Web Components for non-idealists

What are web components?

- Custom HTML element
- Widgets

```
<fancy-card>
  <h3>Fridolin Frontend</h3>
  <p>Frontend developer</p>
</fancy-card>
```

Independent, importable

```
<link rel="import"
   href="components/fancy-card.html">
```

Part of the HTML standard

at some point, hopefully...

Extend other components

```
<super-fancy-card>
  <h3>Fridolin Frontend</h3>
  <p>Frontend developer</p>
</super-fancy-card>
```

Extend HTML elements

```
<link rel="import"
   href="components/fancy-button.html">

<button is="fancy-button">Save</button>
```

Should I care?

Yes, because

- You have solved the "widget problem" in every project differently.
- There are a lot of libraries which solve it (jQuery UI, Angular JS directives, ...).
- Now comes the **HTML standard solution** for it.

Yes, because

This future *has* to excite you:

1. Find nice and matching web component for your UI problem
2. Add it to your bower.json, install it
3. Import the element into your HTML
4. Use it in your HTML as it would be part of the HTML standard

Yes, because

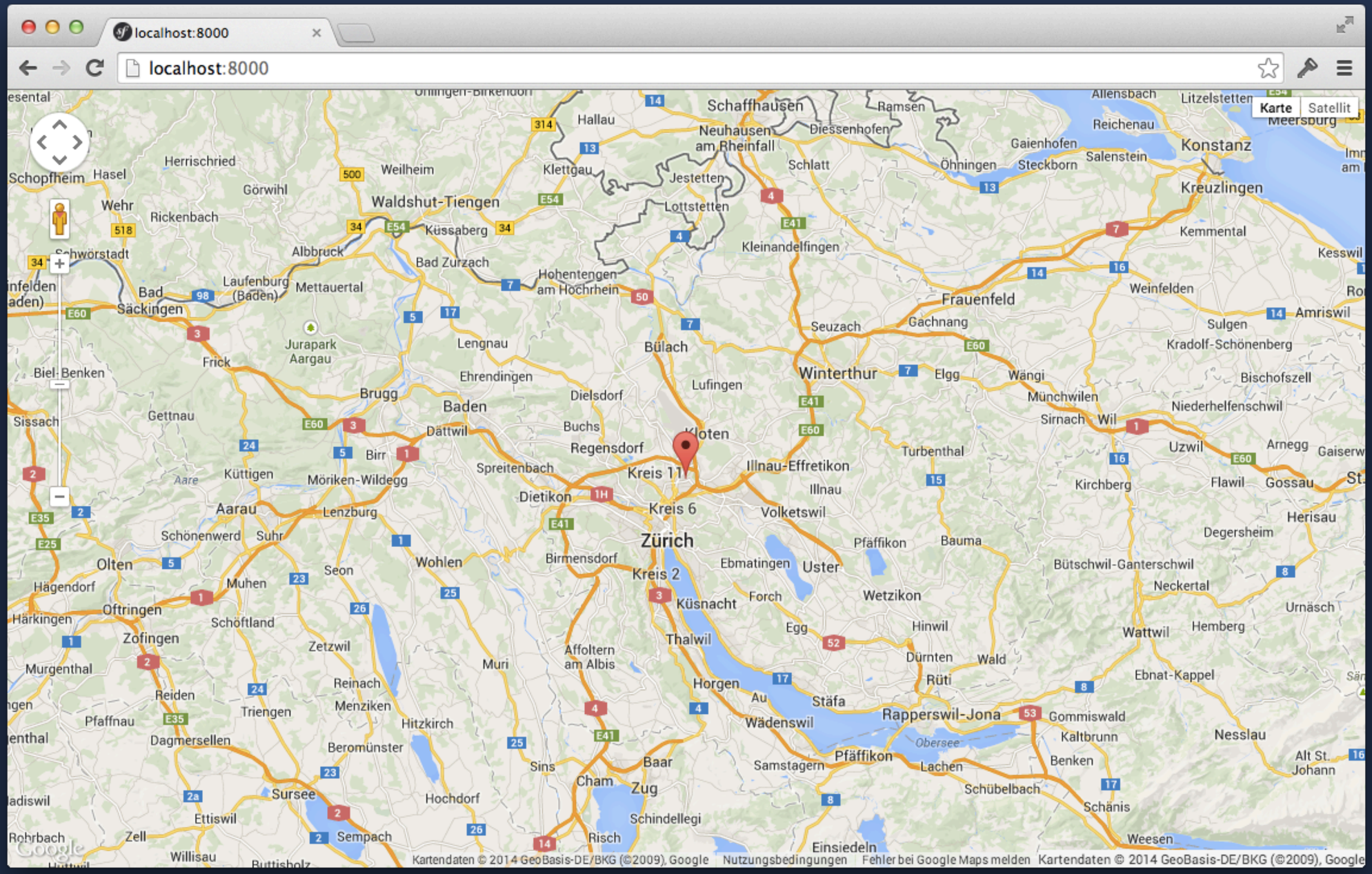
It will make your work easier, not harder. (Just a slight learning curve in the beginning, I promise ;)

