Juiz de Fora, Maio 2015

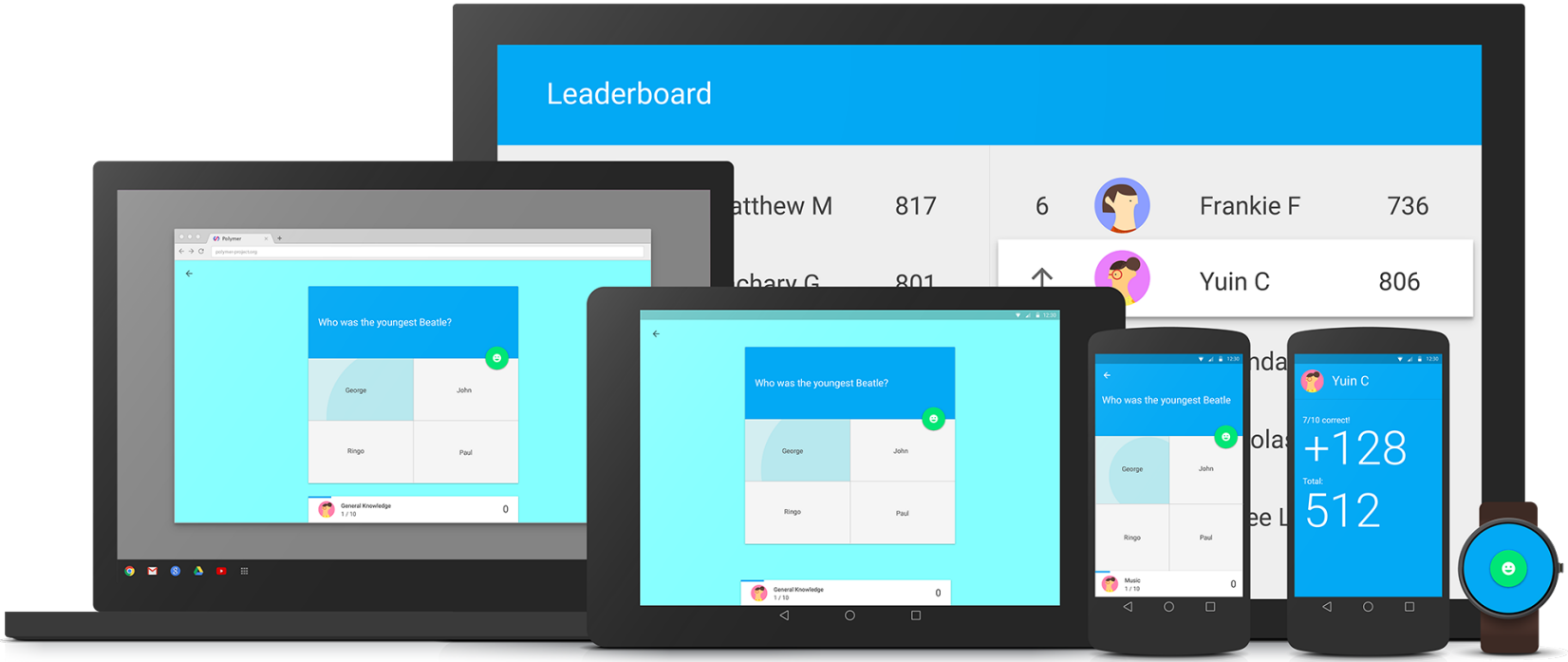
+Gilson

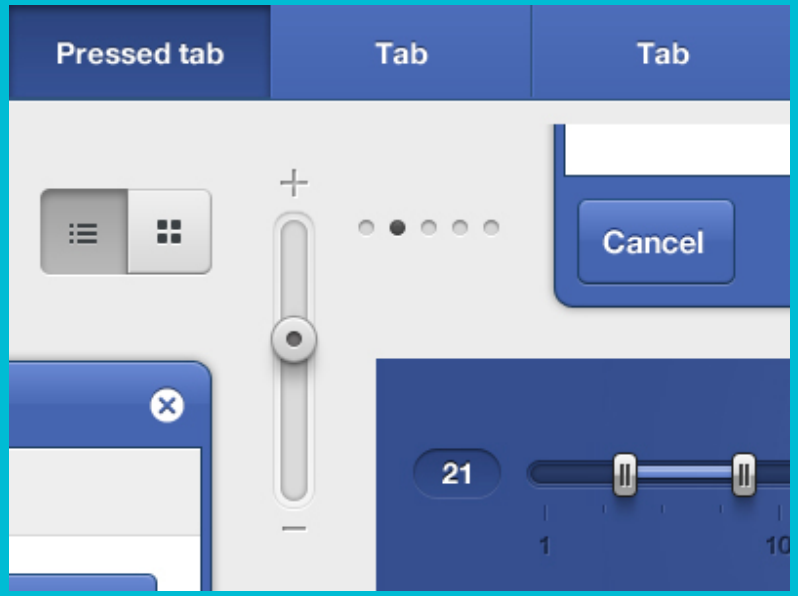
Agenda

Web Components
Polymer
Saiba Mais

Web Components

Quais problemas solucionar?


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



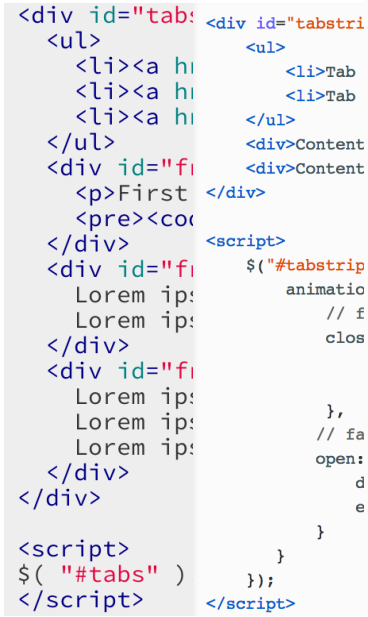
Contruir abas
Deveria ser fácil!

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 hr
    <li><a hr
    <li><a hr
  </ul>
  <div id="fr
    <p>First
    <pre><co
  </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
        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 h
    <li><a h
    <li><a h
  </ul>
  <div id="f
    <p>First
    <pre><co
  </div>
  <div id="f
    Lorem ips
    Lorem ips
  </div>
  <div id="f
    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 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
        animatio
            // f
            clos


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

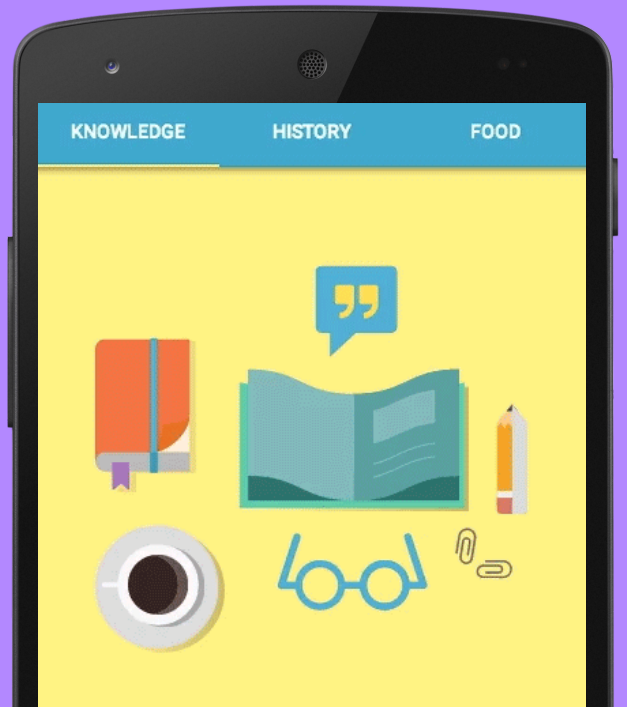
```
var tabview = ne
    children: [-
        label:
        content
    }, {
        label:
        content
    }, {
        label:
        content
    }]
});
```

```
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
  };
}).
```

Web Components

Menos código. Menos confusão.

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



O que são Web Components?

Custom Elements

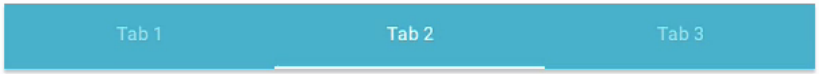
defina novos componentes HTML/DOM

Custom Elements

define novo
HTML

- **declarativo**,legível
- **Com significado** HTML
- **Extensivo** → reusável

```
<paper-tabs selected="1">
  <paper-tab>Tab 1</paper-tab>
  <paper-tab>Tab 2</paper-tab>
  <paper-tab>Tab 3</paper-tab>
</paper-tabs>
```



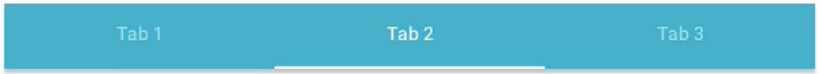
.selected = 1

Custom Elements

define novo
HTML

- **declarativo**,legível
- **Com significado** HTML
- **Extensivo** → reusável

```
var tabs = document.querySelector('paper-tabs');
tabs.addEventListener('core-activate', function() {
  console.log(this.selected);
});
```



.selected = 1

Templates

Template nativo para o client

HTML Templates

templates nativos para o client

- **usa DOM** para criar DOM → no XSS
- **analisado**, não renderizado
- **conteúdo é inerte** até que clonado/usado
- **fragmento doc**→ não é parte da página

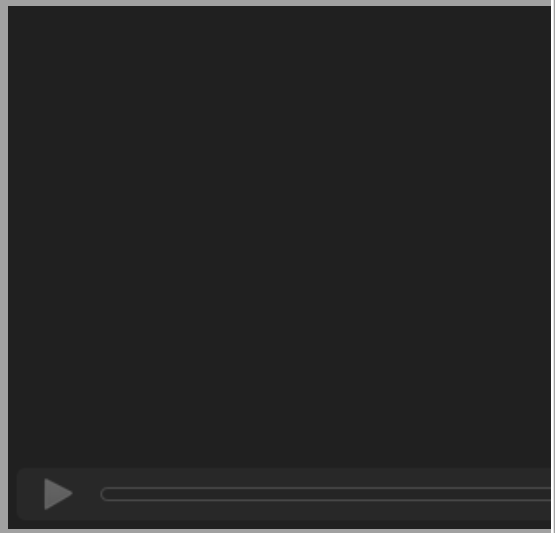
```
<template>
  <div class="comment">
    <img src="image.png">
  </div>
  <script>...</script>
</template>
```

Shadow DOM

DOM/CSS escopo


```
<video src="foo.webm" controls></video>
```


```
<video src="foo.webm" controls></video>
```



