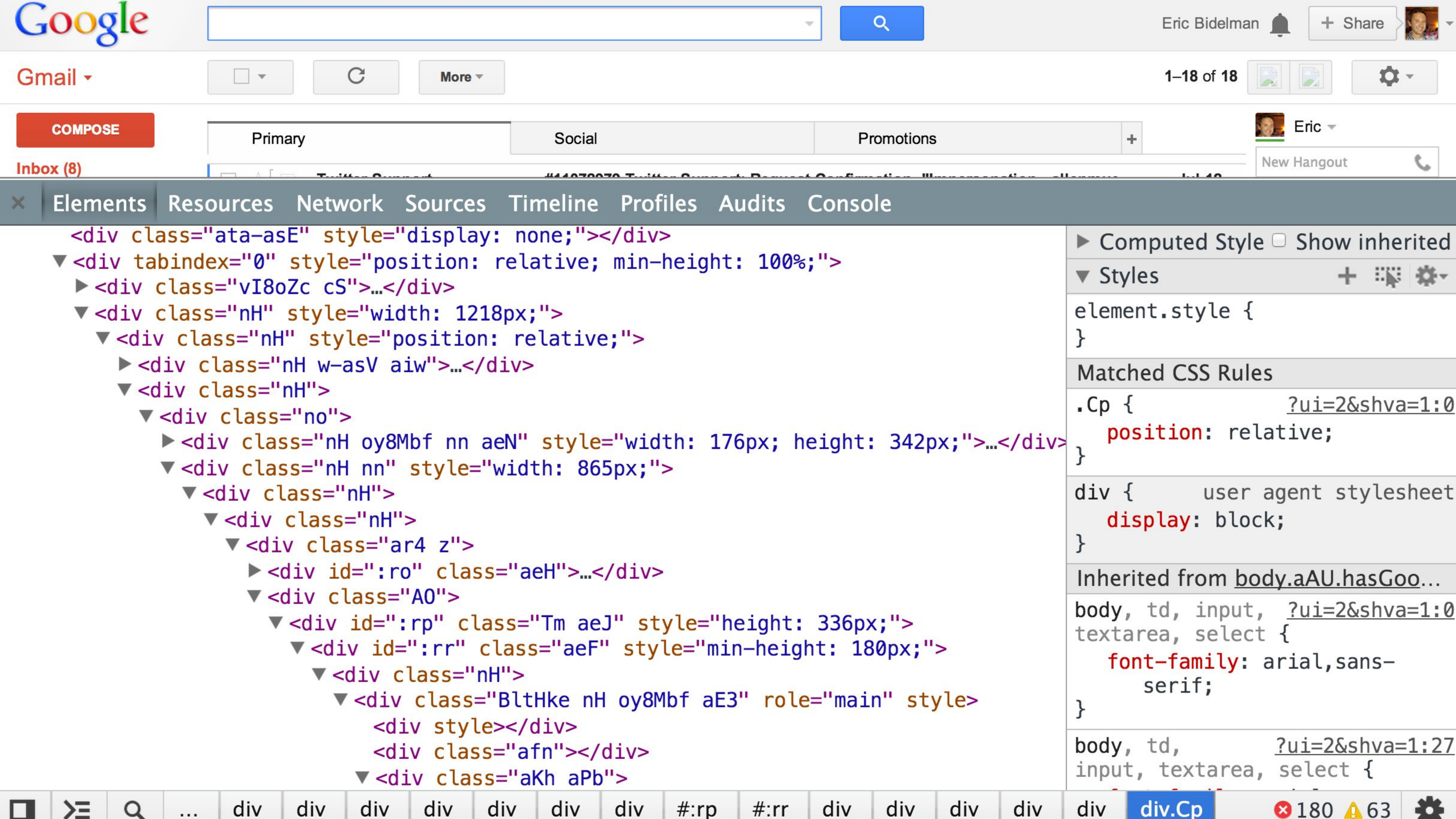
```
<div id="tabs
  <ul>
    <li><a hr
    <li><a hr
    <li><a hr
  </ul>
  <div id="fr
    <p>First
    <pre><coc
  </div>
  <div id="fr
    Lorem ips
    Lorem ips
  </div>
  <div id="fr
    Lorem ips
    Lorem ips
    Lorem ips
  </div>
</div>

<script>
$( "#tabs" )
</script>
```

```
<div id="tabstri
    <ul>
        <li>Tab
        <li>Tab
    </ul>
    <div>Content
    <div>Content
</div>

<script>
    $("#tabstrip
            // f
            clos
            },
          // fa
          open:
              d
              e
          }
      }
    });
</script>
```

```
var tabview = ne
    children: [{
        label:
        content
    }, {
        label:
        content
    }, {
        label:
        content
    }]
});
```

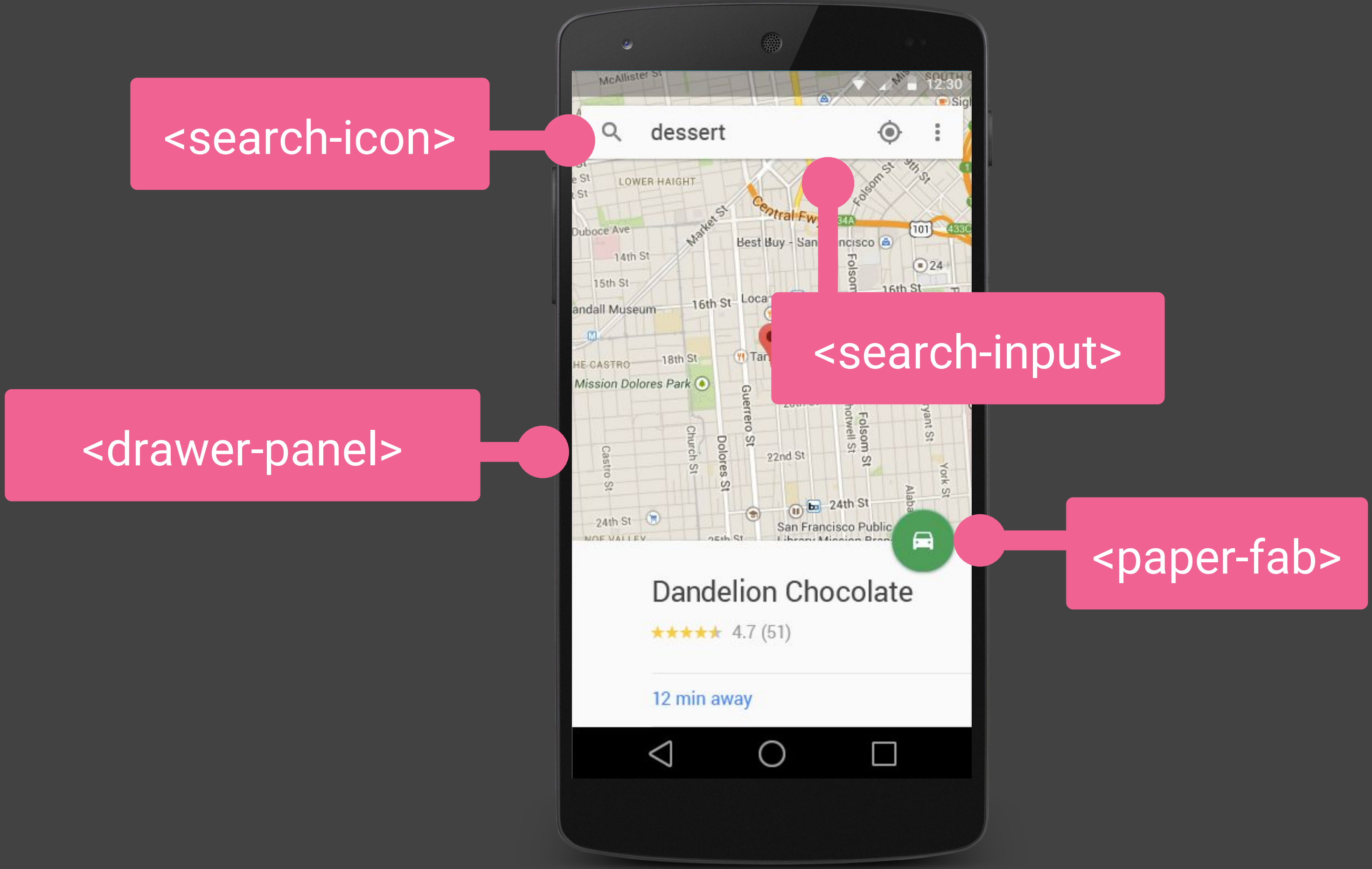
```
angular.module('tabs', []).
directive('angularTabs', function() {
  return {
    restrict: 'E',
    transclude: true,
    scope: { heading: '@' },
    controller: function($scope, $element) {
      var panels = $scope.panels = [];

      $scope.select = function(panel) {
        [].forEach.call(panels, function(panel) {
          panel.selected = false;
        });
        panel.selected = true;
      }

      this.addPanel = function(panel) {
        if (panels.length == 0) {
          $scope.select(panel);
        }
        panels.push(panel);
      }
    },
    template:
  '<div id="container">' +
    '<aside>{{heading}}</aside>' +
    '<div class="tab-wrapper">' +
      '<h2 ng-repeat="panel in panels" ng-click="select(panel)" ng-class="{acti
    '</div>' +
    '<div class="contents" ng-transclude></div>' +
  '</div>',
    replace: false
  };
}).
```

How can we redesign HTML for the (modern mobile) web?



```
    <div class="ata-asE" style="display: none;"></div>
  ▼<div tabindex="0" style="position: relative; min-height: 100%;">
    ▶<div class="vI8oZc cS">…</div>
    ▼<div class="nH" style="width: 1218px;">
      ▼<div class="nH" style="position: relative;">
        ▶<div class="nH w-asV aiw">…</div>
        ▼<div class="nH">
          ▼<div class="no">
            ▶<div class="nH oy8Mbf nn aeN" style="width: 176px; height: 342px;">…</div>
            ▼<div class="nH nn" style="width: 865px;">
              ▼<div class="nH">
                ▼<div class="nH">
                  ▼<div class="ar4 z">
                    ▶<div id=":ro" class="aeH">…</div>
                    ▼<div class="AO">
                      ▼<div id=":rp" class="Tm aeJ" style="height: 336px;">
                        ▼<div id=":rr" class="aeF" style="min-height: 180px;">
                          ▼<div class="nH">
                            ▼<div class="BltHke nH oy8Mbf aE3" role="main" style>
                                <div style></div>
                                <div class="afn"></div>
                              ▼<div class="aKh aPb">
```

```
▶ Computed Style  Show inherited
▼ Styles
element.style {
}
Matched CSS Rules
.Cp {                         ?ui=2&shva=1:0
  position: relative;
}
div {          user agent stylesheet
  display: block;
}
Inherited from body.aAU.hasGoo…
body, td, input,              ?ui=2&shva=1:0
textarea, select {
  font-family: arial,sans-
     serif;
}
body, td,                     ?ui=2&shva=1:27
input, textarea, select {
```



<search-icon>
<search-input>
<drawer-panel>
<paper-fab>
dessert
Dandelion Chocolate
4.7 (51)
12 min away

Build a better framework.

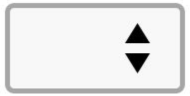
Build a better ~~framework~~ web.

That sounds hard.

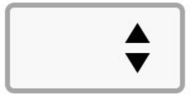
Where do we start?

The web platform is already
incredibly powerful.

