Web Components for non-idealists

What are web components?

Basically: widgets

Called "components"

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

Independent, importable

```html
<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

```html
<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

A future where you pick your HTML components like a backend library (eg. Serializer, Guzzle, ...) and do a `bower install` and lets you use this HTML component **has** to excite you.
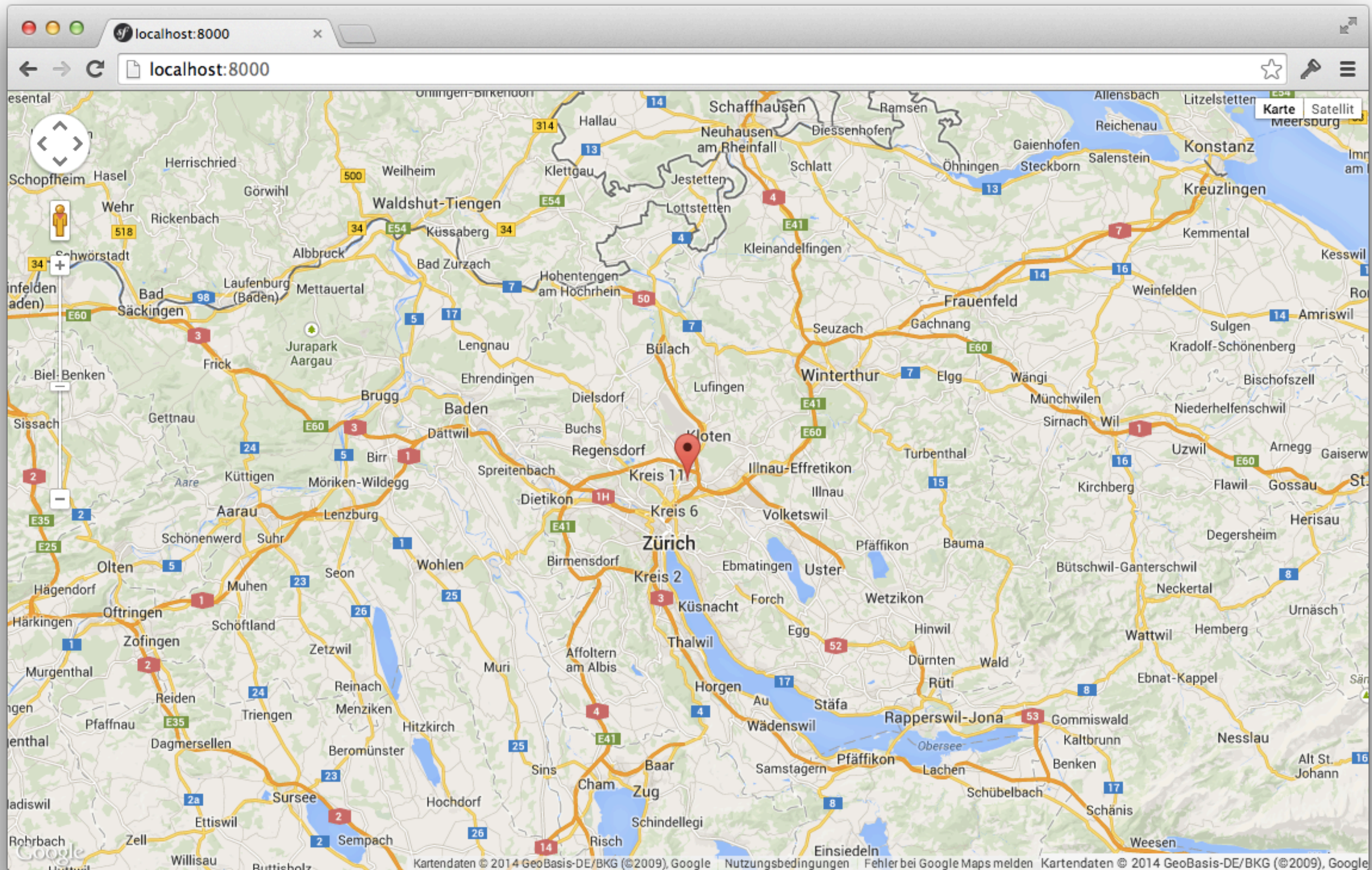
Yes, because

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 :)

As an example: Google maps

```html
<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>
```

As an example: Voice recognision

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


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

Browser support August 2014

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

jQuery for web components

Polyfill for web components, which allow web components even for Internet Explorer 9!

https://github.com/Polymer/platform

Libraries

- **Polymer from Google**
- **Mozilla Bricks**
- **X-Tags**

Polymer

- Based on platform.js like Mozilla Bricks.
- On top of that: Binding
- On top of that: Some sugar

LET'S START

Setup

```
bower init

bower install --save Polymer/polymer

mkdir my-timeline
```

INDEX.HTML

```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

```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>
```

Templates (or shadow DOM)

The HTML in the template is not visible from outside. Also, the id does not collide with outside id.

CSS for the element

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

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

Style elements from outside

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

But... 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:

```css
#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 in light HTML

```
<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>
})
```