<video src=“foo.webm

Elements Network Sources » 1

```
<h3 class="__web-inspector-hide-shortcut__"><vid
▼<video src="./videos/bunny.webm" controls>
  ▼#shadow-root (user-agent)
    ▼<div>
      ▼<div>
        ▼<div>
          ▶<input type="button">
          ▶<input type="range" step="any" max="0">
           <div style="display: none;">0:00</div>
           <div>0:00</div>
          ▶<input type="button">
          ▶<input type="range" step="any" max="1"
          none;">
          ▶<input type="button" style="display: no
          ▶<input type="button" style="display: no
          </div>
        </div>
      </div>
  </video>
▶<pre style="text-align: center; font-size: 33px;
inspector-hide-shortcut__">…</pre>
```

html body div div section video #shadow-root div

Styles Event Listeners DOM Breakpoints Properties

Find in Styles Filter

HTML Imports

carregando web components

Example: Bootstrap

```
<link rel="stylesheet" href="bootstrap.css">
<link rel="stylesheet" href="fonts.css">
<script src="jquery.js"></script>
<script src="bootstrap.js"></script>
<script src="bootstrap-tooltip.js"></script>
<script src="bootstrap-dropdown.js"></script>
```

Example: Bootstrap

```
<link rel="import" href="bootstrap.html">
```

Custom Elements

Cria novos elementos HTML e extende os já existentes

Templates

Native templating in the browser

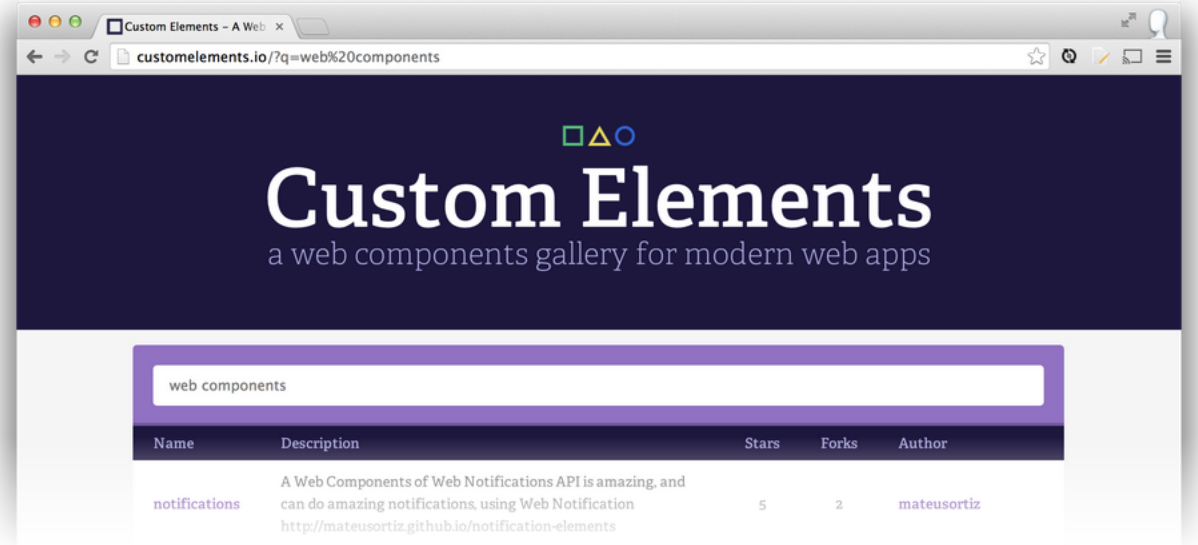
Shadow DOM

Scoped CSS!!! + encapsulated markup

HTML Imports

Carrega elementos customizados, definições e resources

FIND




```
$ bower install Polymer/core-toolbar
$ bower install Polymer/core-icon-button
```

IMPORT

```
<head>
  <link rel="import" href="core-toolbar.html">
  <link rel="import" href="core-icon-button.html">
</head>
```

USE

```
<core-toolbar>
  <core-icon-button icon="menu"></core-icon-button>
  <span flex>Toolbar</span>
    <core-icon-button icon="refresh"></core-icon-button>
  <core-icon-button icon="add"></core-icon-button>
</core-toolbar>
```

USE

```
<core-toolbar>
  <core-icon-button icon="menu"></core-icon-button>
  <span flex>Toolbar</span>
    <core-icon-button icon="refresh"></core-icon-button>
  <core-icon-button icon="add"></core-icon-button>
</core-toolbar>
```

Browser support

Winter 2015

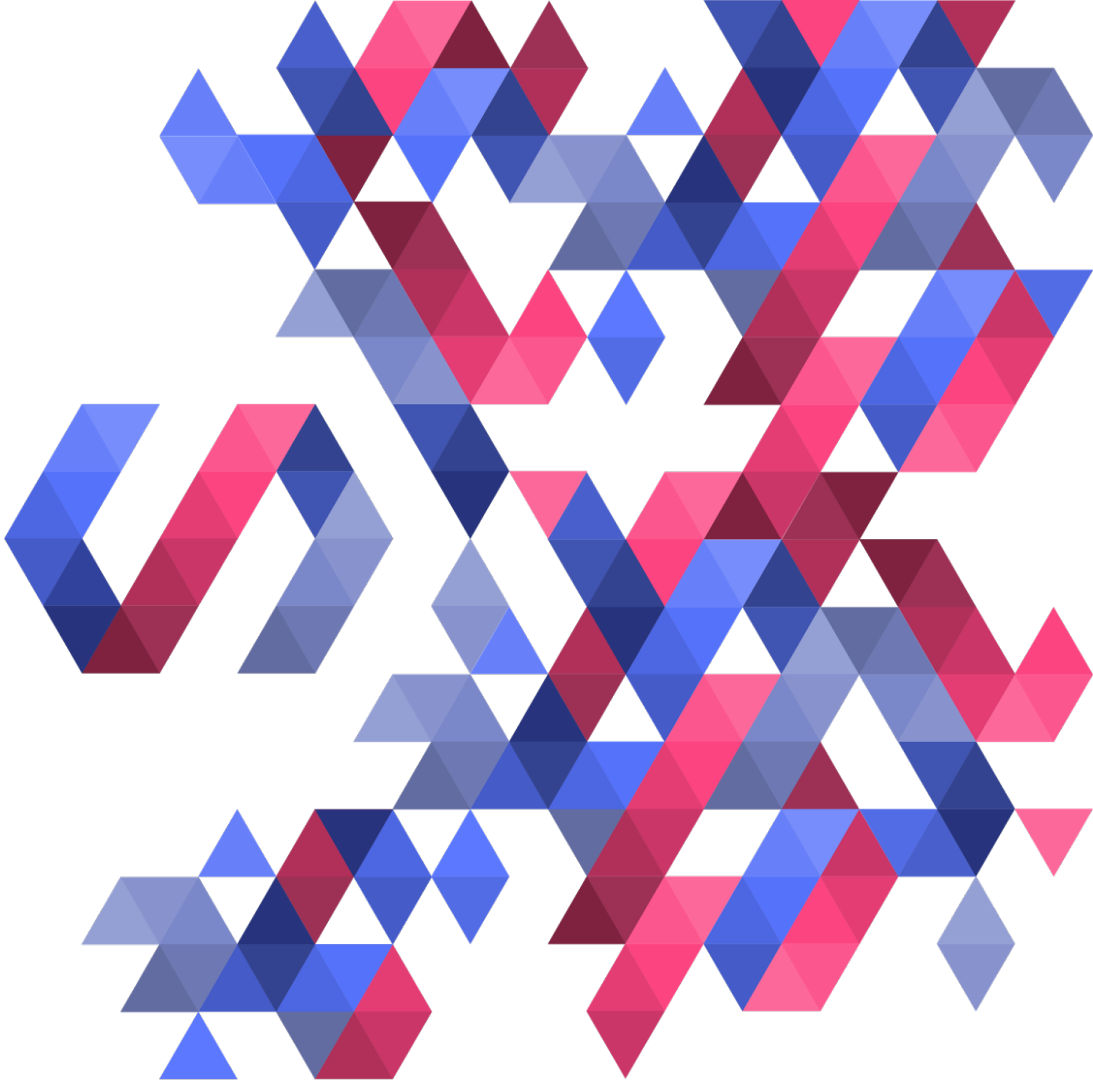
Templates

Custom Elements

Shadow DOM

HTML Imports

polymer



Incrementa Web Components com `webcomponents.js`*

* anteriormente chamado `platform.js`

Adiciona “syntactic sugar” com `polymer.js`

Facilita o entendimento

Suporte a browsers

Inverno 2015 (com Polymer)

Templates

Custom Elements

Shadow DOM

HTML Imports

Suporte a browsers

Inverno 2015 (com Polymer)

Templates

Custom Elements

Shadow DOM

HTML Imports

Sugaring: Custom Elements

vanilla

```
document.registerElement('paper-tabs', {
  prototype: Object.create(HTMLElement.prototype)
});
```

polymer

```
<polymer-element name="paper-tabs">
    …
</polymer-element>
```

usage

```
<paper-tabs>…</paper-tabs>
// document.createElement('paper-tabs');
```

vanilla

```
document.registerElement('paper-tabs', {
  prototype: Object.create(HTMLElement.prototype)
});
```

polymer

```
<polymer-element name="paper-tabs">
    …
</polymer-element>
```

usage

```
<paper-tabs>…</paper-tabs>
// document.createElement('paper-tabs');
```

vanilla

```
document.registerElement('paper-tabs', {
  prototype: Object.create(HTMLElement.prototype)
});
```

polymer

```
<polymer-element name="paper-tabs">
    …
</polymer-element>
```

usage

```
<paper-tabs>…</paper-tabs>
// document.createElement('paper-tabs');
```

vanilla

```
document.registerElement('paper-tabs', {
  prototype: Object.create(HTMLElement.prototype)
});
```

polymer

```
<polymer-element name="paper-tabs">
    …
</polymer-element>
```

usage

```
<paper-tabs>…</paper-tabs>
// document.createElement('paper-tabs');
```

vanilla

```
document.registerElement('super-button', {
  prototype: Object.create(HTMLButtonElement.prototype),
  extends: 'button'
});
```

polymer

```
<polymer-element name="super-button" extends="button">
    …
</polymer-element>
```

usage

```
<button is="super-button">…</button>
// document.createElement('button', 'super-button');
```

vanilla

```
document.registerElement('super-button', {
  prototype: Object.create(HTMLButtonElement.prototype),
  extends: 'button'
});
```

polymer

```
<polymer-element name="super-button" extends="button">
    …
</polymer-element>
```

usage

```
<button is="super-button">…</button>
// document.createElement('button', 'super-button');
```

vanilla

```
document.registerElement('super-button', {
  prototype: Object.create(HTMLButtonElement.prototype),
  extends: 'button'
});
```

polymer

```
<polymer-element name="super-button" extends="button">
    …
</polymer-element>
```

usage

```
<button is="super-button">…</button>
// document.createElement('button', 'super-button');
```

vanilla

```
document.registerElement('super-button', {
  prototype: Object.create(HTMLButtonElement.prototype),
  extends: 'button'
});
```

polymer

```
<polymer-element name="super-button" extends="button">
    …
</polymer-element>
```

usage

```
<button is="super-button">…</button>
// document.createElement('button', 'super-button');
```