A study in <select>

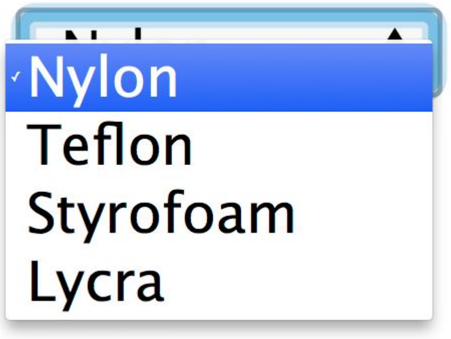


Simple



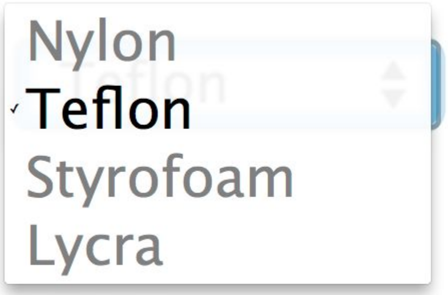
```
<select>
</select>
```

Composable



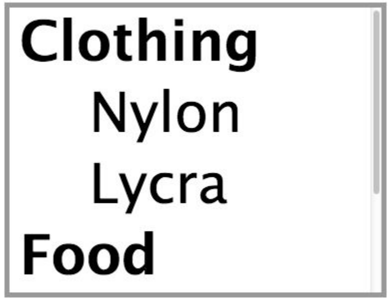
```
<select>
        <option>Nylon</option>
        <option>Teflon</option>
        <option>Styrofoam</option>
        <option>Lycra</option>
</select>
```

Declarativ e



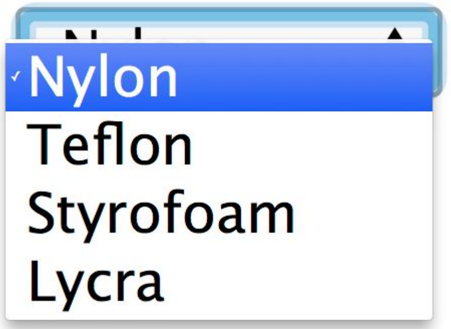
```
<select>
        <option disabled>Nylon</option>
        <option selected>Teflon</option>
        <option disabled>Styrofoam</option>
        <option disabled>Lycra</option>
</select>
```

Flexible


Clothing
Nylon
Lycra
Food

```
<select multiple>
        <optgroup label="Clothing">
                <option>Nylon</option>
                <option>Lycra</option>
        </optgroup>
        …
</select>
```

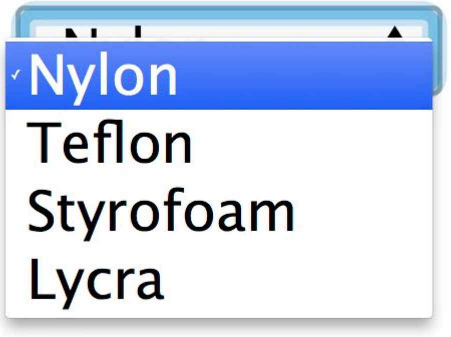
Forgiving



```
<select>
        <option>Nylon</option>
        <option>Teflon</option>
        ~~<p>Propylene?</p>~~
        <option>Styrofoam</option>
        <option>Lycra</option>
</select>
```

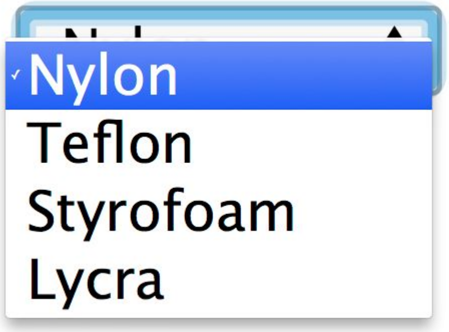
Accessible

Pick a Polymer



```
<label for="polymer-picker">Pick a Polymer</label>
<select id="polymer-picker">
        <option>Nylon</option>
        <option>Teflon</option>
        <option>Styrofoam</option>
        <option>Lycra</option>
</select>
```

Programmable



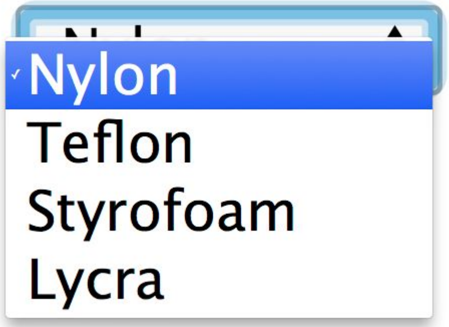
```
> this.selectedIndex;
> 0
> this.addEventListener('change', function() {
>                 console.log("I changed!");
>       });
```

<select>

an amazing little element

- ✓ **declarative**, readable
- ✓ **encapsulated** behavior
- ✓ **reusable** across contexts

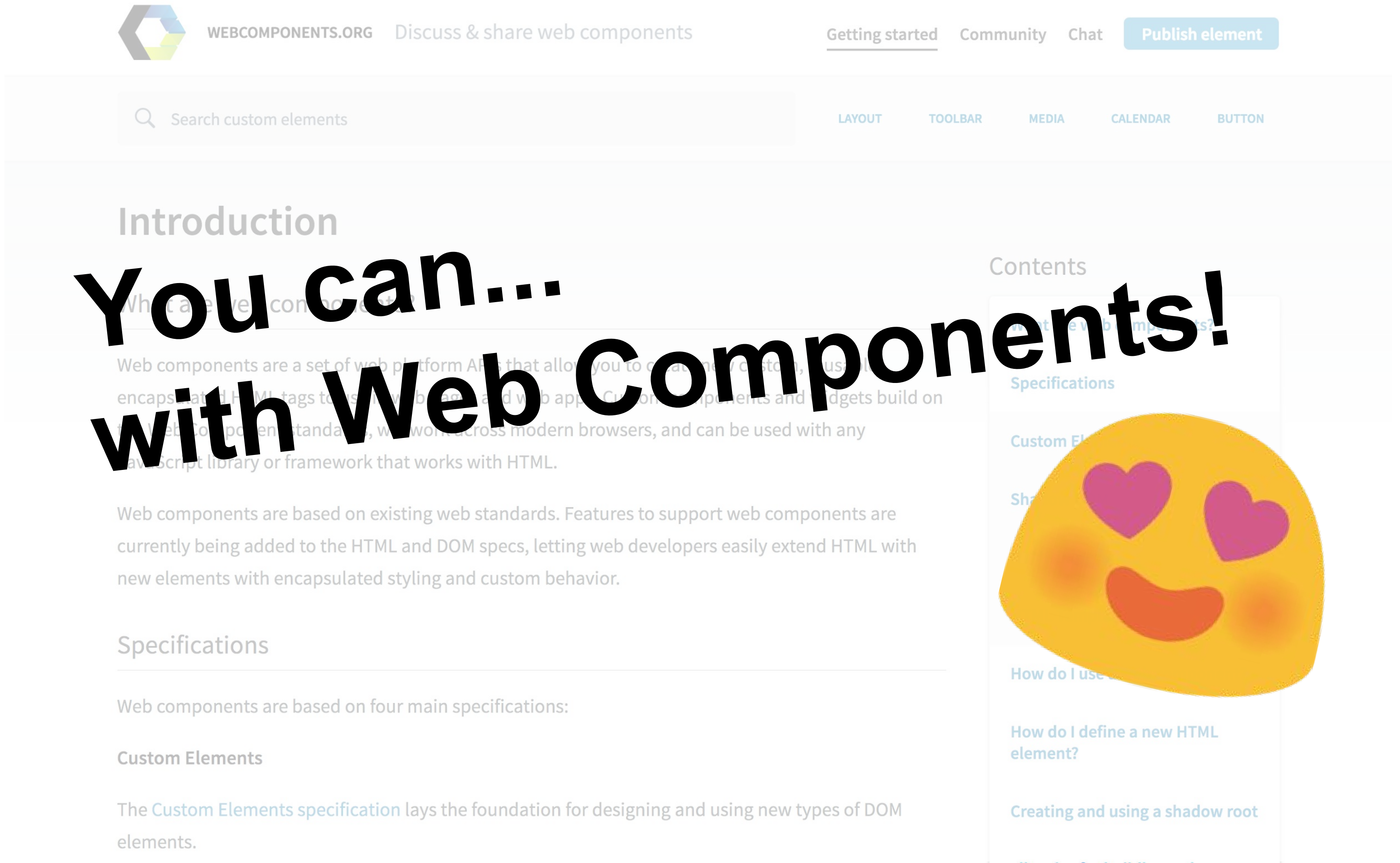
```
<select>
    <option>Nylon</option>
    <option>Teflon</option>
    <option>Styrofoam</option>
    <option>Lycra</option>
</select>
```



I want more elements like this.

I want to **build** elements like this.

You can… with Web Components!



WEBCOMPONENTS.ORG Discuss & share web components

Getting started Community Chat Publish element

Search custom elements

LAYOUT TOOLBAR MEDIA CALENDAR BUTTON

Introduction

What are web components?

Web components are a set of web platform APIs that allow you to create new custom, reusable, encapsulated HTML tags to use in web pages and web apps. Custom components and widgets build on the Web Component standards, will work across modern browsers, and can be used with any JavaScript library or framework that works with HTML.

Web components are based on existing web standards. Features to support web components are currently being added to the HTML and DOM specs, letting web developers easily extend HTML with new elements with encapsulated styling and custom behavior.

Specifications

Web components are based on four main specifications:

Custom Elements

The Custom Elements specification lays the foundation for designing and using new types of DOM elements.

Contents

What are web components?

Specifications

Custom El

Sha

How do I use

How do I define a new HTML element?

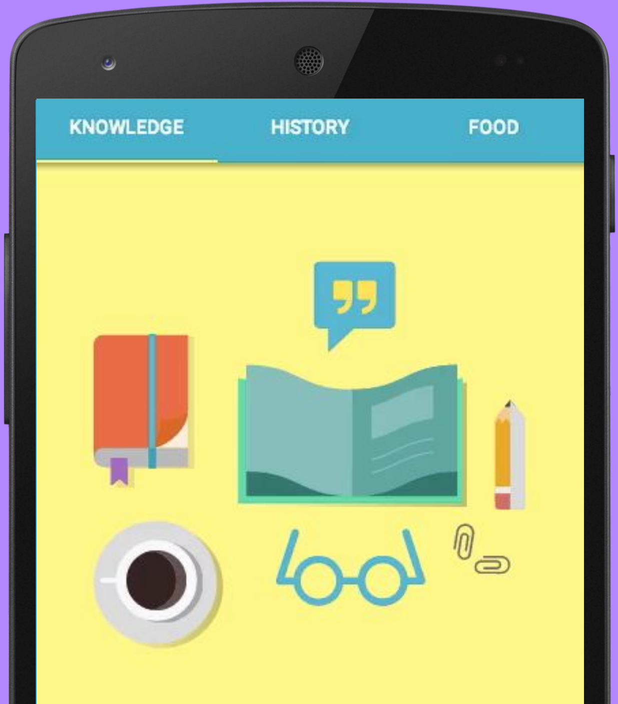
Creating and using a shadow root

Polymer == “An **opinionated library** that **sugars** Web Components APIs, to make it easy to build encapsulated, reusable custom elements.”

Web Components
+ Polymer

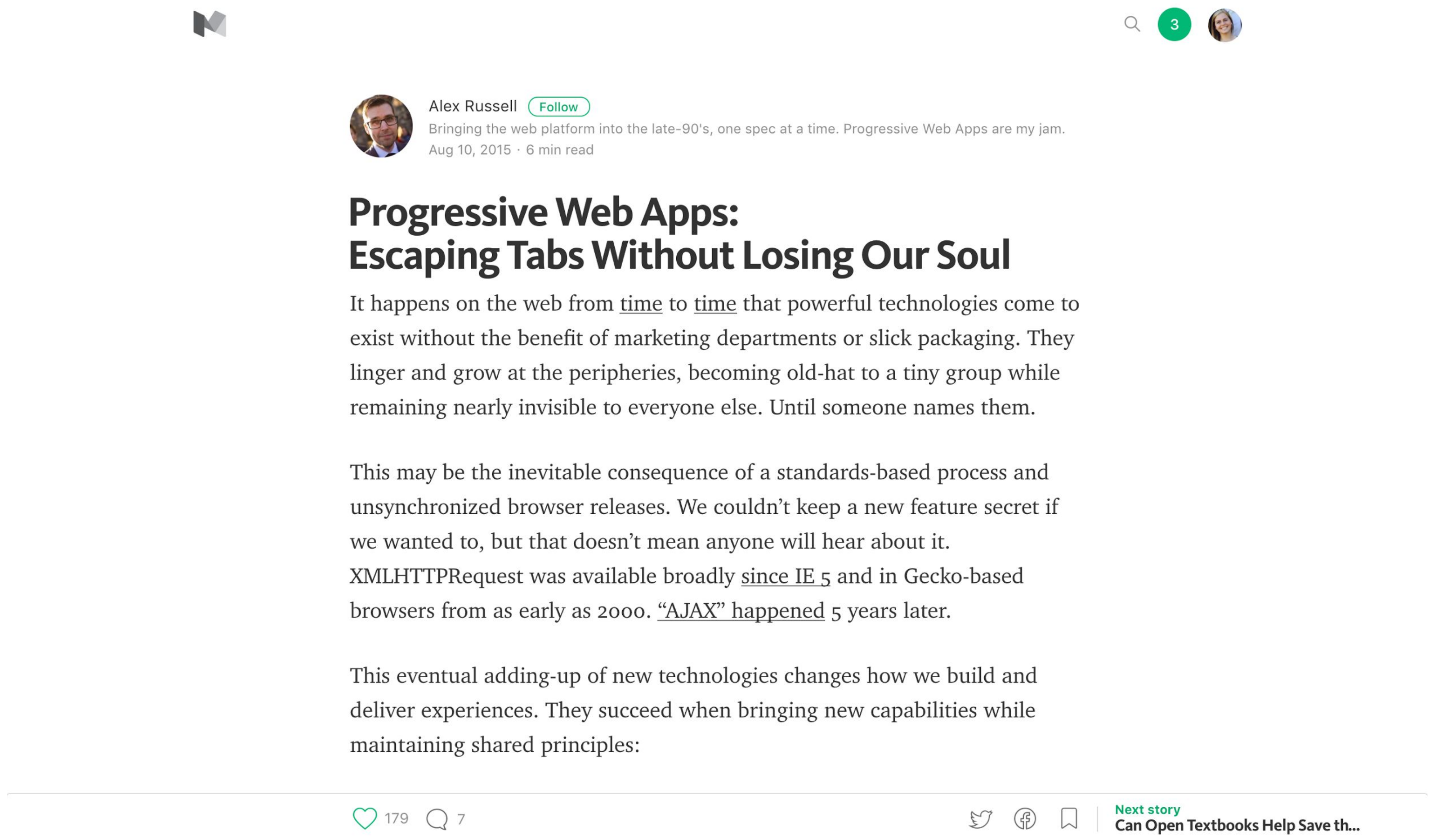
Less markup. Less JS. **Less confusion.**

```
<paper-tabs>
  <paper-tab>KNOWLEDGE</paper-tab>
  <paper-tab>HISTORY</paper-tab>
  <paper-tab>FOOD</paper-tab>
</paper-tabs>
```



2. PWAs

Progressive Web Apps



Alex Russell Follow

Bringing the web platform into the late-90's, one spec at a time. Progressive Web Apps are my jam.