Yes, because

You can start **now**. (Github uses it already in production :)

Example: Google maps

```
<google-map latitude="37.779"
    longitude="-122.3892">
  <google-map-marker latitude="37.779"
      longitude="-122.3892">
    <h3>Hoi!</h3>
  </google-map-marker>
</google-map>
```


localhost:8000
localhost:8000
Karte
Satellit
Zürich
Kreis 1
Kreis 2
Kreis 6
Winterthur
Baden
Brugg
Aarau
Bülach
Kloten
Dietikon
Uster
Wetzikon
Rapperswil-Jona
Konstanz
Schaffhausen
Waldshut-Tiengen
Bad Säckingen
Frauenfeld
Kreuzlingen
Kartendaten © 2014 GeoBasis-DE/BKG (©2009), Google
Nutzungsbedingungen
Fehler bei Google Maps melden

Example: Voice recognition

```
<voice-recognition id="recognition-element">

<script>
document.querySelector('#recognition-element')
   .addEventListener('result', function(e) {
     console.log(e.detail.result);
});
</script>
```

Example: Resizing iframe

```
<link rel="import"
  href="components/resizing-iframe.html">

<iframe is="resizing-iframe"
  src="http://www.swisstxt.ch/sport-results">
</iframe>
```

Browser support August 2014

- Chrome: Yep
- Firefox: Kind of
- Safari: Nope
- Internet Explorer: No way

jQuery for web components

Polyfill for web components, which allow web components for most of the browsers we support.
Only exception is IE9 which works partly. (IE10+ is mostly fine).

=> Test early but chances are good it works.

https://github.com/Polymer/platform

Libraries

- Polymer
- Mozilla Bricks
- X-Tags

Polymer

http://www.polymer-project.org/

- Polyfill: platform.js
- Data binding + some sugar
- core-elements
- paper-elements

Let's start

Setup

```
bower init

bower install --save Polymer/polymer

mkdir my-timeline
```

index.html

```
<html>
<head>
    <script src="bower_components/platform/platform.js"></script>
    <link rel="import" href="my-timeline/my-timeline.html">

</head>
<body>
    <my-timeline>
</body>
</html>
```

my-timeline/my-timeline.html

```
<link rel="import" href="../bower_components/polymer/polymer.html">

<polymer-element name="my-timeline">

  <template>
    <div>
        <div id="segment"></div>
    </div>
  </template>

  <script>
    Polymer('my-timeline')
  </script>

</polymer-element>
```

<template>

HTML in template is implementation detail of component

Shadow DOM

Not visible from "outside" DOM (light DOM).

CSS for the element

```
:host {
    border: 1px solid grey;
}
#segment {
    background-color: grey;
    display: block;
}
```

This CSS is not visible from outside. (Shadow DOM)

Style elements from outside

```
#my-timeline {
    border: 1px solid blue;
}
```

How do I style the segment?

Shadow DOM is in general not stylable from outside. There are some non-standardized ways to "cross" into shadow DOM:

```
#my-timeline {
    border: 1px solid blue;
}
#my-timeline::shadow #segment {
    background-color: red;
}
```

Better: redesign element

```
<style>
#my-timeline {
    border: 1px solid blue;
}
#my-timeline-segment {
    background-color: red;
}
</style>

<my-timeline>
    <my-timeline-segment>
</my-timeline>
```

Attributes

```
<my-timeline duration="10">
    <my-timeline-segment start="2" end="3">
</my-timeline>
```

Attributes in JS

```
<polymer-element name="my-timeline" attributes="duration">
  <script>
    Polymer('my-timeline', {
        duration: 5,
        durationChange: function() {
            // duration has changed
        }
        domReady: function() {
            // upgraded HTML elements are ready
        }
    });
  </script>
</polymer-element>
})
```

Backend as HTML elements

```
<my-backend-segment duration="{{duration}}">
<my-timeline duration="{{duration}}">
```

`my-backend-segment` encapsulates the backend API.

Polymer does the binding between the attributes of the elements: Setting duration in `<my-backend-segment>` triggers the JS function `durationChange()` in `<my-timeline>`.

Doesn't this make sense?

Starting points

- http://webcomponents.org/
- http://www.polymer-project.org/
- http://mozbrick.github.io/
- http://googlewebcomponents.github.io/
- http://component.kitchen/

Thx.