Sugaring: Templates

vanilla

```
<template>
    …
</template>
```

polymer

```
<polymer-element name="user-list" noscript>
  <template>
    <ul>
      <template repeat="{{user, i in users}}">
        <li>{{user.name}}</li>
      </template>
    </ul>
  </template>
</polymer-element>
```

vanilla

```
<template>
    …
</template>
```


polymer

```
<polymer-element name="user-list" noscript>
  <template>
    <ul>
      <template repeat="{{user, i in users}}">
        <li>{{user.name}}</li>
      </template>
    </ul>
  </template>
</polymer-element>
```

vanilla

```
<template>
    …
</template>
```

polymer

```
<polymer-element name="user-list" noscript>
  <template>
    <ul>
      <template repeat="{{user, i in users}}">
        <li>{{user.name}}</li>
      </template>
    </ul>
  </template>
</polymer-element>
```

Sugaring: Shadow DOM

vanilla

```
var shadow = el.createShadowRoot();
shadow.innerHTML = “<style>h2 { color: red; }</style>” +
                   “<h2>I’m a profile-card</h2>”;
```


polymer

```
<polymer-element name=“profile-card” noscript>
  <template>
    <link rel=“stylesheet” href=“styles.css”>
    <h2>I’m a profile-card</h2>
  </template>
</polymer-element>
```

vanilla

```
var shadow = el.createShadowRoot();
shadow.innerHTML = “<style>h2 { color: red; }</style>” +
                   “<h2>I’m a profile-card</h2>”;
```


polymer

```
<polymer-element name=“profile-card” noscript>
  <template>
    <link rel=“stylesheet” href=“styles.css”>
    <h2>I’m a profile-card</h2>
  </template>
</polymer-element>
```

vanilla

```
var shadow = el.createShadowRoot();
shadow.innerHTML = “<style>h2 { color: red; }</style>” +
                   “<h2>I’m a profile-card</h2>”;
```

polymer

```
<polymer-element name=“profile-card” noscript>
  <template>
    <link rel=“stylesheet” href=“styles.css”>
    <h2>I’m a profile-card</h2>
  </template>
</polymer-element>
```

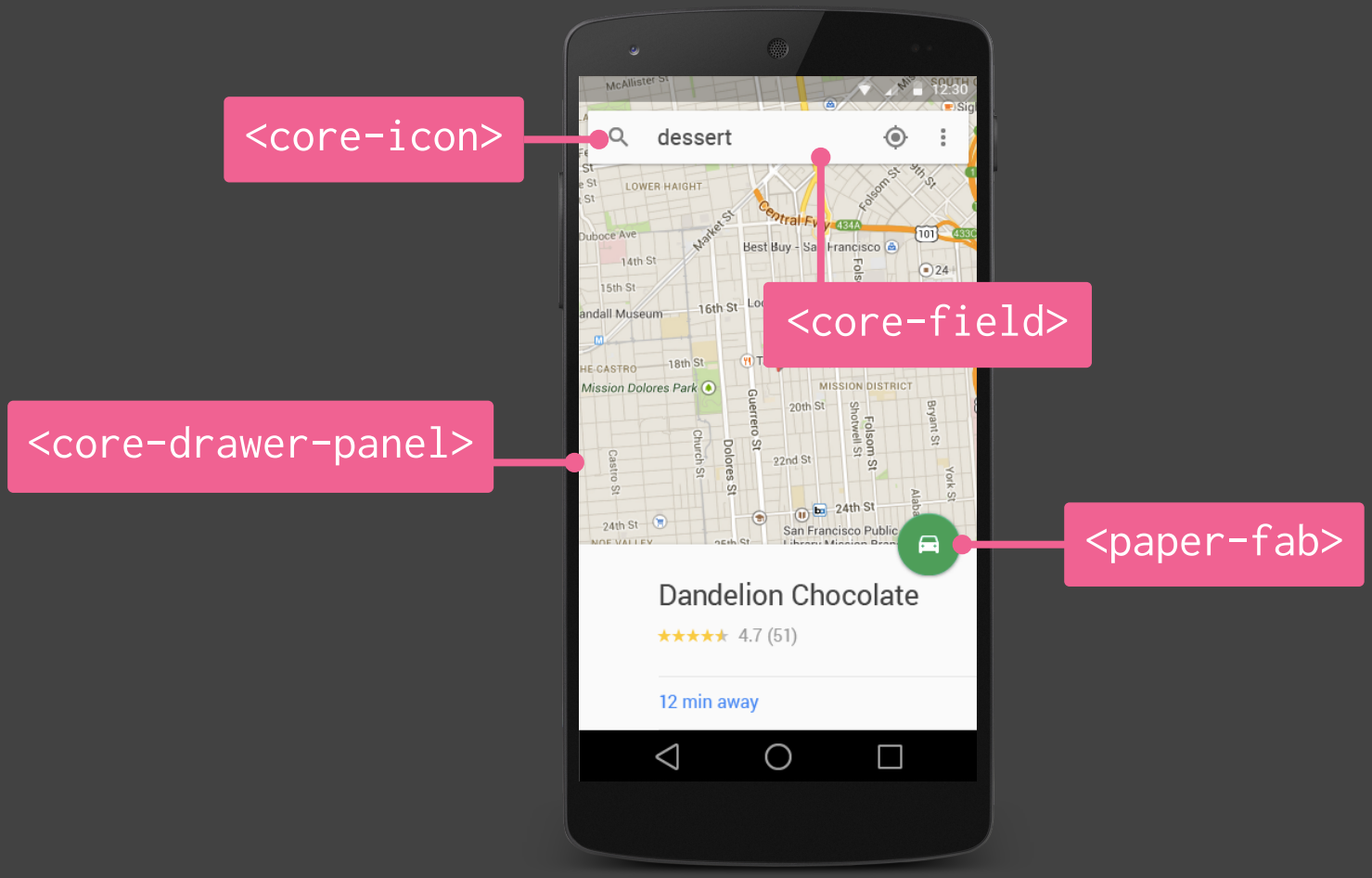
Components

<h1>

<p>

<animated-pages>
<menu-button>
<page-scaffold>

E se nós desenvolvessemos HTML para web mobile?


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

core-elements

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

<core-toolbar>

Um container basíco para controlar tabs ou botões

<core-toolbar>

Um container basíco para controlar tabs ou botões

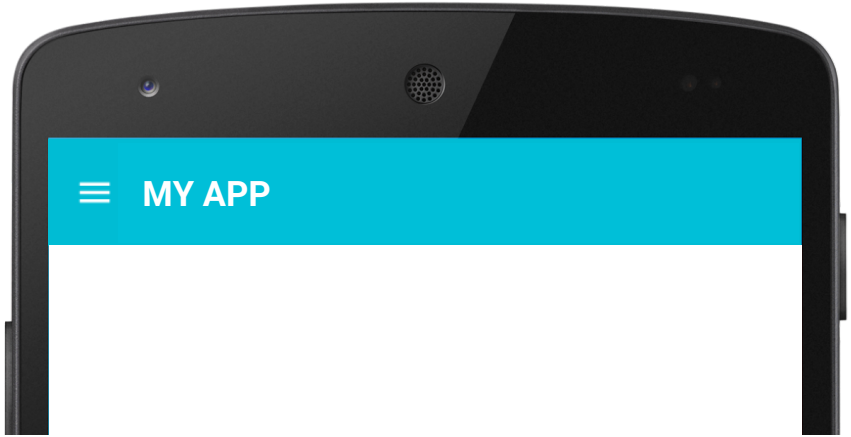
```
<link rel="import"
      href="core-toolbar.html">
```

<core-toolbar>

Um container basíco para controlar tabs ou botões

```
<link rel="import"
      href="core-toolbar.html">
```

```
<core-toolbar>
  <div>MY APP</div>
</core-toolbar>
```



<core-toolbar>

Um container basíco para controlar tabs ou botões

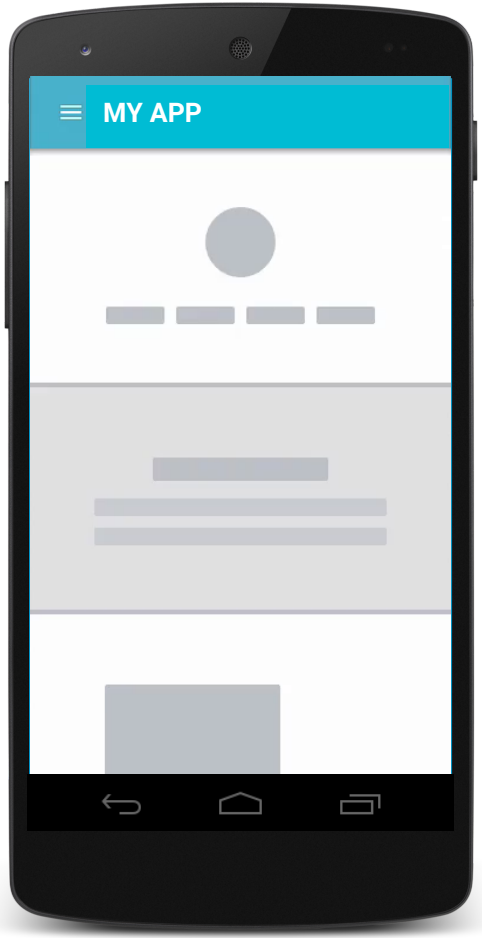
```
<link rel="import"
      href="core-toolbar.html">
```

```
<core-toolbar>
  <core-icon-button icon="menu">
  </core-icon-button>
  <div>MY APP</div>
</core-toolbar>
```

<core-header-panel>

Um container simples com uma seção de cabeçalho e outra de conteúdo

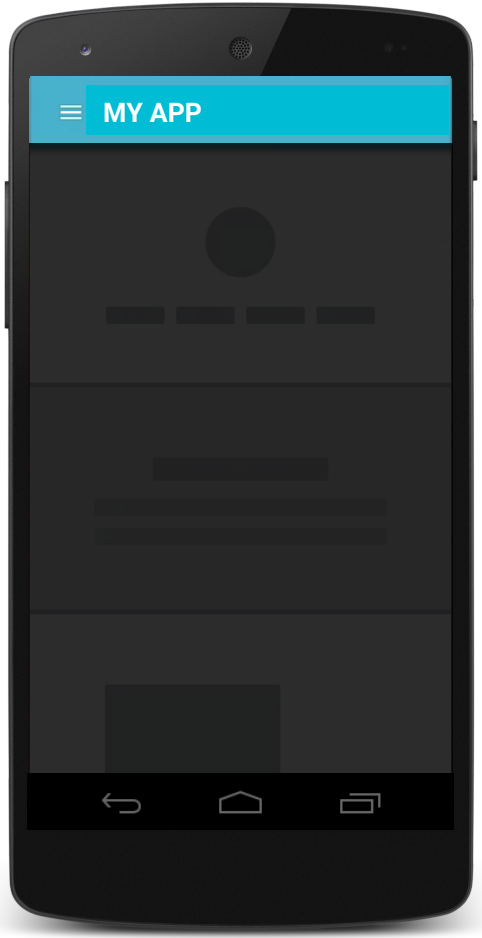
```
<core-header-panel flex>
  <core-toolbar>
    <core-icon-button icon="menu">
    </core-icon-button>
    <div>MY APP</div>
  </core-toolbar>
  <div class="content">…</div>
</core-header-panel>
```