Aug 10, 2015 · 6 min read

Progressive Web Apps: Escaping Tabs Without Losing Our Soul

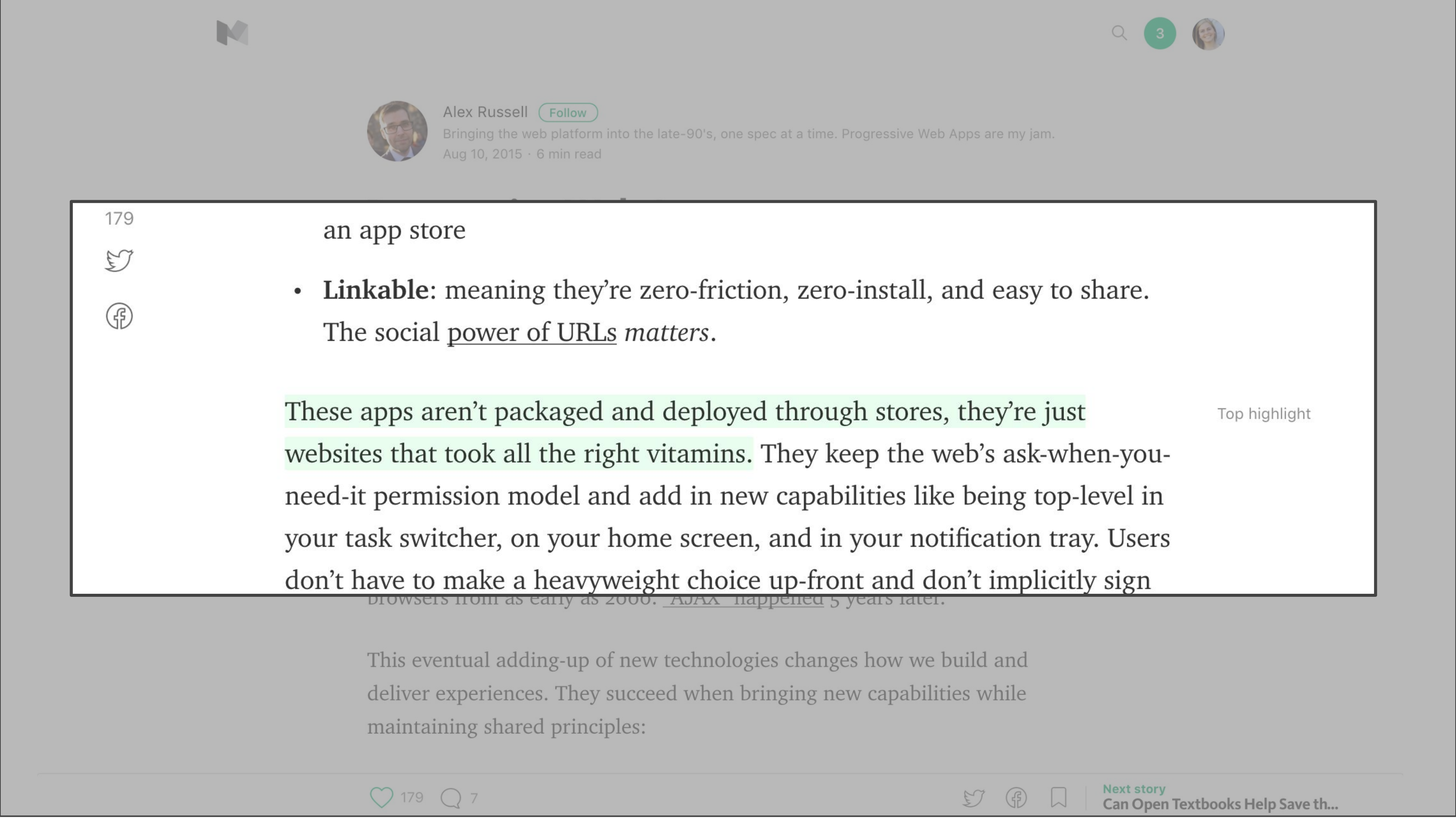
It happens on the web from time to time that powerful technologies come to exist without the benefit of marketing departments or slick packaging. They linger and grow at the peripheries, becoming old-hat to a tiny group while remaining nearly invisible to everyone else. Until someone names them.

This may be the inevitable consequence of a standards-based process and unsynchronized browser releases. We couldn't keep a new feature secret if we wanted to, but that doesn't mean anyone will hear about it. XMLHTTPRequest was available broadly since IE 5 and in Gecko-based browsers from as early as 2000. "AJAX" happened 5 years later.

This eventual adding-up of new technologies changes how we build and deliver experiences. They succeed when bringing new capabilities while maintaining shared principles:

179 7

Next story
Can Open Textbooks Help Save th...



Alex Russell Follow

Bringing the web platform into the late-90's, one spec at a time. Progressive Web Apps are my jam.

Aug 10, 2015 · 6 min read

179

an app store

- **Linkable**: meaning they're zero-friction, zero-install, and easy to share. The social power of URLs *matters*.

These apps aren't packaged and deployed through stores, they're just websites that took all the right vitamins. They keep the web's ask-when-you-need-it permission model and add in new capabilities like being top-level in your task switcher, on your home screen, and in your notification tray. Users don't have to make a heavyweight choice up-front and don't implicitly sign

Top highlight

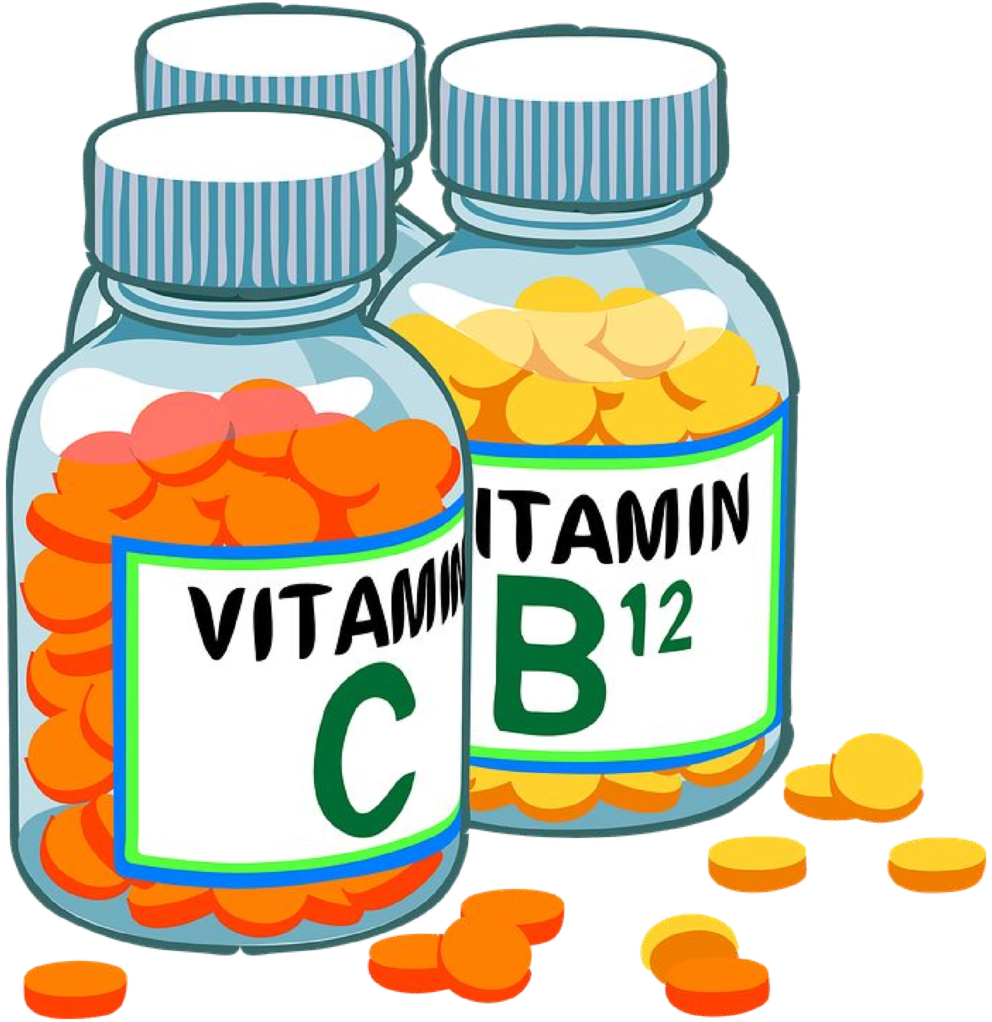
browsers from as early as 2006. "AJAX" happened 5 years later.

This eventual adding-up of new technologies changes how we build and deliver experiences. They succeed when bringing new capabilities while maintaining shared principles:

179 7

Next story

Can Open Textbooks Help Save th...


VITAMIN
C
VITAMIN
B12

AliExpress

Case Study

2X increase

in page visits per session

104% increase

in conversion rate for new users

https://developers.google.com/web/showcase/

Konga & Jumia

Case Study

Loads 2x faster
than their previous web site

Uses 6x less data
than their native app counterparts

https://developers.google.com/web/showcase/

How did they do it?

How did they do it?

PWAs

Reliable

Fast

Engaging

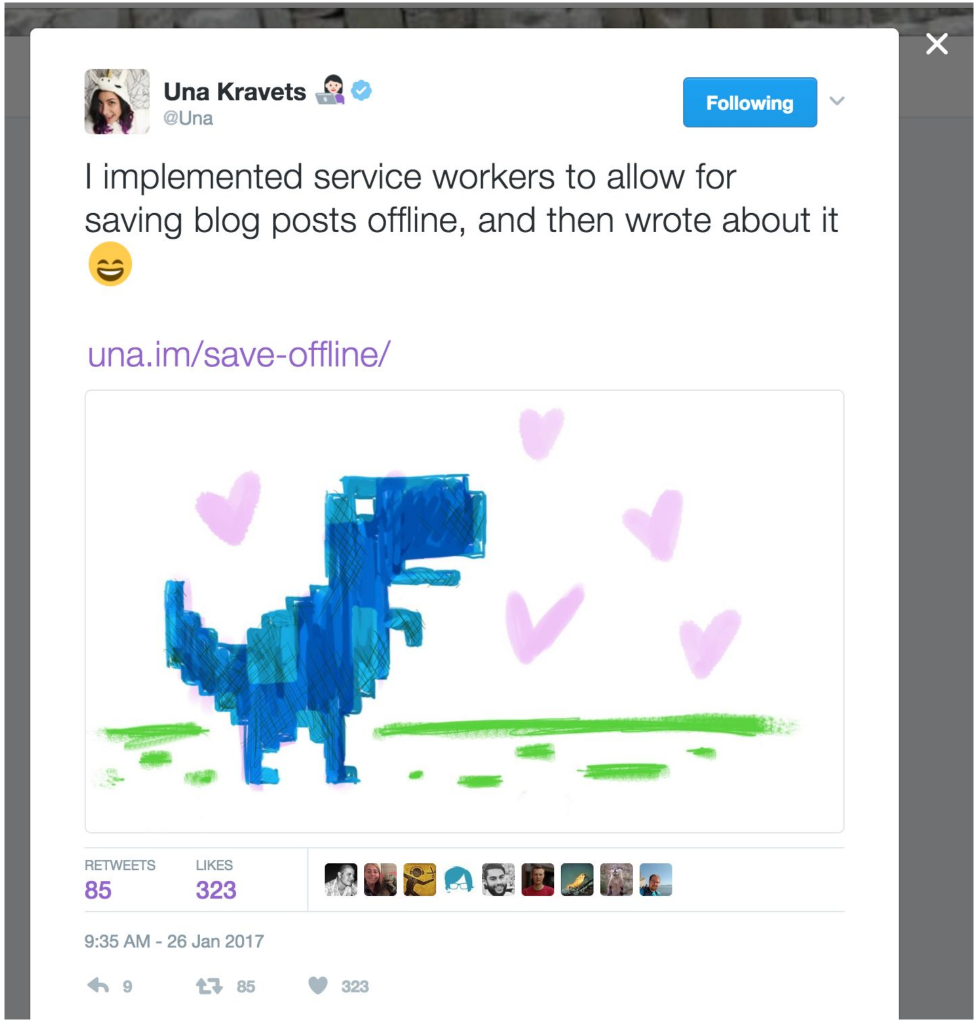
checklist from /web

Reliable

What?
Load instantly, regardless of network connection

Why?
Send less data for initial page loads, save your users money

How?
Service Workers



Una Kravets 👩🏻‍💻 @Una

Following

I implemented service workers to allow for saving blog posts offline, and then wrote about it 😄

una.im/save-offline/

RETWEETS 85 LIKES 323

9:35 AM - 26 Jan 2017

9 85 323

The Service Worker

If you're wondering what a service worker is, it's like a little alien that lives on your page and relays messages for you. It can detect when you have an Internet connection and

una.im/save-offline/

RETWEETS 85 LIKES 323

9:35 AM - 26 Jan 2017

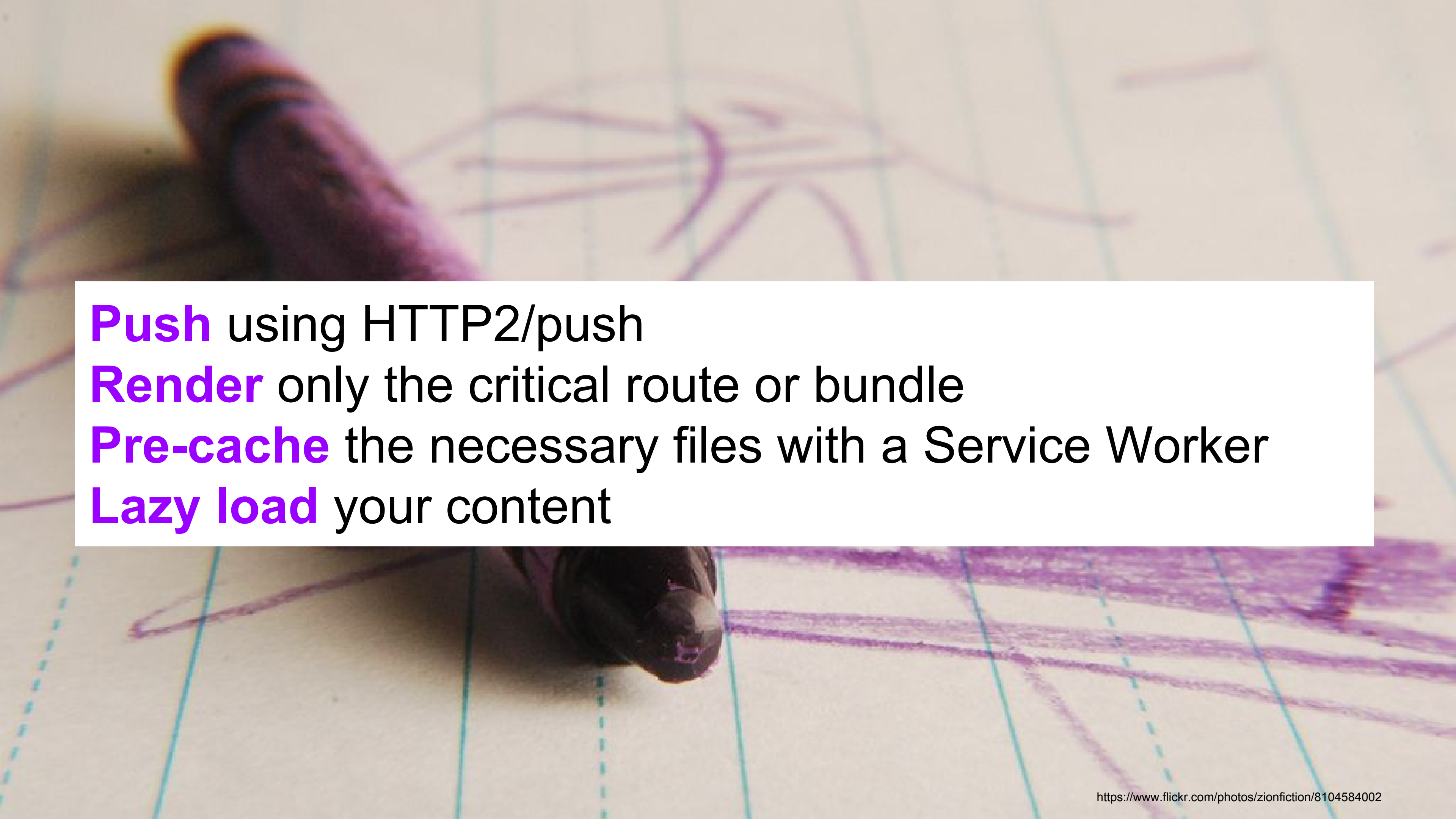
9 85 323

Fast

What?
No janky scrolling or slow to respond interfaces.

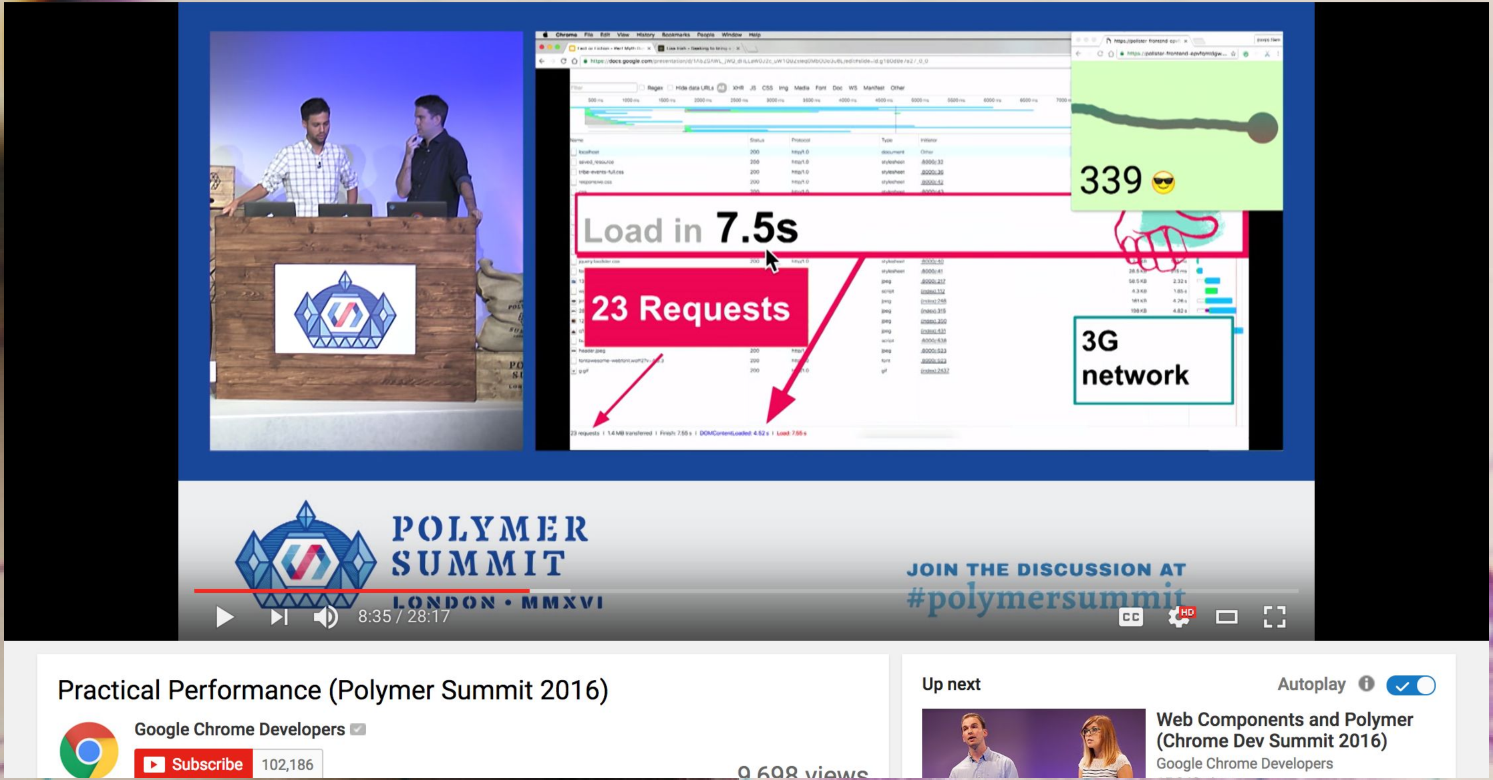
Why?
53% of users will abandon a site if it takes longer than 3 seconds to load!

How?
PRPL pattern
(Push, Render, Pre-cache, Lazy load)



Push using HTTP2/push
Render only the critical route or bundle
Pre-cache the necessary files with a Service Worker
Lazy load your content

https://www.flickr.com/photos/zionfiction/8104584002



Practical Performance (Polymer Summit 2016)

Google Chrome Developers

Subscribe 102,186

9,698 views

Up next

Autoplay

Web Components and Polymer (Chrome Dev Summit 2016)

Google Chrome Developers

https://www.youtube.com/watch?v=6m_E-mC0y3Y

https://www.flickr.com/photos/zionfiction/8104584002

Engaging

What?

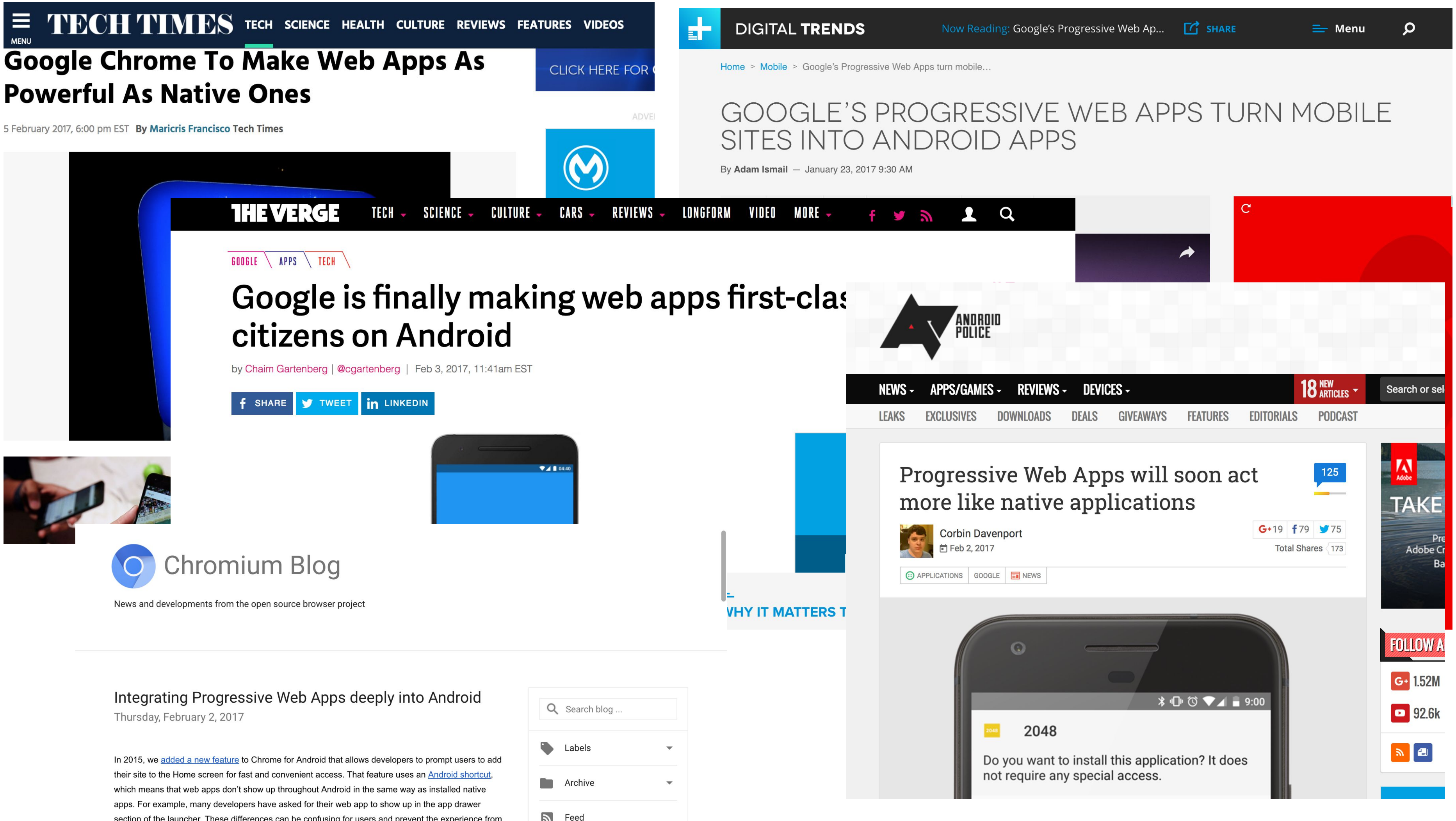
Live anywhere an app can, send notifications to the user

Why?