<core-header-panel>

Um container simples com uma seção de cabeçalho e outra de conteúdo

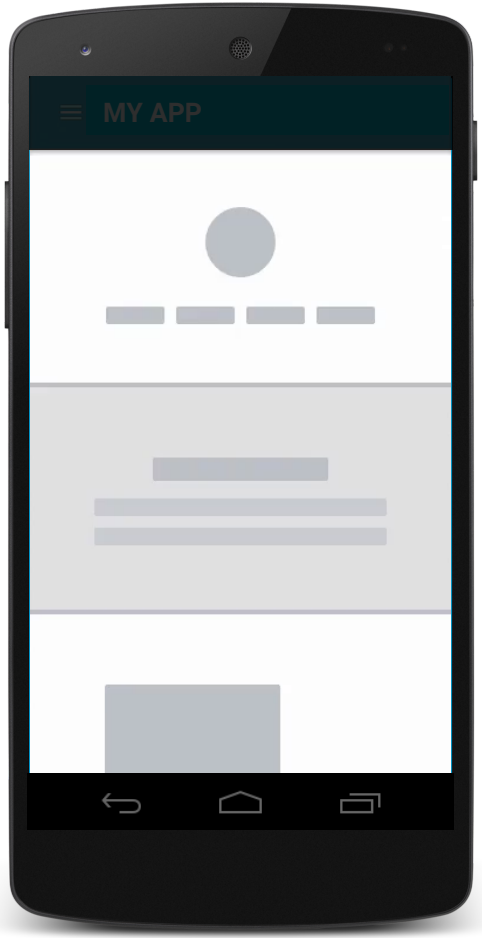
```
<core-header-panel flex>
  <core-toolbar>
    <core-icon-button icon="menu">
    </core-icon-button>
    <div>MY APP</div>
  </core-toolbar>
  <div class="content">…</div>
</core-header-panel>
```



<core-header-panel>

Um container simples com uma seção de cabeçalho e outra de conteúdo

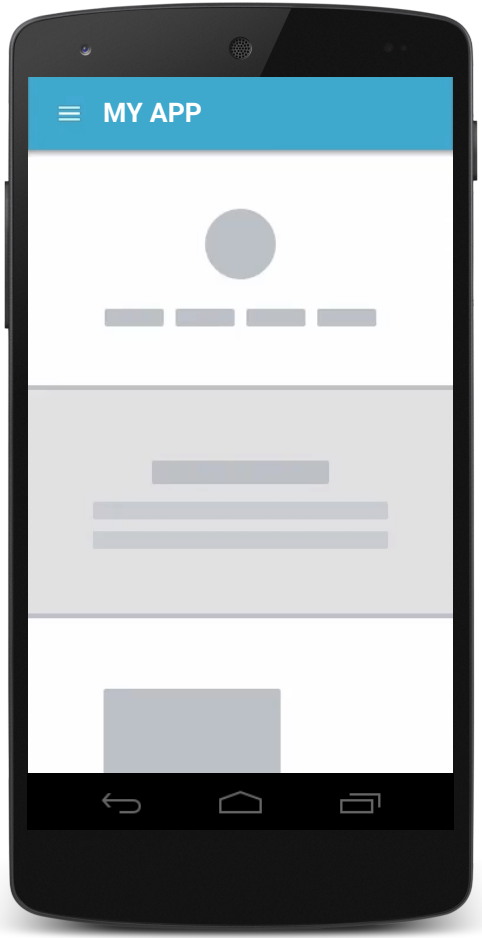
```
<core-header-panel flex>
  <core-toolbar>
    <core-icon-button icon="menu">
    </core-icon-button>
    <div>MY APP</div>
  </core-toolbar>
  <div class="content">…</div>
</core-header-panel>
```



<core-header-panel>

Um container simples com uma seção de cabeçalho e outra de conteúdo

```
<core-header-panel flex>
  <core-toolbar>
    <core-icon-button icon="menu">
    </core-icon-button>
    <div>MY APP</div>
  </core-toolbar>
  <div class="content">…</div>
</core-header-panel>
```



<core-header-panel>

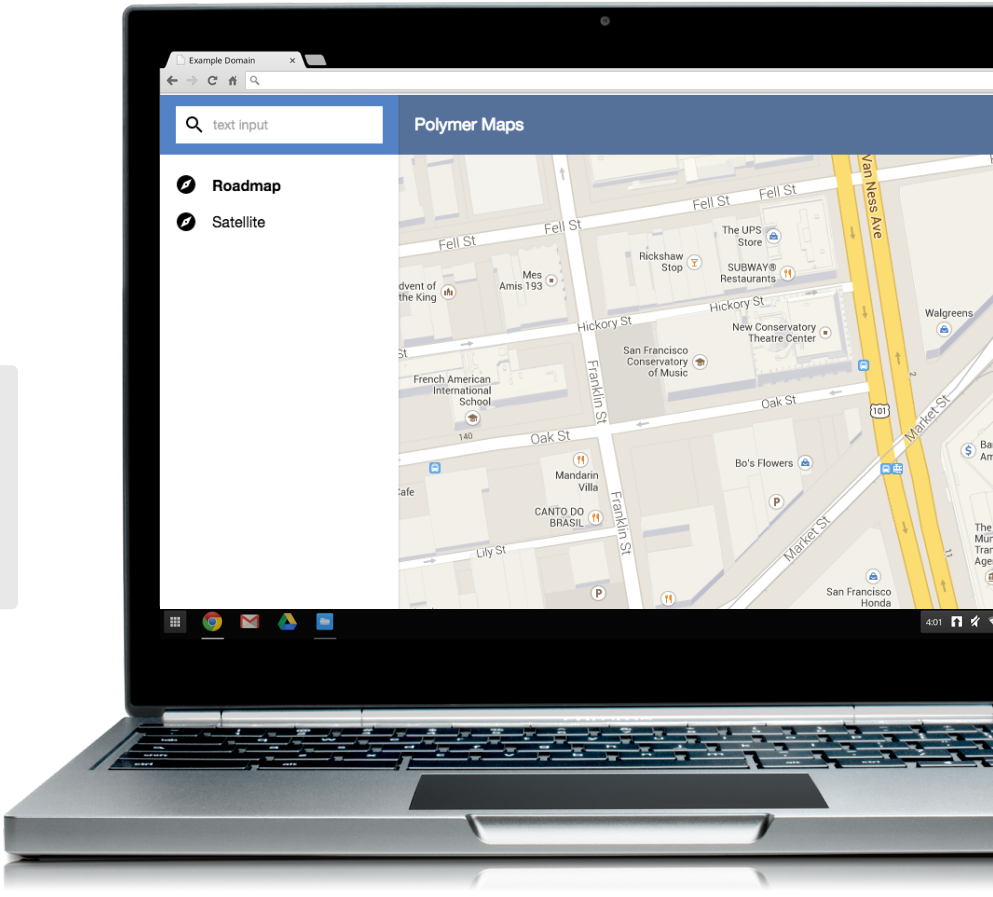
Toolbar vai rolar com a página

```
<core-header-panel mode="scroll" flex>
  <core-toolbar>
    <core-icon-button icon="menu">
    </core-icon-button>
    <div>MY APP</div>
  </core-toolbar>
  <div class="content">…</div>
</core-header-panel>
```

<core-drawer-panel>

Um container **responsivo** que combina um drawer panel e uma área para o conteúdo principal.

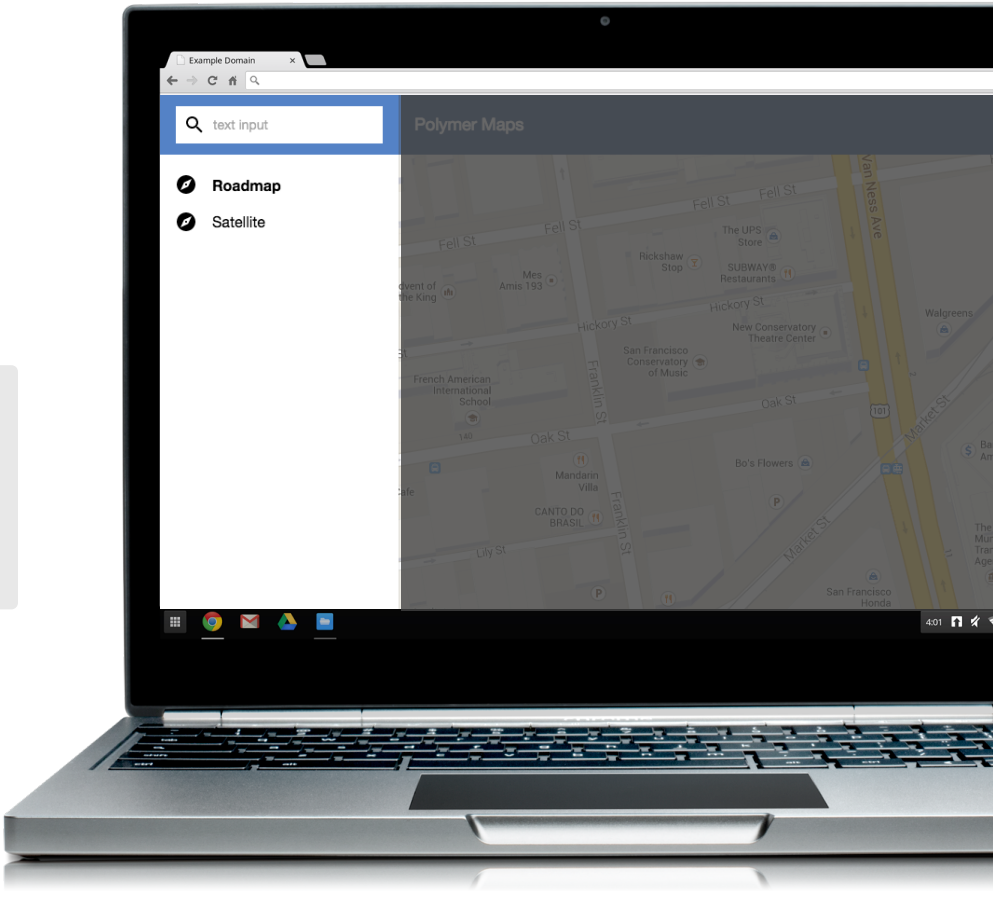
```
<core-drawer-panel>
  <div drawer> Drawer panel... </div>
  <div main> Main panel... </div>
</core-drawer-panel>
```



<core-drawer-panel>

Um container **responsivo** que combina um drawer panel e uma área para o conteúdo principal.

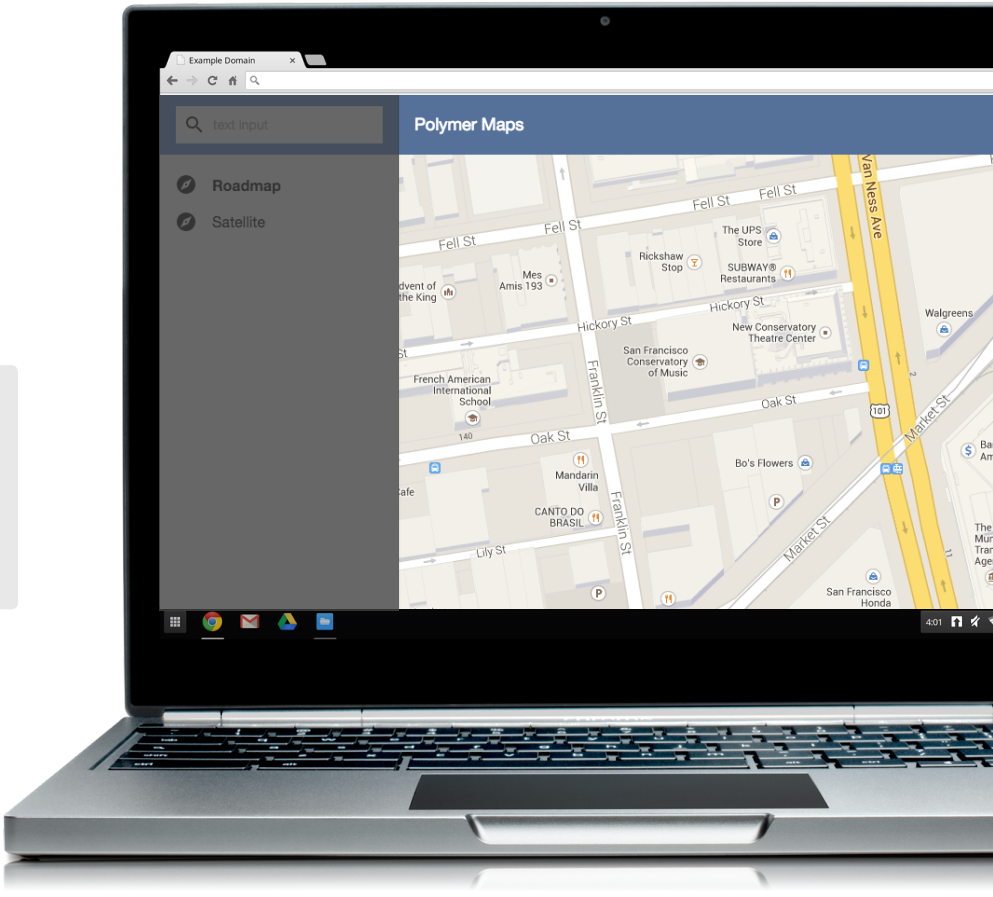
```
<core-drawer-panel>
  <div drawer> Drawer panel... </div>
  <div main> Main panel... </div>
</core-drawer-panel>
```

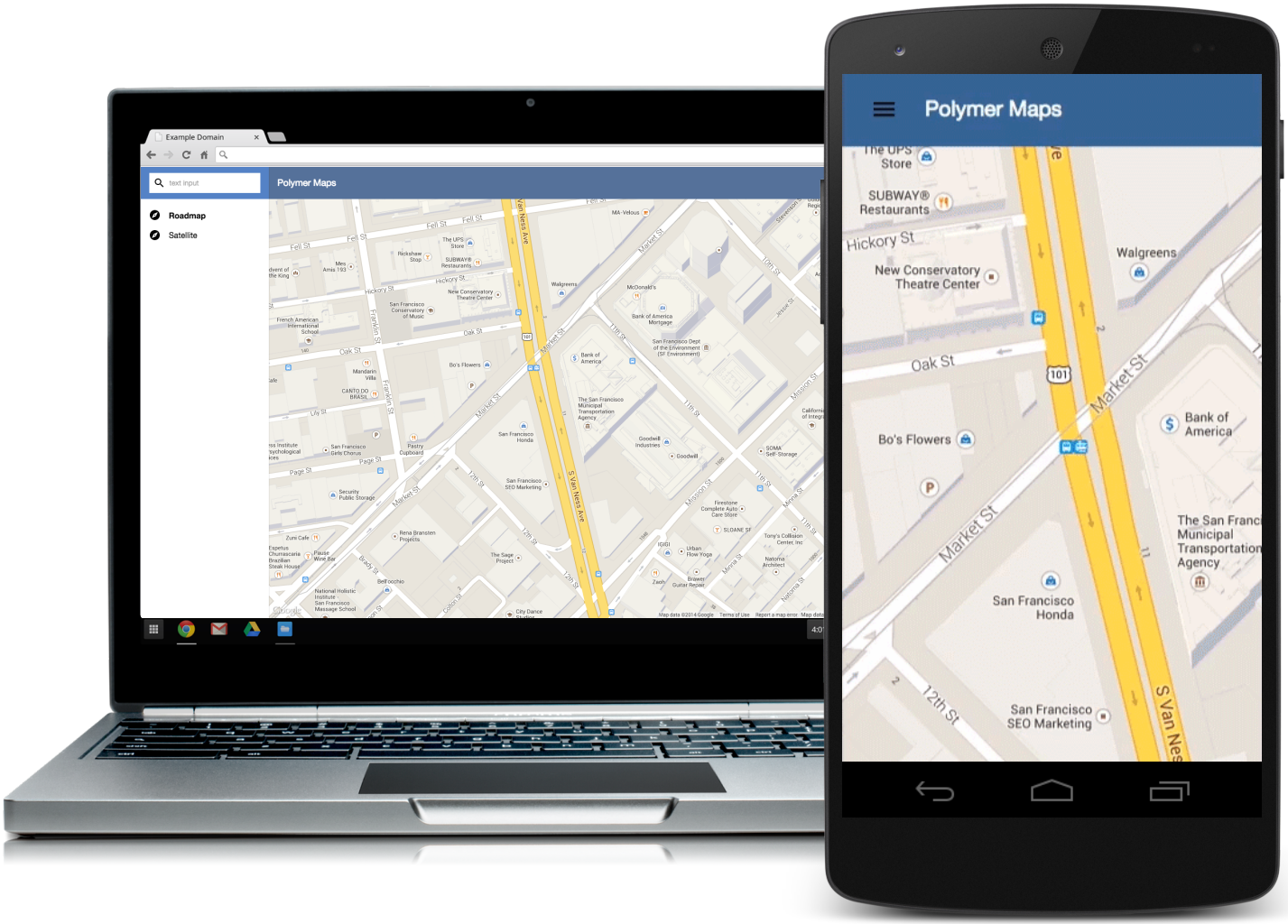


<core-drawer-panel>

Um container **responsivo** que combina um drawer panel e uma área para o conteúdo principal.

```
<core-drawer-panel>
  <div drawer> Drawer panel... </div>
  <div main> Main panel... </div>
</core-drawer-panel>
```



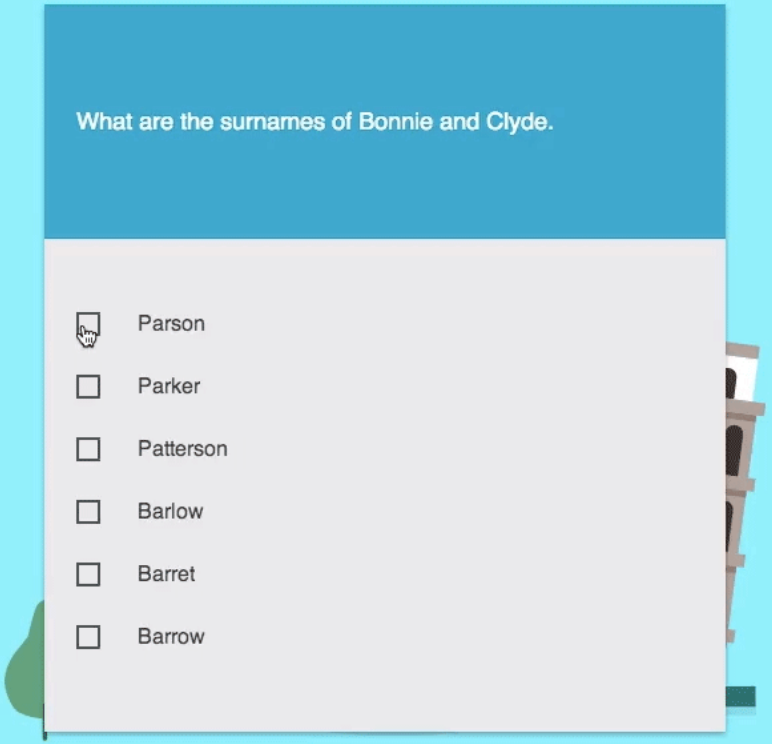

Example Domain
text input
Polymer Maps
Roadmap
Satellite
Polymer Maps
The UPS Store
SUBWAY® Restaurants
Hickory St
New Conservatory Theatre Center
Walgreens
Oak St
101
Market St
Bank of America
Bo's Flowers
The San Francisco Municipal Transportation Agency
San Francisco Honda
12th St
San Francisco SEO Marketing
S Van Ness

paper-elements

Type only numbers... (floating)

```
<paper-input floatinglabel
  label="Type only numbers... (floating)"
  validate="^[0-9]*$"
  error="Input is not a number!">
</paper-input>
```

```
<paper-checkbox></paper-checkbox>
```



<paper-ripple>

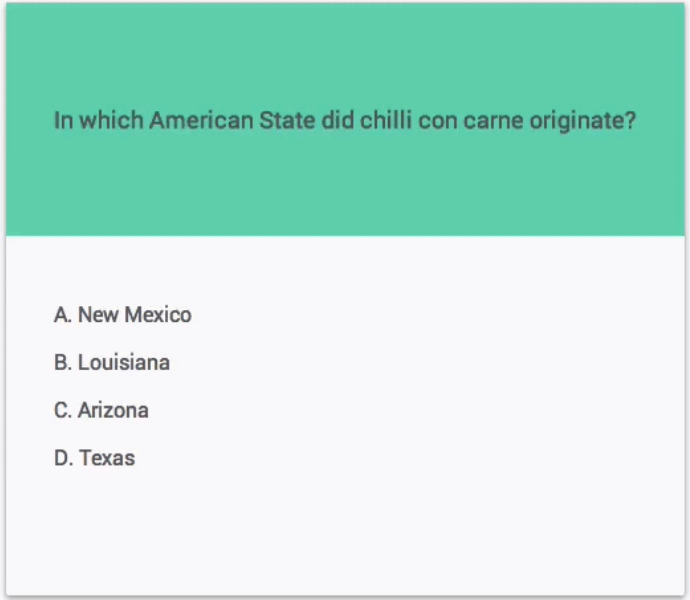
Um efeito para indicar toque ou ações do mouse

```
<div class=“card”>
  <img src=“science.svg”>
  <paper-ripple fit></paper-ripple>
</div>
```

<paper-shadow>

Uma sombra dinâmica para carregar sombra e relações espaciais

```
<paper-shadow z="5" animated>
  <div class="card">...</div>
</paper-shadow>
```



Estilo

::shadow

Permite a você dar um estilo interno ao elemento da sombra

```
<paper-slider min=“0” max=“100”>
</paper-slider>
```