Increase transactions and interactions with your app

How?

Web Push Notifications API, Add to Home screen



MENU
TECH TIMES
TECH SCIENCE HEALTH CULTURE REVIEWS FEATURES VIDEOS
CLICK HERE FOR

Google Chrome To Make Web Apps As Powerful As Native Ones

5 February 2017, 6:00 pm EST By Maricris Francisco Tech Times

DIGITAL TRENDS
Now Reading: Google's Progressive Web Ap...
SHARE
Menu
Home > Mobile > Google's Progressive Web Apps turn mobile...

GOOGLE'S PROGRESSIVE WEB APPS TURN MOBILE SITES INTO ANDROID APPS

By Adam Ismail — January 23, 2017 9:30 AM

THE VERGE
TECH SCIENCE CULTURE CARS REVIEWS LONGFORM VIDEO MORE
GOOGLE APPS TECH

Google is finally making web apps first-clas citizens on Android

by Chaim Gartenberg | @cgartenberg | Feb 3, 2017, 11:41am EST

SHARE TWEET LINKEDIN
ANDROID POLICE
NEWS APPS/GAMES REVIEWS DEVICES
18 NEW ARTICLES
Search or sel
LEAKS EXCLUSIVES DOWNLOADS DEALS GIVEAWAYS FEATURES EDITORIALS PODCAST

Progressive Web Apps will soon act more like native applications

Corbin Davenport
Feb 2, 2017

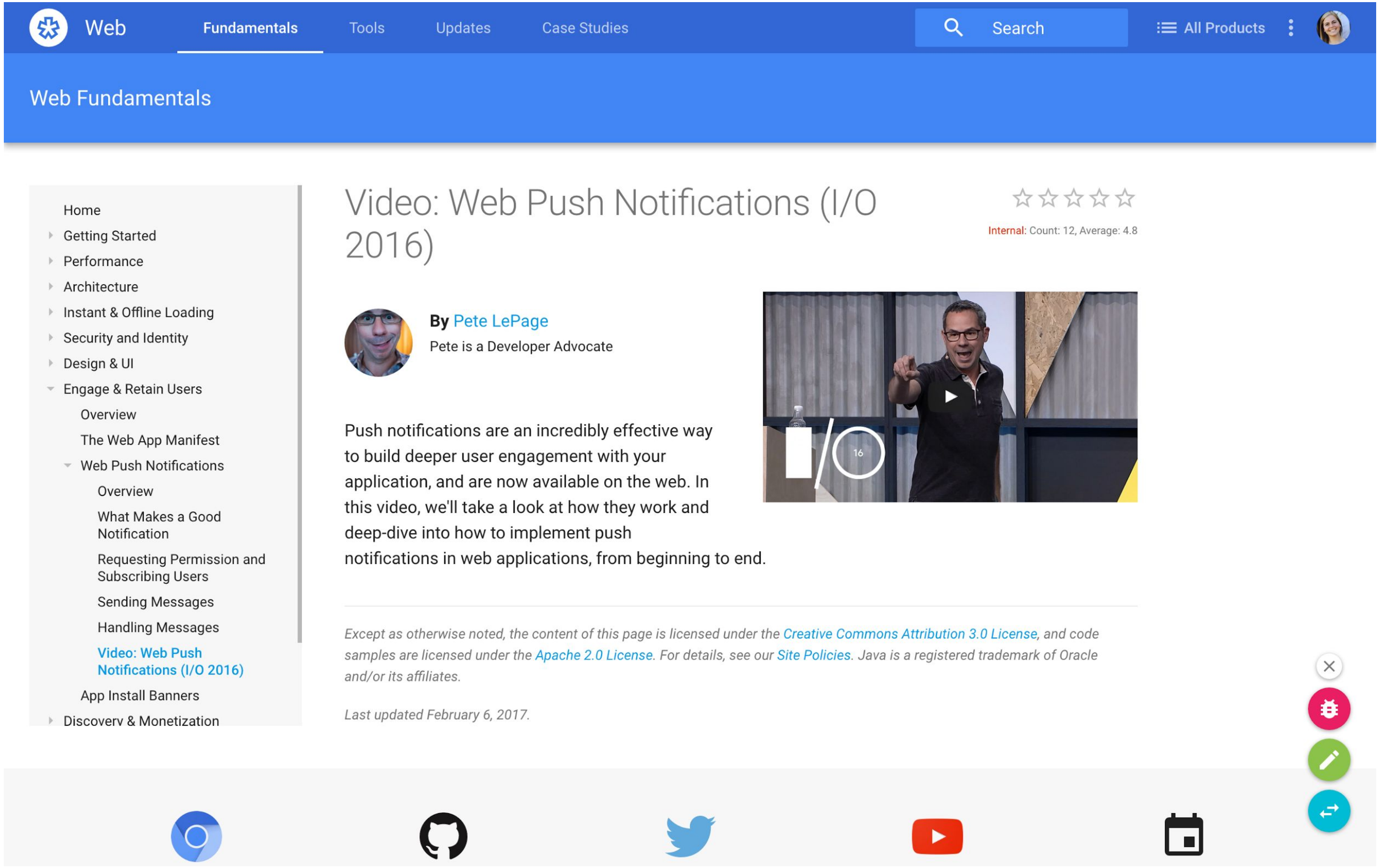
Do you want to install this application? It does not require any special access.

News and developments from the open source browser project

Integrating Progressive Web Apps deeply into Android

Thursday, February 2, 2017

In 2015, we added a new feature to Chrome for Android that allows developers to prompt users to add their site to the Home screen for fast and convenient access. That feature uses an Android shortcut, which means that web apps don't show up throughout Android in the same way as installed native apps. For example, many developers have asked for their web app to show up in the app drawer section of the launcher. These differences can be confusing for users and prevent the experience from



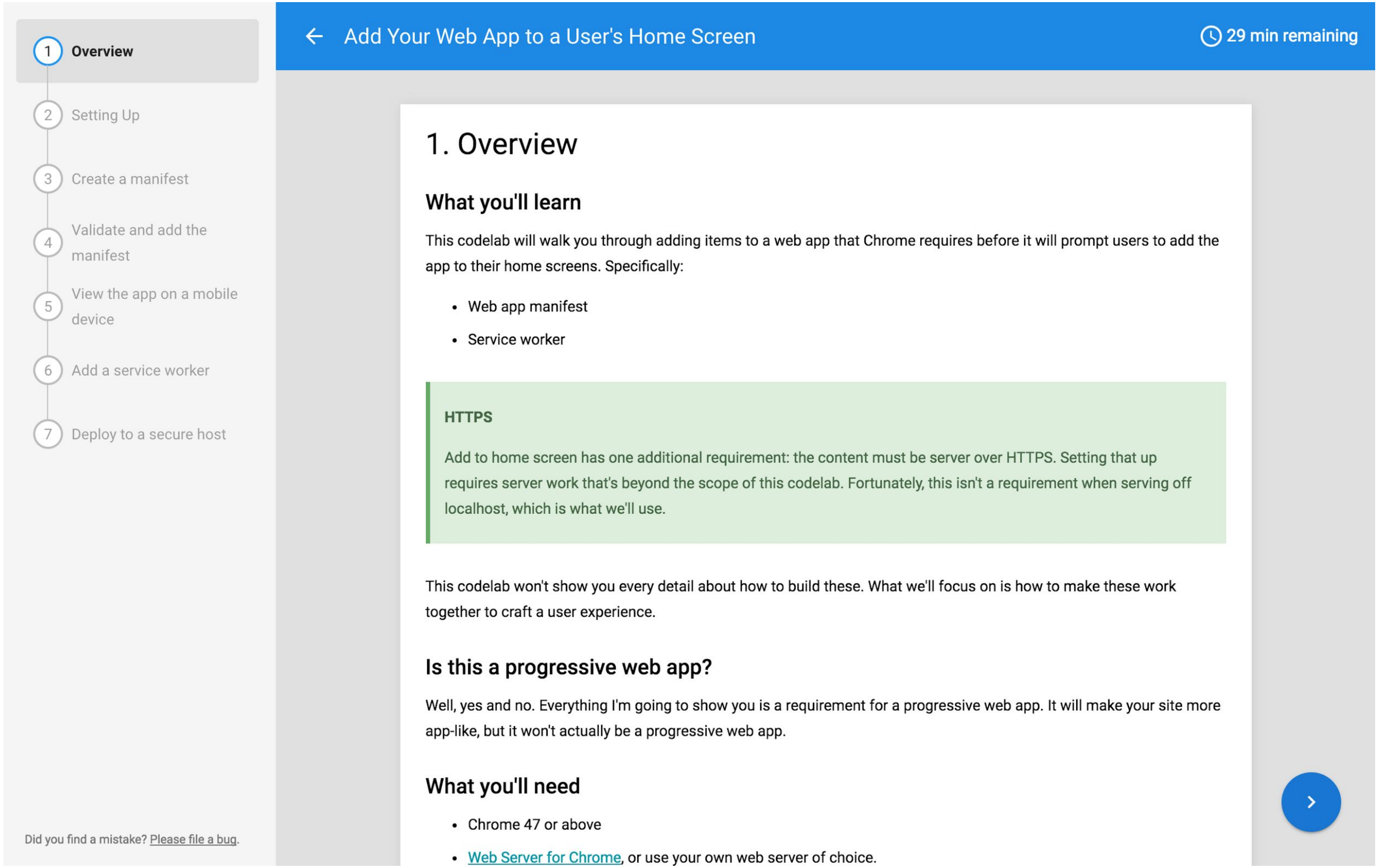
Video: Web Push Notifications (I/O 2016)

Internal: Count: 12, Average: 4.8

By Pete LePage
Pete is a Developer Advocate

Push notifications are an incredibly effective way to build deeper user engagement with your application, and are now available on the web. In this video, we'll take a look at how they work and deep-dive into how to implement push notifications in web applications, from beginning to end.

Last updated February 6, 2017.



Add Your Web App to a User's Home Screen

29 min remaining

Did you find a mistake? Please file a bug.

1. Overview

What you'll learn

This codelab will walk you through adding items to a web app that Chrome requires before it will prompt users to add the app to their home screens. Specifically:

- Web app manifest
- Service worker

HTTPS

Add to home screen has one additional requirement: the content must be server over HTTPS. Setting that up requires server work that's beyond the scope of this codelab. Fortunately, this isn't a requirement when serving off localhost, which is what we'll use.

This codelab won't show you every detail about how to build these. What we'll focus on is how to make these work together to craft a user experience.

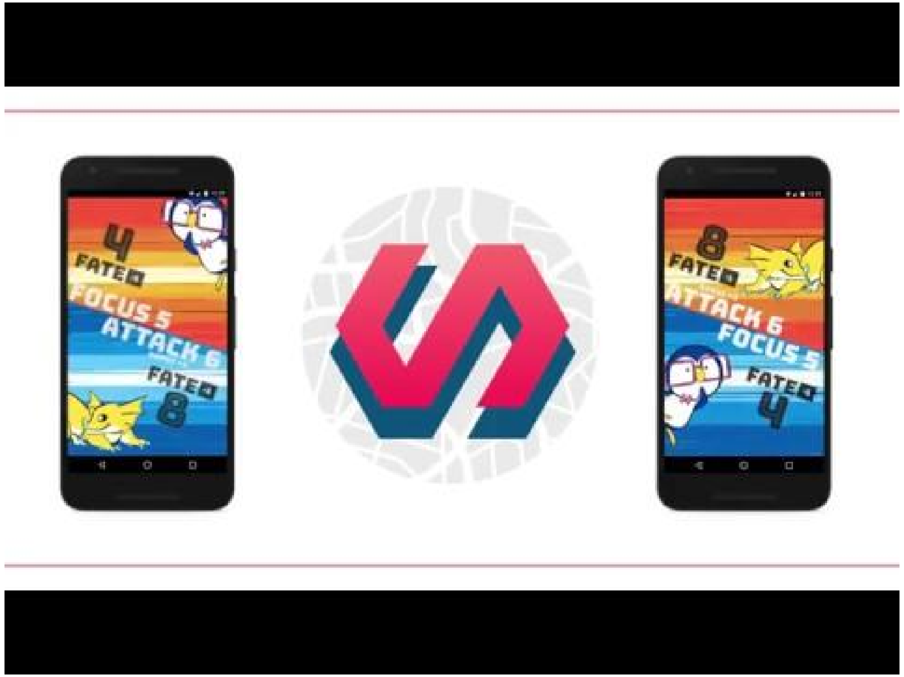
Is this a progressive web app?