::shadow

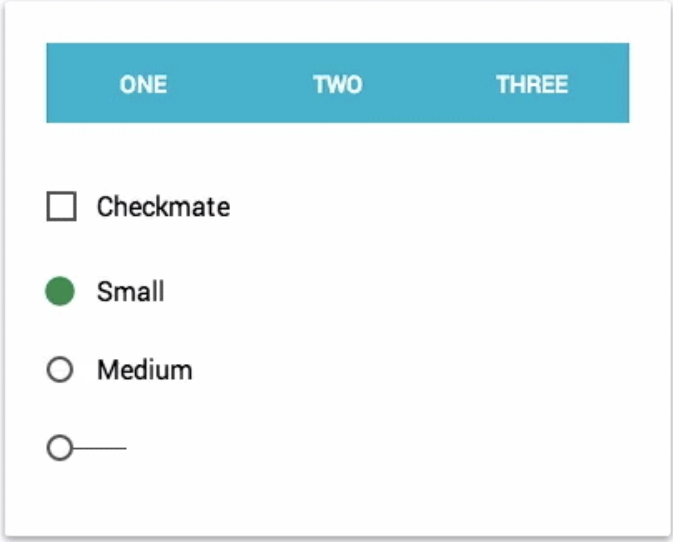
Permite a você dar um estilo interno ao elemento da sombra

```
<paper-slider min="0" max="100">
</paper-slider>
```

```
paper-slider::shadow #sliderKnobInner {
  background-color: #f4b400;
}
```

/deep/

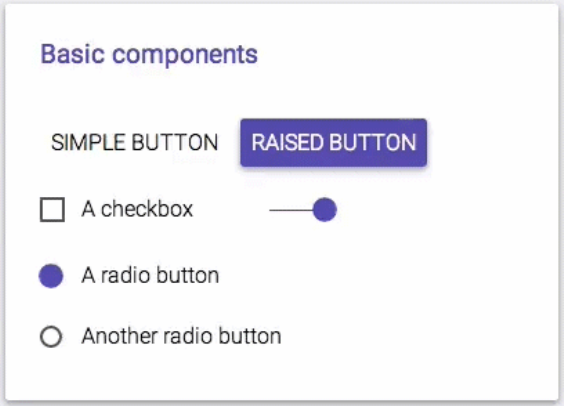
Estilo vai sobrescrever todas as sombras

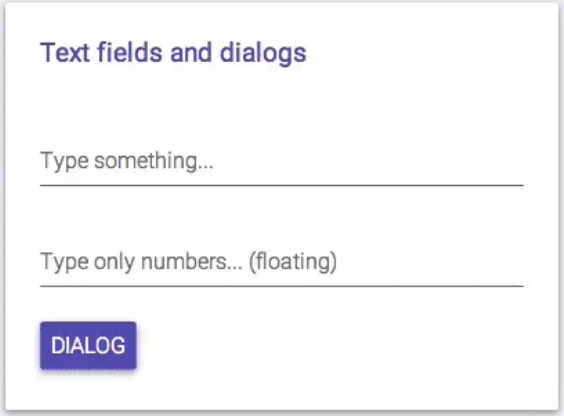


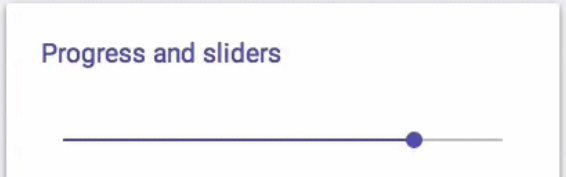

```
html /deep/ paper-ripple {
  background-color: #E91E63;
}
```

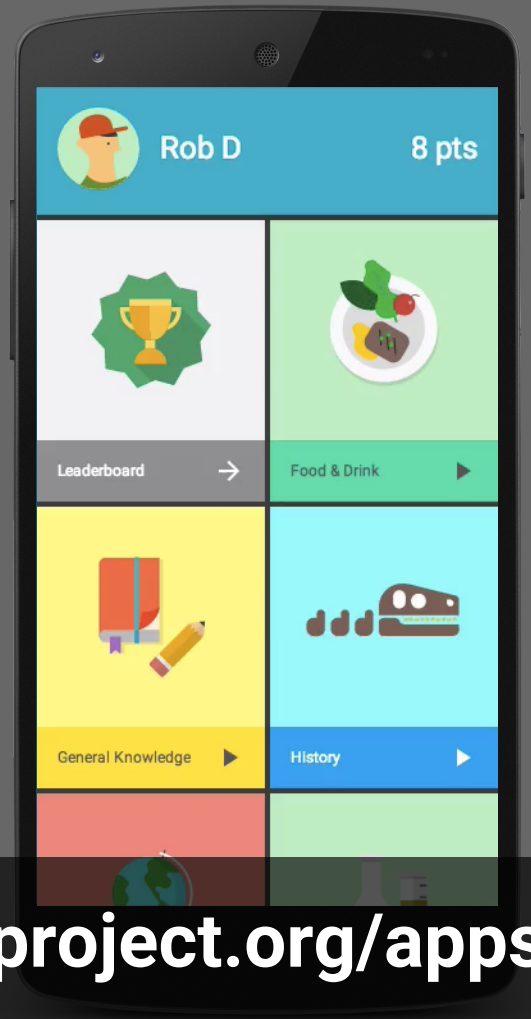
Com **::shadow** e **/deep/** você pode aplicar temas a todo site