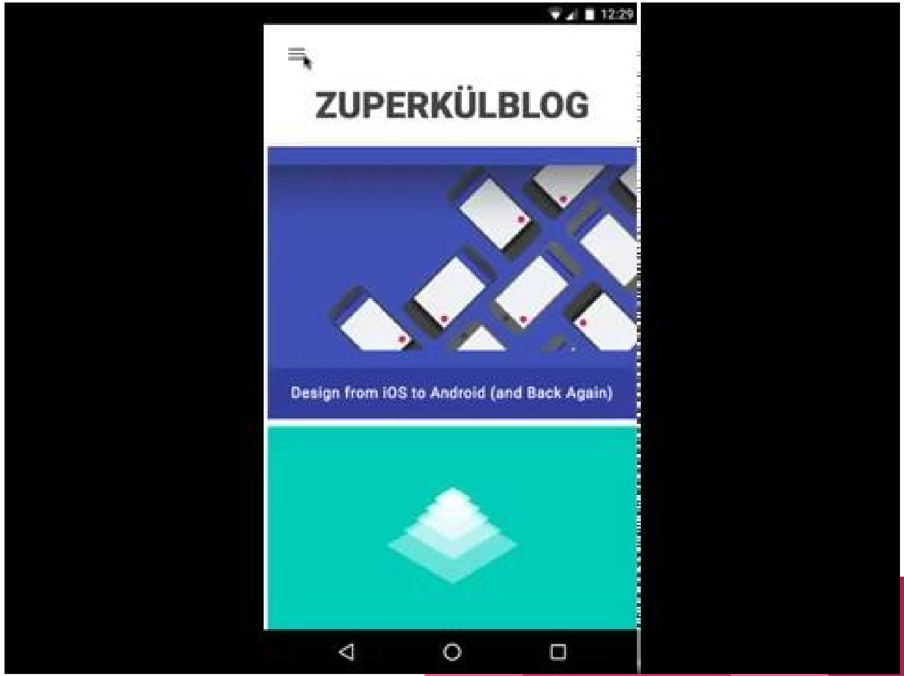
Well, yes and no. Everything I'm going to show you is a requirement for a progressive web app. It will make your site more app-like, but it won't actually be a progressive web app.

What you'll need

- Chrome 47 or above
- Web Server for Chrome, or use your own web server of choice.

What do they look like?





checklist from /web

We’re here to help!

developers.google.com/web

Progressive Web Apps
Progressive Web App Checklist

Site's content is indexed by Google

To Test	Use the Fetch as Google tool to preview how Google will see your site when it is crawled.
To Fix	Google's indexing system does run JavaScript but some issues may need to be fixed to make content accessible. For example, if you are using new browser features like the Fetch API, ensure that they are polyfilled in browsers without support.

Schema.org metadata is provided where appropriate

Schema.org metadata can help improve the appearance of your site in search engines.

To Test	Use the testing tool to ensure title, image, description etc. are available.
To Fix	Markup the content. For example: • A recipe app should have the Recipe type markup for Rich Cards. • A news app should have the NewsArticle type markup for Rich Cards and/or AMP support. • An ecommerce app should have the Product type markup for Rich Cards.

Social metadata is provided where appropriate

To Test	• Open a representative page in Facebook's crawler and ensure it looks reasonable. • Check that Twitter Cards meta data is present (for example `<meta name="twitter:card" content="summary" />`) if you feel it would be appropriate.
To Fix	Mark up content with Open Graph tags and as advised by Twitter.

Canonical URLs are provided when necessary

This is only necessary if your content is available at multiple URLs.

To Test	• Determine whether any piece of content is available at two different URLs. • Open both of these pages and ensure they use `<link rel=canonical>` tags in the head to indicate the canonical version.

Here’s a head start

https://www.polymer-project.org/1.0/toolbox/

3. Polymer + PWAs

with the Polymer App Toolbox

Polymer App Toolbox

Templates

Layout

Localization

Routing

App Storage

Service Worker

Polymer App Toolbox

Templates

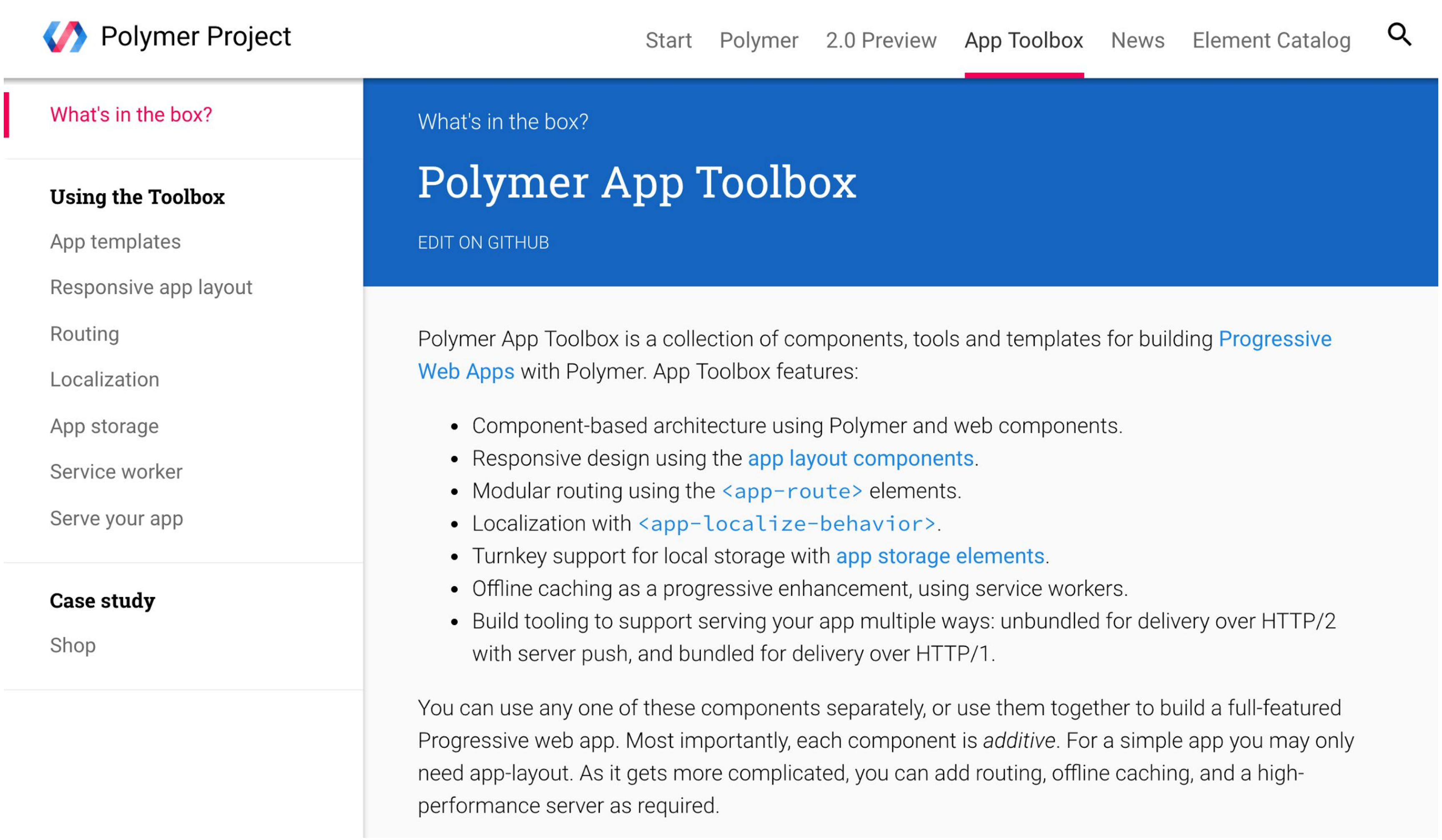
Layout

Localization

Routing

App Storage

Service Worker


Polymer Project

Start Polymer 2.0 Preview App Toolbox News Element Catalog

What's in the box?

Using the Toolbox

App templates

Responsive app layout

Routing

Localization

App storage

Service worker

Serve your app

Case study

Shop

What's in the box?

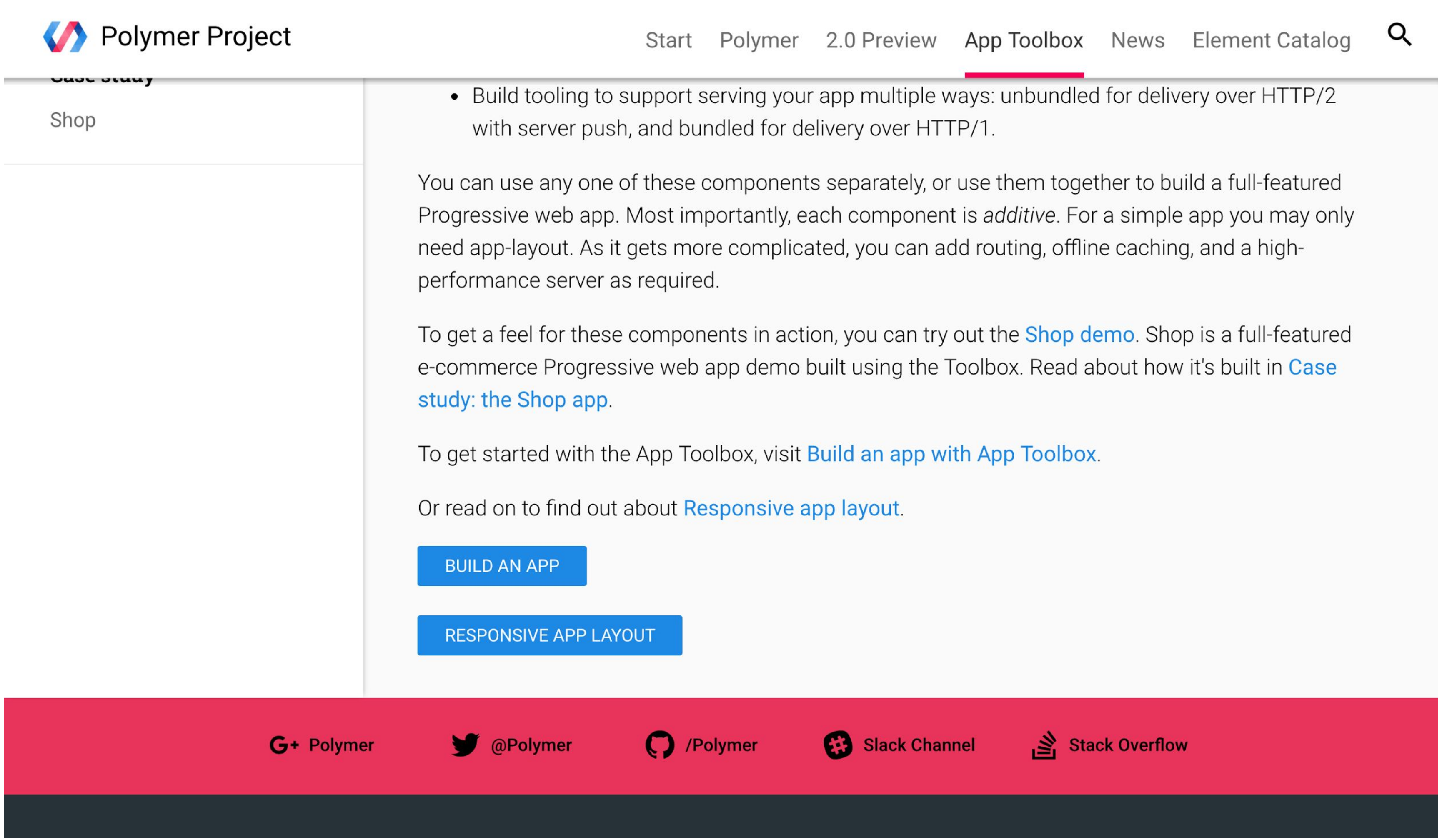
Polymer App Toolbox

EDIT ON GITHUB

Polymer App Toolbox is a collection of components, tools and templates for building Progressive Web Apps with Polymer. App Toolbox features:

- Component-based architecture using Polymer and web components.
- Responsive design using the app layout components.
- Modular routing using the `<app-route>` elements.
- Localization with `<app-localize-behavior>`.
- Turnkey support for local storage with app storage elements.
- Offline caching as a progressive enhancement, using service workers.
- Build tooling to support serving your app multiple ways: unbundled for delivery over HTTP/2 with server push, and bundled for delivery over HTTP/1.