source: ebidel.github.io/material-playground

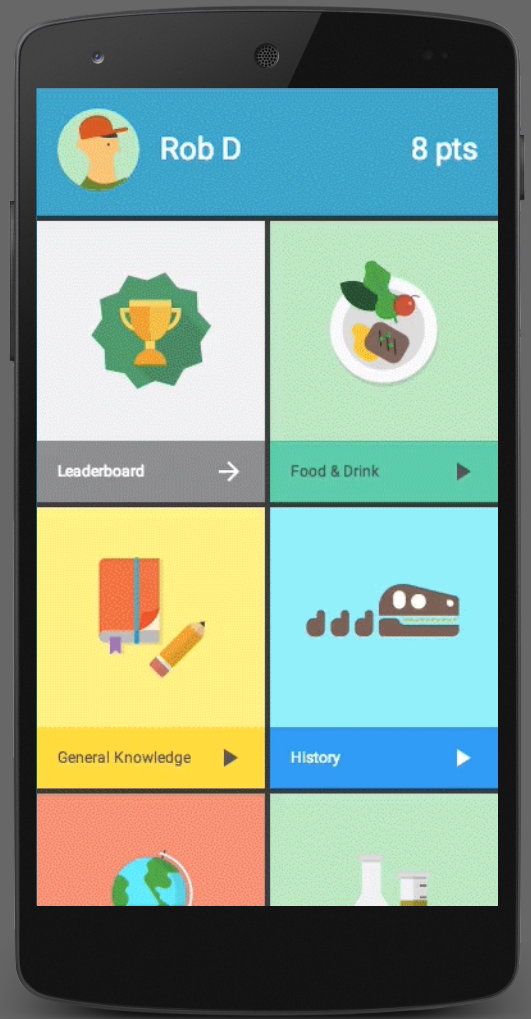




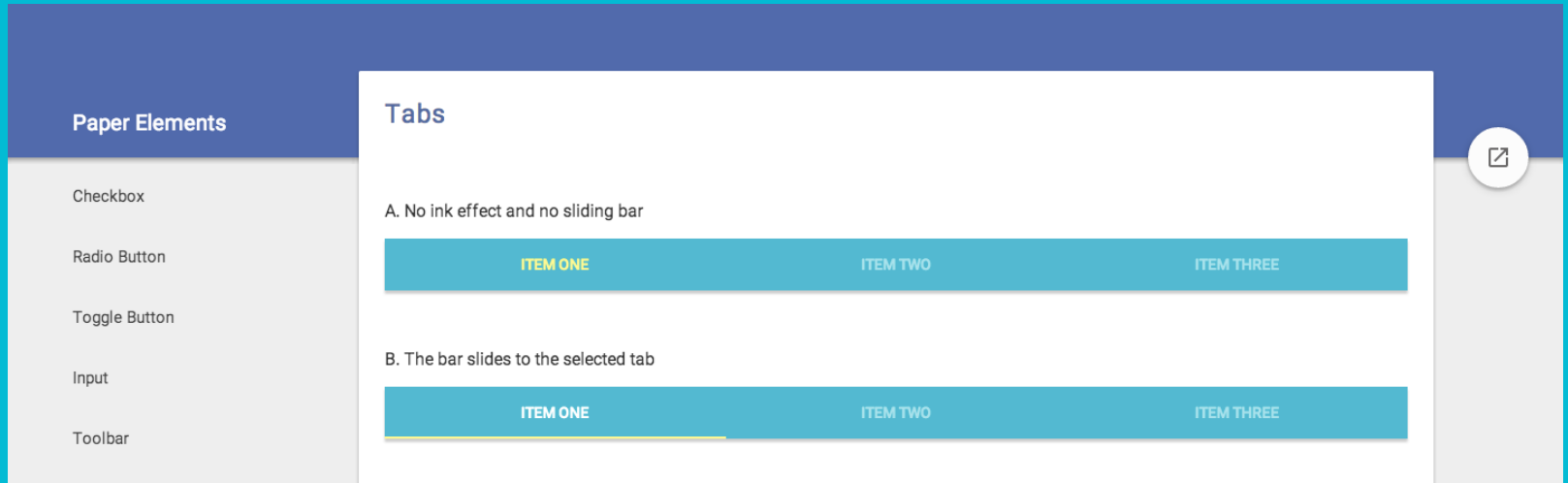




polymer-project.org/apps/topeka/


Rob D
8 pts
Leaderboard
Food & Drink
General Knowledge
History

polymer-project.org



Apps

polymer

Learn Elements Resources

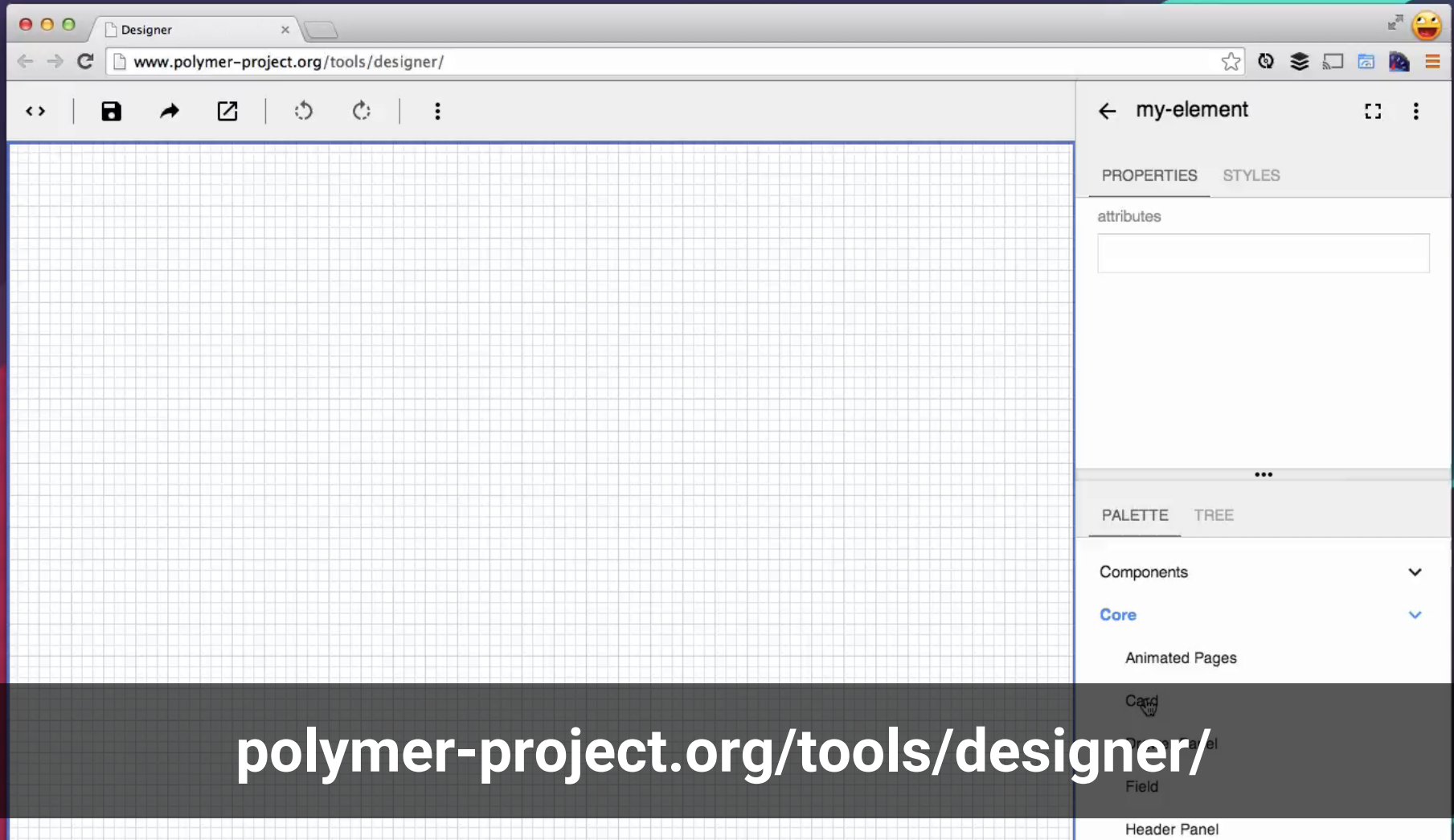
Welcome to the future

Web Components usher in a new era of web development based on encapsulated and interoperable custom elements that extend HTML itself. Built atop these new standards, Polymer makes it easier and faster to create anything from a button to a complete application across desktop, mobile, and beyond.

GET POLYMER VIEW ON GITHUB

Use Elements (30 sec) ➜ Create Elements (5 min) ➜ Build an app (30 min)

polymer-project.org


Designer
www.polymer-project.org/tools/designer/
my-element
PROPERTIES
STYLES
attributes
PALETTE
TREE
Components
Core
Animated Pages
Card
Field
Header Panel
polymer-project.org/tools/designer/

APIs

APIs (as elements)

“Eu quero adicionar um marcador ao Google map.”

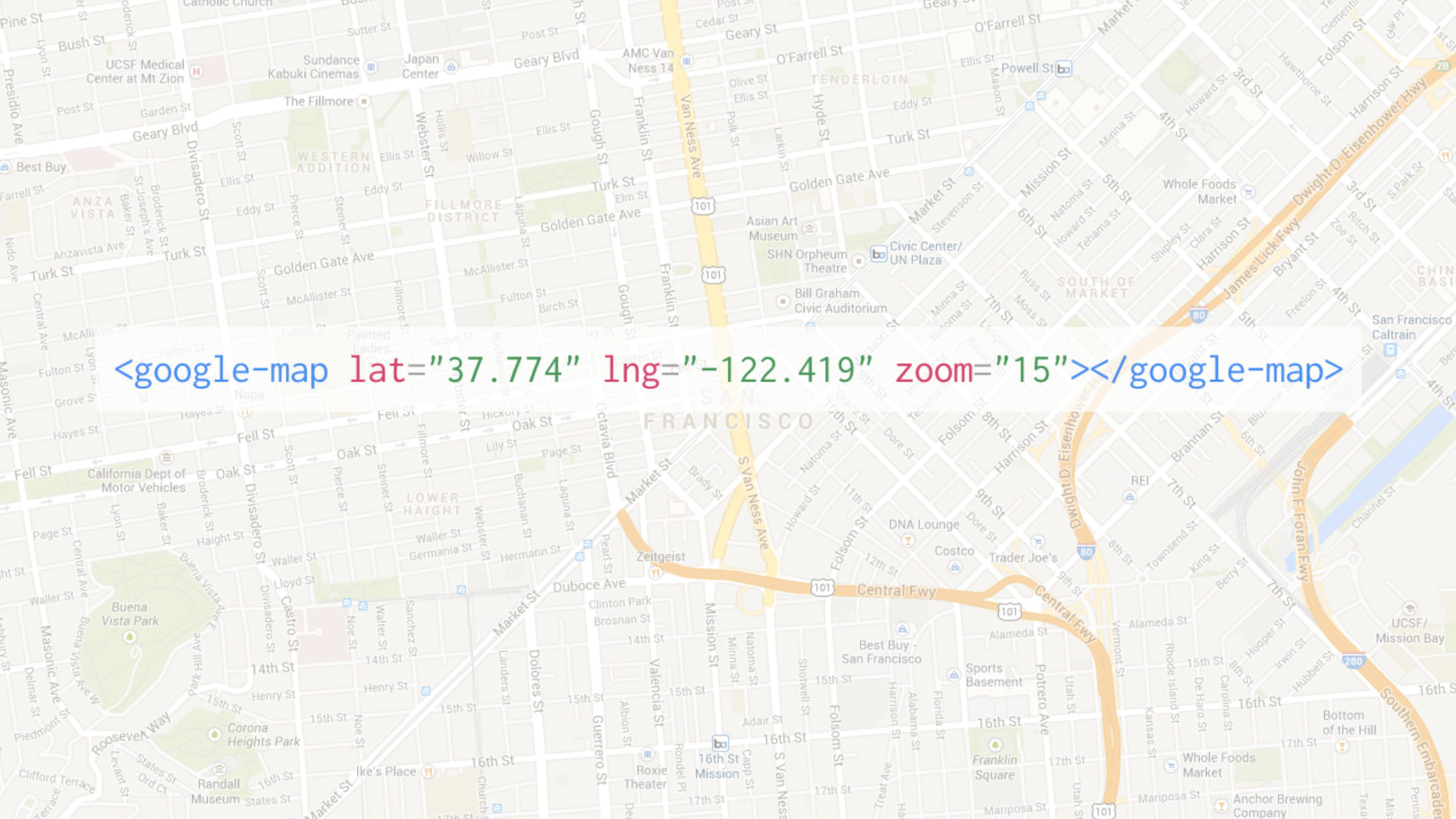
```
  }
</style>

<div id="map"></div>

<script src="http://maps.googleapis.com/maps/api/js?callback=mapReady">
</script>
<script>
  var marker = null;

  function getCurrentLocation(callback) {
    navigator.geolocation.watchPosition(callback);
  }

  function addMarker(opts, info) {
    var marker = new google.maps.Marker(opts);

    var infoWindow = new google.maps.InfoWindow({content: info});

    google.maps.event.addListener(marker, 'click', function() {
      infoWindow.open(opts.map, marker);
    });

    return marker;
  }

  function mapReady() {
    var container = document.querySelector('#map');
    var map = new google.maps.Map(container, {
      zoom: 14, disableDefaultUI: true
    });

    getCurrentLocation(function(pos) {
      var current = new google.maps.LatLng(pos.coords.latitude,
                                           pos.coords.longitude);
      map.setCenter(current);

      // Re-position marker or create new one.
      if (marker) {
        marker.setPosition(map.getCenter());
      } else {
        marker = addMarker({
          position: current, map: map, title: 'Your location'
        }, '<b>Your location</b>');
      }
```

Tanto código para um marcador

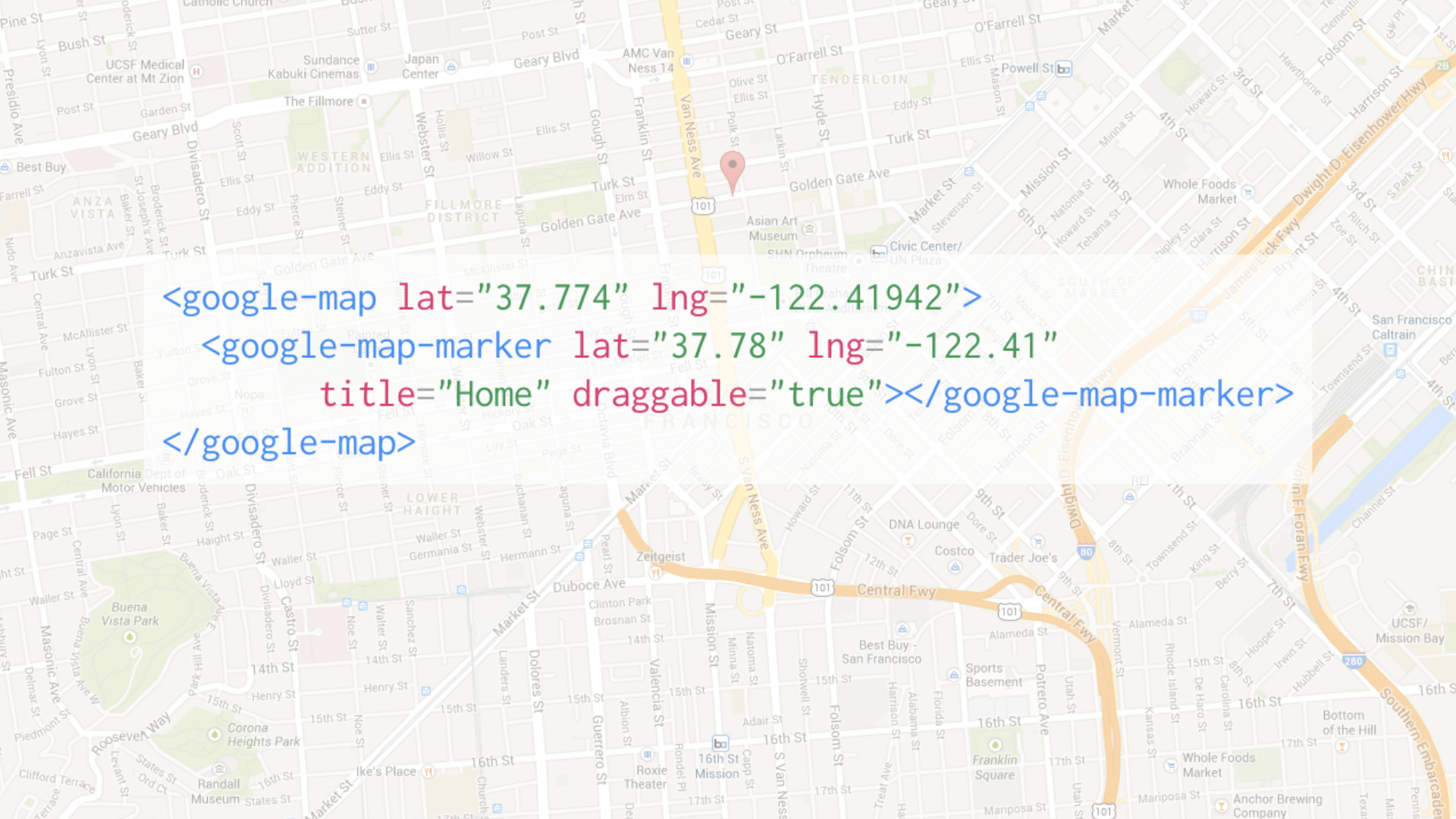
```
<google-map></google-map>
```


<google-map></google-map>
San Jose
Cupertino
Fruitdale
Evergreen
Campbell
South San Jose
Saratoga
Los Gatos
Edenvale
Almaden Quicksilver County Park
Morgan Hill
Henry W. Coe State Park
San Martin
Coyote Lake - Harvey Bear Ranch County Park
The Forest of Nisene Marks State Park
Scotts Valley
Soquel
Aptos
Soquel Cove
Santa Cruz
Gilroy
Freedom
Watsonville
Monterey Bay
Hollister
Prunedale
Paicines
Marina
Salinas
Monterey
Fort Ord National Monument
Pebble Beach
Carmel Bay
Toro County Park
Carmel Valley
Gonzales
Pinnacles National Park
Newman
Merced
San Luis National Wildlife Refuge
Merced National Wildlife Refuge
Chowchilla
Volta
O'Neill Forebay
San Luis Reservoir
Pacheco State Park
Los Banos
Dos Palos
Firebaugh
Mendota


```
<google-map lat="37.774" lng="-122.419"></google-map>
```