You can use any one of these components separately, or use them together to build a full-featured Progressive web app. Most importantly, each component is *additive*. For a simple app you may only need app-layout. As it gets more complicated, you can add routing, offline caching, and a high-performance server as required.



Polymer Project

Start Polymer 2.0 Preview App Toolbox News Element Catalog

Case study

Shop

- Build tooling to support serving your app multiple ways: unbundled for delivery over HTTP/2 with server push, and bundled for delivery over HTTP/1.

You can use any one of these components separately, or use them together to build a full-featured Progressive web app. Most importantly, each component is *additive*. For a simple app you may only need app-layout. As it gets more complicated, you can add routing, offline caching, and a high-performance server as required.

To get a feel for these components in action, you can try out the Shop demo. Shop is a full-featured e-commerce Progressive web app demo built using the Toolbox. Read about how it's built in Case study: the Shop app.

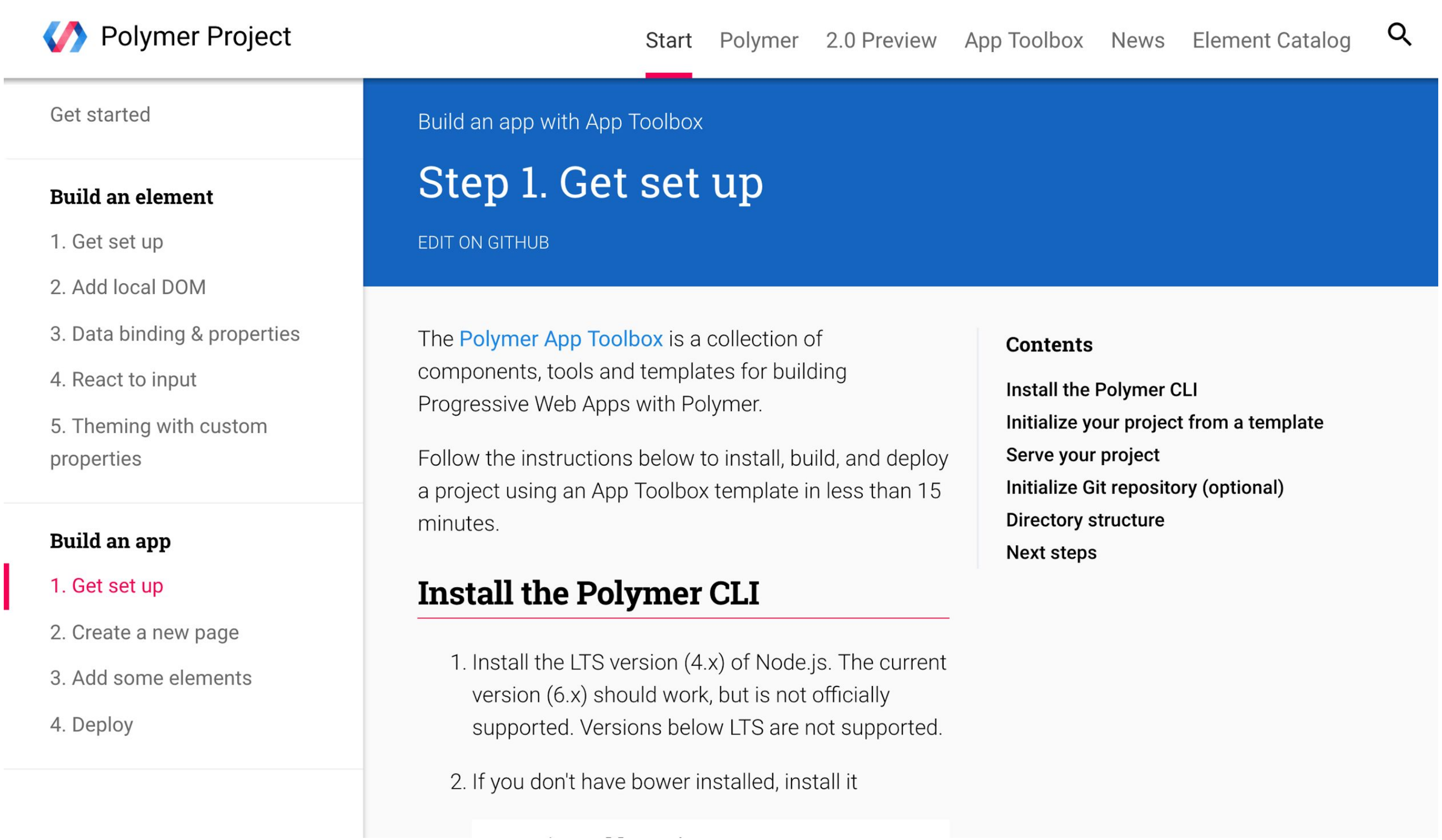
To get started with the App Toolbox, visit Build an app with App Toolbox.

Or read on to find out about Responsive app layout.

BUILD AN APP

RESPONSIVE APP LAYOUT

G+ Polymer @Polymer /Polymer Slack Channel Stack Overflow


Polymer Project

Start Polymer 2.0 Preview App Toolbox News Element Catalog

Get started

Build an element

1. Get set up
2. Add local DOM
3. Data binding & properties
4. React to input
5. Theming with custom properties

Build an app

1. Get set up
2. Create a new page
3. Add some elements
4. Deploy

Build an app with App Toolbox

Step 1. Get set up

EDIT ON GITHUB

The Polymer App Toolbox is a collection of components, tools and templates for building Progressive Web Apps with Polymer.

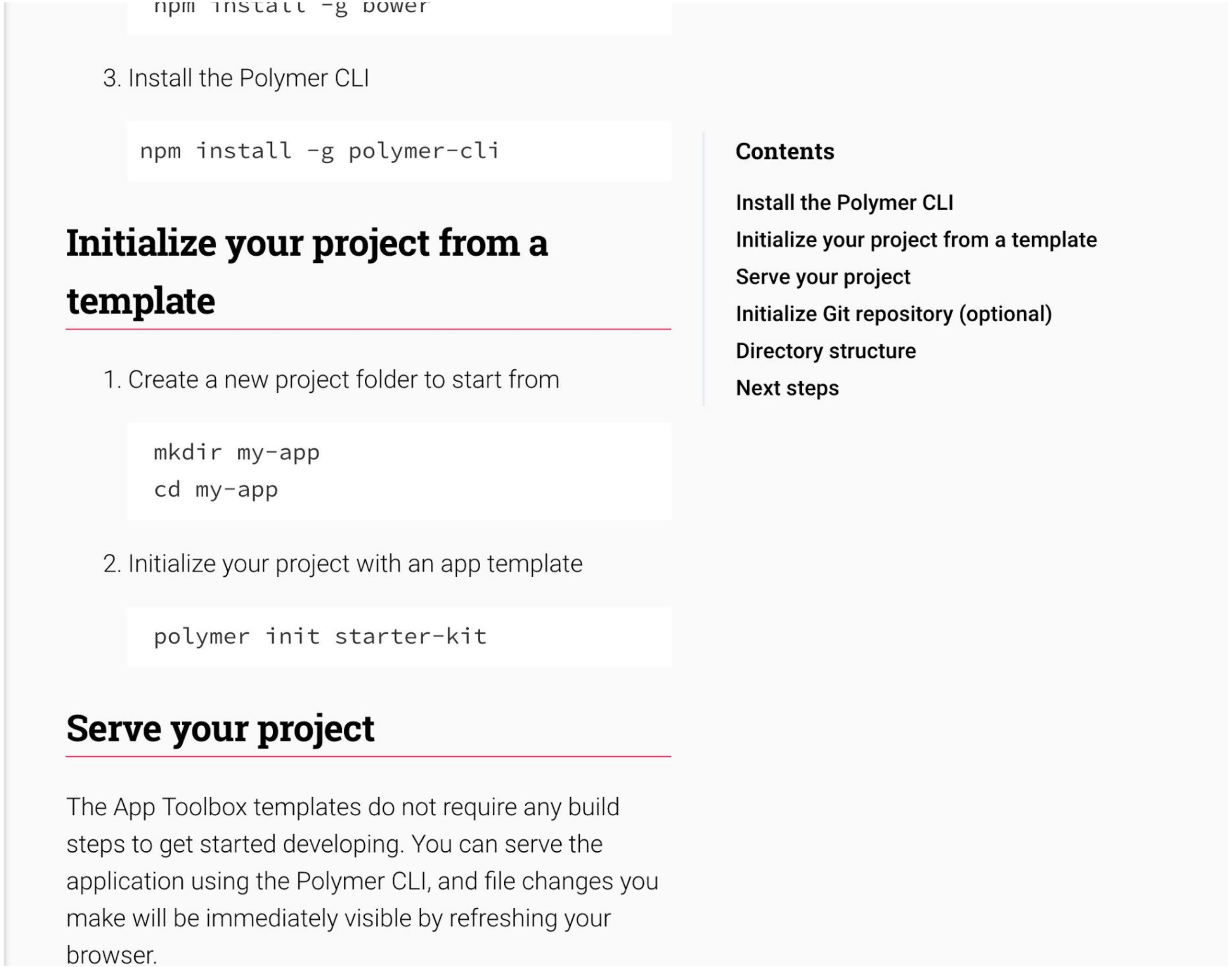
Follow the instructions below to install, build, and deploy a project using an App Toolbox template in less than 15 minutes.

Contents

Install the Polymer CLI
Initialize your project from a template
Serve your project
Initialize Git repository (optional)
Directory structure
Next steps

Install the Polymer CLI

1. Install the LTS version (4.x) of Node.js. The current version (6.x) should work, but is not officially supported. Versions below LTS are not supported.
2. If you don't have bower installed, install it



```
npm install -g bower
```

3. Install the Polymer CLI

```
npm install -g polymer-cli
```

Initialize your project from a template

1. Create a new project folder to start from

```
mkdir my-app
cd my-app
```

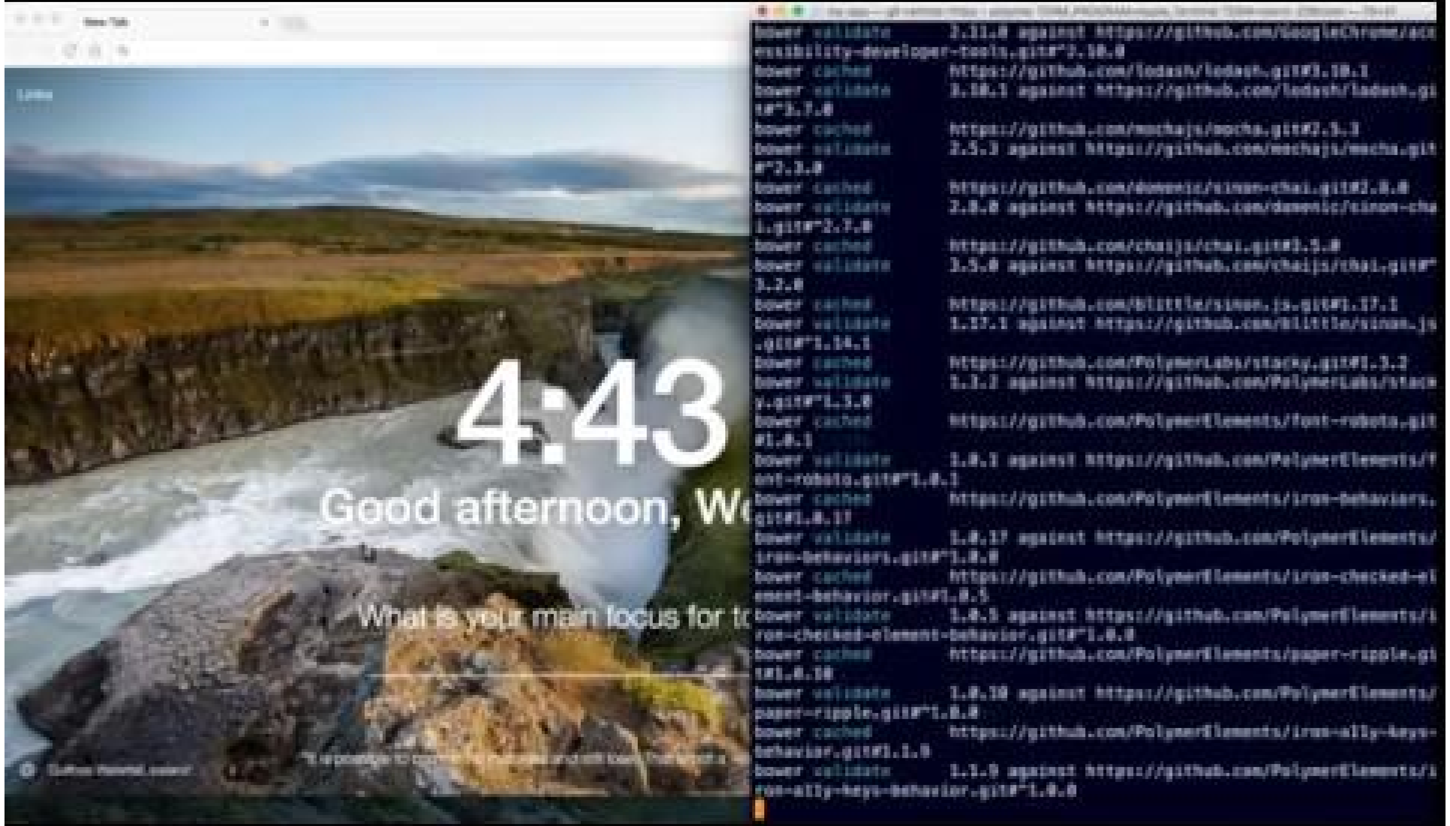
2. Initialize your project with an app template

```
polymer init starter-kit
```

Serve your project

The App Toolbox templates do not require any build steps to get started developing. You can serve the application using the Polymer CLI, and file changes you make will be immediately visible by refreshing your browser.

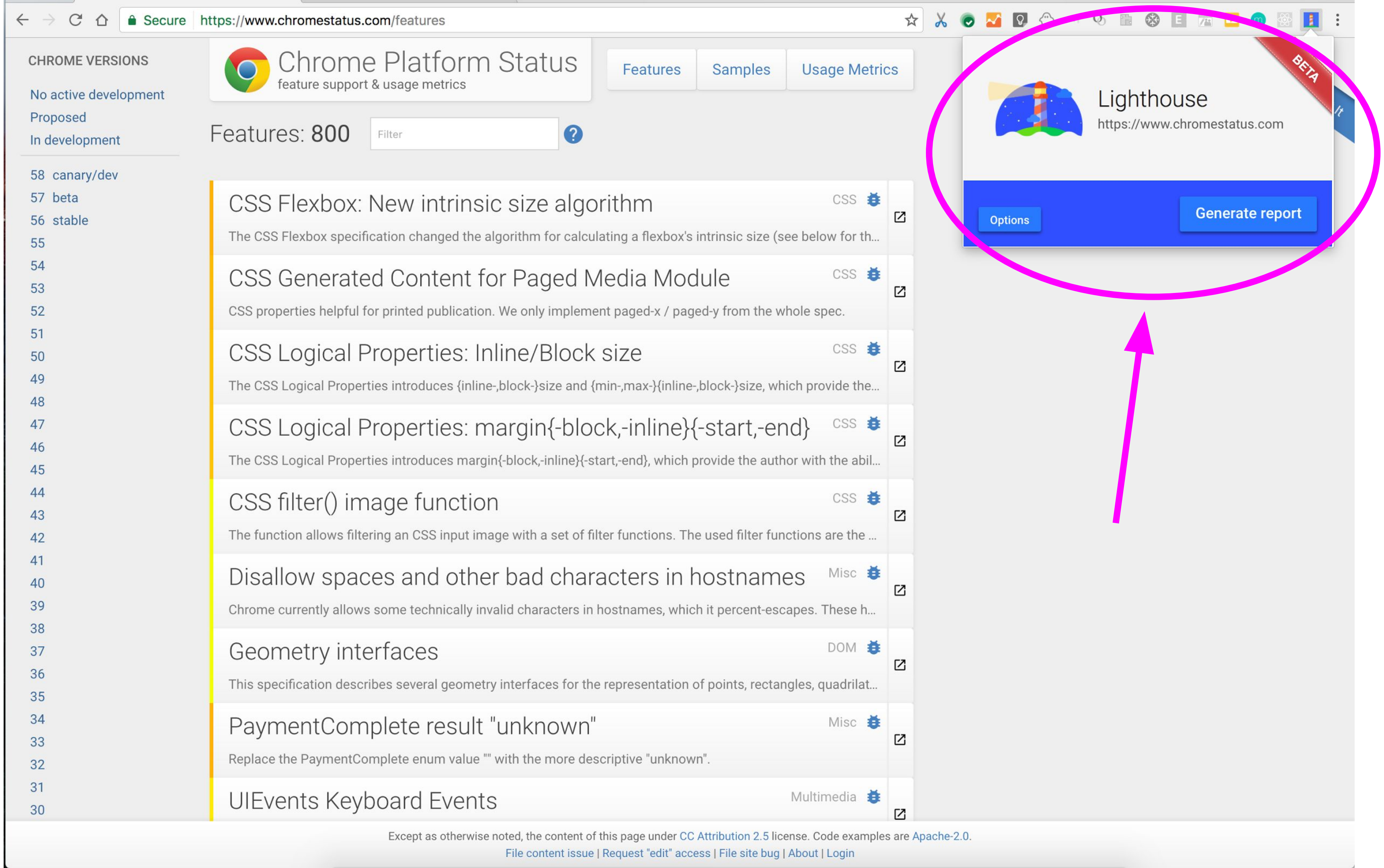
Contents



```
bower validate      2.11.0 against https://github.com/GoogleChrome/acc
essibility-developer-tools.git#^2.10.0
bower cached        https://github.com/lodash/lodash.git#3.10.1
bower validate      3.10.1 against https://github.com/lodash/lodash.gi
t#^3.7.0
bower cached        https://github.com/mochajs/mocha.git#2.5.3
bower validate      2.5.3 against https://github.com/mochajs/mocha.git
#^2.3.0
bower cached        https://github.com/domenic/sinon-chai.git#2.8.0
bower validate      2.8.0 against https://github.com/domenic/sinon-cha
i.git#^2.7.0
bower cached        https://github.com/chaijs/chai.git#3.5.0
bower validate      3.5.0 against https://github.com/chaijs/chai.git#^
3.2.0
bower cached        https://github.com/blittle/sinon.js.git#1.17.1
bower validate      1.17.1 against https://github.com/blittle/sinon.js
.git#^1.14.1
bower cached        https://github.com/PolymerLabs/stacky.git#1.3.2
bower validate      1.3.2 against https://github.com/PolymerLabs/stack
y.git#^1.3.0
bower cached        https://github.com/PolymerElements/font-roboto.git
#1.0.1
bower validate      1.0.1 against https://github.com/PolymerElements/f
ont-roboto.git#^1.0.1
bower cached        https://github.com/PolymerElements/iron-behaviors.
git#1.0.17
bower validate      1.0.17 against https://github.com/PolymerElements/
iron-behaviors.git#^1.0.0
bower cached        https://github.com/PolymerElements/iron-checked-el
ement-behavior.git#1.0.5
bower validate      1.0.5 against https://github.com/PolymerElements/i
ron-checked-element-behavior.git#^1.0.0
bower cached        https://github.com/PolymerElements/paper-ripple.gi
t#1.0.10
bower validate      1.0.10 against https://github.com/PolymerElements/
paper-ripple.git#^1.0.0
bower cached        https://github.com/PolymerElements/iron-a11y-keys-
behavior.git#1.1.9
bower validate      1.1.9 against https://github.com/PolymerElements/i
ron-a11y-keys-behavior.git#^1.0.0
```

How can I evaluate my PWA?

https://developers.google.com/web/tools/lighthouse/



https://www.chromestatus.com/features

CHROME VERSIONS

No active development
Proposed
In development

58 canary/dev
57 beta
56 stable
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30

Chrome Platform Status
feature support & usage metrics

Features | Samples | Usage Metrics

Features: 800

CSS Flexbox: New intrinsic size algorithm — CSS
The CSS Flexbox specification changed the algorithm for calculating a flexbox's intrinsic size (see below for th...

CSS Generated Content for Paged Media Module — CSS
CSS properties helpful for printed publication. We only implement paged-x / paged-y from the whole spec.

CSS Logical Properties: Inline/Block size — CSS
The CSS Logical Properties introduces {inline-,block-}size and {min-,max-}{inline-,block-}size, which provide the...

CSS Logical Properties: margin{-block,-inline}{-start,-end} — CSS
The CSS Logical Properties introduces margin{-block,-inline}{-start,-end}, which provide the author with the abil...

CSS filter() image function — CSS
The function allows filtering an CSS input image with a set of filter functions. The used filter functions are the ...

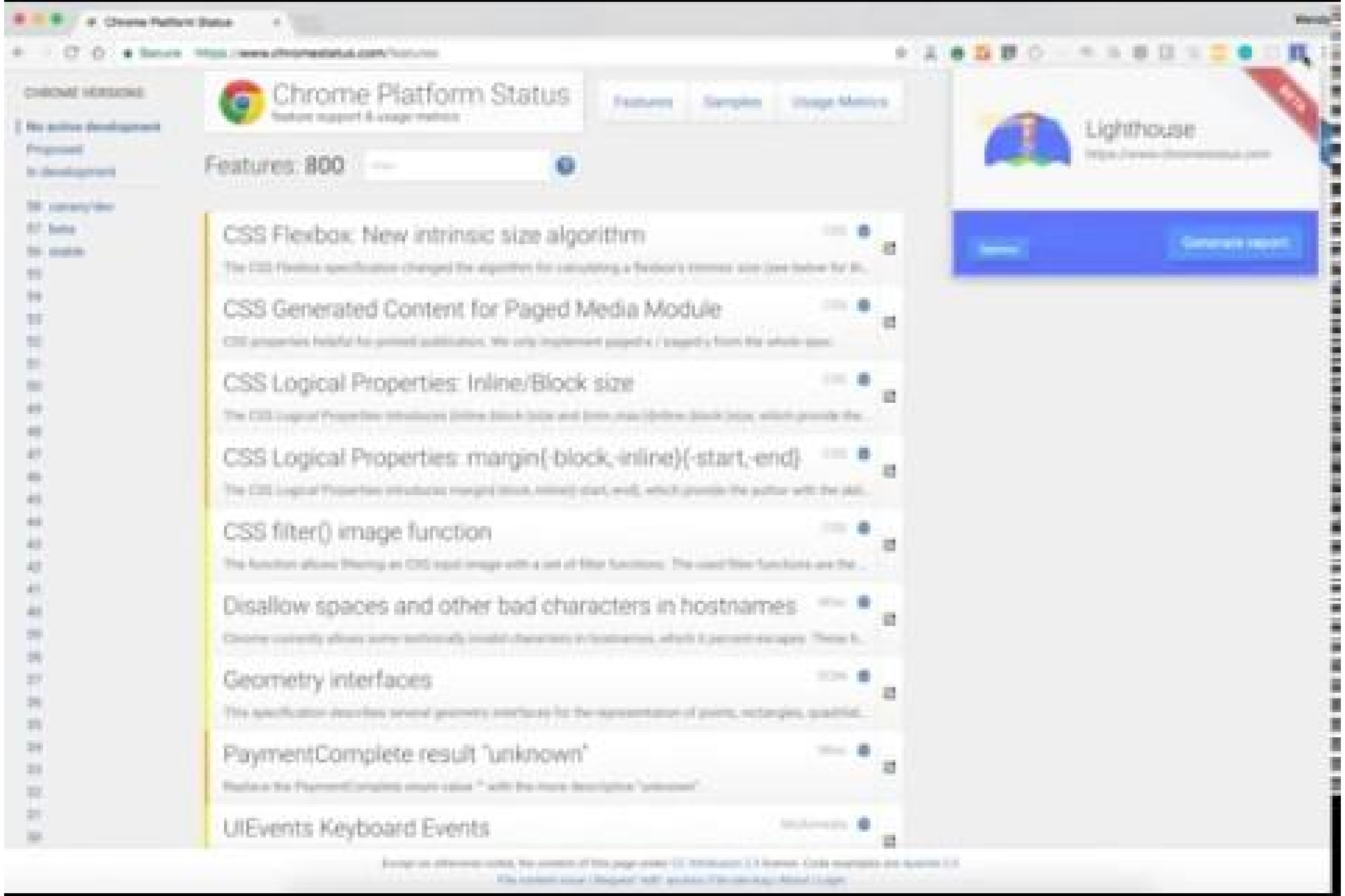
Disallow spaces and other bad characters in hostnames — Misc
Chrome currently allows some technically invalid characters in hostnames, which it percent-escapes. These h...

Geometry interfaces — DOM
This specification describes several geometry interfaces for the representation of points, rectangles, quadrilat...

PaymentComplete result "unknown" — Misc
Replace the PaymentComplete enum value "" with the more descriptive "unknown".

UIEvents Keyboard Events — Multimedia

Except as otherwise noted, the content of this page under CC Attribution 2.5 license. Code examples are Apache-2.0.
File content issue | Request "edit" access | File site bug | About | Login



CSS Flexbox: New intrinsic size algorithm

CSS Generated Content for Paged Media Module

CSS Logical Properties: Inline/Block size

CSS Logical Properties: margin{-block,-inline}{-start,-end}

CSS filter() image function

Disallow spaces and other bad characters in hostnames

Geometry interfaces

PaymentComplete result "unknown"

UIEvents Keyboard Events

1. Polymer 😎
2. PWAs 😎
3. Polymer + PWAs 😎

Thank you!

Wendy Ginsberg
@wmginsberg