<google-map lat="37.774" lng="-122.419" zoom="15"></google-map>


```
<google-map lat="37.774" lng="-122.41942">
  <google-map-marker lat="37.78" lng="-122.41"
      title="Home" draggable="true"></google-map-marker>
</google-map>
```


<google-map lat="37.774" lng="-122.41942">
<google-map-marker lat="37.78" lng="-122.41"
title="Home" draggable="true"></google-map-marker>
</google-map>

googlewebcomponents.github.io

github.com/GoogleWebComponents

Google Web Components

A collection of web components for Google APIs & services. Built with Polymer.

<google-analytics>

DEMO DOCS GITHUB

```
bower install GoogleWebComponents/google-analytics [--save]
```

<google-apis>

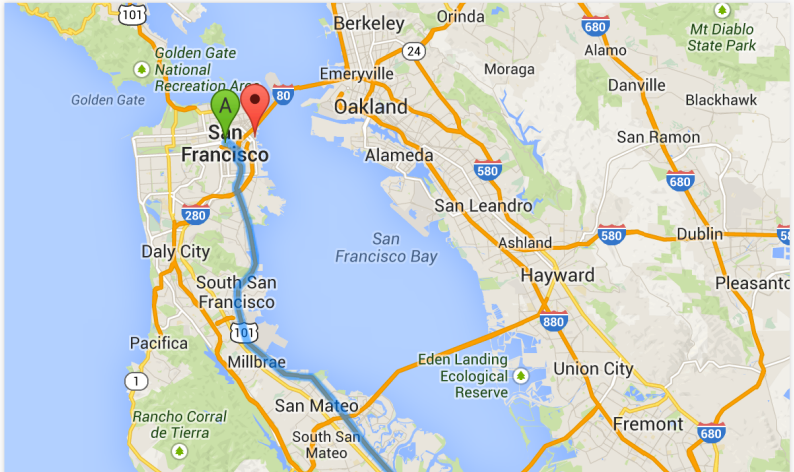
DEMO DOCS GITHUB

```
bower install GoogleWebComponents/google-apis [--save]
```

<google-calendar>

DEMO DOCS GITHUB

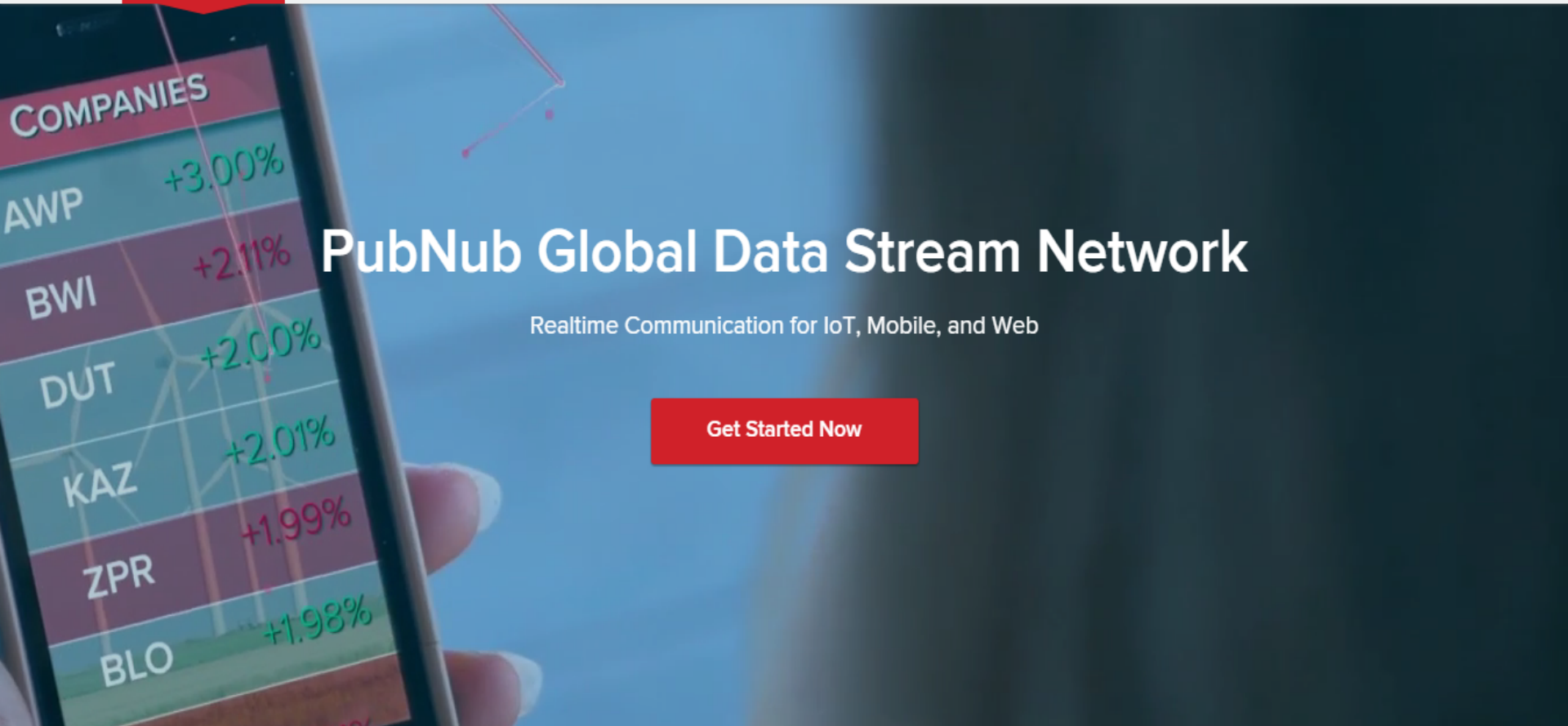
```
bower install GoogleWebComponents/google-calendar [--save]
```



I/O Bytes
Develop

youtube.com/watch?v=eORqFaf_QzM

PubNub



PubNub Global Data Stream Network

Realtime Communication for IoT, Mobile, and Web

Get Started Now

Mão na Massa

Junte-se a revolução

Aprenda

WebComponents.org

a place to discuss and evolve web component best-practices

WHAT?

WebComponents.org is where pioneers and community-members of the Web Components ecosystem (like *Polymer*, *X-tags*, and other interested parties) document web components best practices so that others can follow the same path instead of needlessly striking out on their own.

BROWSER SUPPORT

	CHROME	OPERA	FIREFOX	SAFARI	IE
⚙					

SPECS

WEB COMPONENTS

This document is a non-normative reference, which provides an overview of Web Components. It summarizes the normative information in the respective specifications in easy-to-digest prose with illustrations.

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.

ARTICLES

WEB COMPONENTS BEST PRACTICES

Web Components (WC) are a new set of web platform features that enable developers to build applications in a declarative, composable way. The following is an initial list of best practices we advocate component authors consider to ensure their elements are good citizens in the Web Component ecosystem.

Read More >

see all articles

polymer

Learn Elements Resources

Welcome to the future

Web Components usher in a new era of web development based on encapsulated and interoperable custom elements that extend HTML itself. Built atop these new standards, Polymer makes it easier and faster to create anything from a button to a complete application across desktop, mobile, and beyond.

GET POLYMER VIEW ON GITHUB

Use Elements (30 sec) ➜ Create Elements (5 min) ➜ Build an app (30 min)

polymer-project.org

polycasts

```
<link rel="import"
 href="bower_components/core-iconset-svg/core-iconset-svg.html">

<core-iconset-svg></core-iconset-svg>
```

goo.gl/Ji3WdW

Construa

Comece com <seed-element>

github.com/PolymerLabs/seed-element

PolymerLabs / seed-element

Polymer element boilerplate http://www.polymer-project.org/docs/start/reusableelements.html — Edit

56 commits | 2 branches | 0 releases | 7 contributors

branch: master | seed-element / +

Update Polymer to 0.3.4

addyosmani authored 4 days ago — latest commit 4b144241fd

tests	Fixed encoding of file	8 days ago
.bowerrc	Added trailing newline to file	8 days ago
README.md	Converted DOS file endings to unix	8 days ago
bower.json	Update Polymer to 0.3.4	4 days ago
demo.html	Add meta viewport tag	3 months ago
index.html	Add meta viewport tag	3 months ago
seed-element.css	remove hidden unicode char at beg of file, also add missing newline a…	2 months ago
seed-element.html	Update example docs with additional support pragmas	a month ago

Code
Issues 4
Pull Requests 0
Wiki
Pulse
Graphs
Settings

HTTPS clone URL

https://github.cc

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

README.md

seed-element

See the component page for more information.

Getting Started

I/O Bytes Develop

youtube.com/watch?v=2toYLLcoY14


Chrome Dev Editor
http://goo.gl/UjLvb2

Compartilhe!

Custom Elements

a web components gallery for modern web apps

Tweet 1,312 Like 1k +1 325 Star 309

What are Web Components?

Web Components are a collection of standards which are working their way through the W3C. They enable truly encapsulated and reusable components for the web. And if you think HTML5 changed the web, wait to see what Web Components will do.

For lots more information about it, including articles and presentations, check out webcomponents.org.

— Zeno Rocha, project lead.

Submit your own

Got a great idea for a custom element? Awesome! There are boilerplates for Polymer, X-Tag, and VanillaJS that you can fork and get up and running with a simple component.

When you're ready to go, submit it to the form below and it'll appear on this site for others to play and use!

https://github.com/me/my-element Submit

Latest elements

basic-synth

A basic 8-bit synthesizer bound to the <audio>-tag

yin-yang-cat

Most popular elements

flexible GIF playback

voice-elements

customelements.io

bower.json

```
{
  "name": "my-element",
  "version": "0.0.0",
  "description": "My awesome Custom Element",
  "license": "MIT",
  "keywords": [
    "web-components"
  ],
  "ignore": [
    "**/.*",
    "node_modules",
    "bower_components"
  ]
}
```


EXPLORE

<obrigado>