

# Custom HTML-Elemente Polymer & AngularJS



# Zur Person

**Norbert Schmidt**, Web Developer

**Mayflower GmbH**, Würzburg und München



# Agenda

- Web-Components als neuer Web-Standard
- Web Components mit Polymer
- Custom HTML-Elemente mit AngularJS
- Polymer und AngularJS - passt das zusammen?

# Web Components

- Erster Working Draft des W3C: Mai 2012
- Sammlung von Techniken und Standards
- Templates
- Shadow DOM
- Custom Elements

# Web Components

## Templates

- Neuer Web-Standard
- Clientseitiges Templating, nativ in HTML
- Browser: Chrome, Opera, Firefox, Safari 8

# Web Components

## Beispiel für Templates

```
<body>
  <!-- Declaring the template -->
  <template id="logo">
    <style>
      div {...}
    </style>
    <div><img src="" alt=""/></div>
  </template>
  <script>
    var template = document.getElementById('logo');
    // Set image attributes at runtime
    template.content.querySelector('img').src = 'path-to-image';
    template.content.querySelector('img').alt = '';
    // Clone template's content, deep copy
    var templateClone = document.importNode(template.content, true);
    // "Activating" the template
    document.body.appendChild(templateClone);
  </script>
</body>
```

# Web Components

## Shadow DOM

- Die “dunkle Seite” des DOM
- Kapselung von DOM-Nodes
- JavaScript API f. Zugriff auf den Shadow DOM
- Browser: Chrome, Opera, Firefox

# Web Components

## Beispiel für Shadow DOM

```
<body>
  <template id="logo">...</template>
  <!-- Declaring the shadow host -->
  <div id="imageTag"></div>

  <script>
    // Shadow root definition
    var shadow = document.querySelector('#imageTag').createShadowRoot();
    var template = document.querySelector('#logo');

    // Set image attributes at runtime
    template.content.querySelector('img').src = '';
    template.content.querySelector('img').alt = '';

    // Clone template's content, deep copy
    var templateClone = document.importNode(template.content, true);

    // Appending the template to the shadow host
    shadow.appendChild(templateClone);
  </script>
</body>
```



# Web Components

## Custom Elements

- Dem Browser neue Tricks beibringen
- Neues HTML-Element mit **document.registerElement()**
- Elemente erweitern: `<input type="text" is="secure-input">`
- Browser: Chrome, Opera, Firefox

# **Web Components**

Elemente erweitern + erstellen

Beispiele...

# Polymer

- Bibliothek für Web Components
- Google-Projekt, Mitte 2013
- Vergleichbar mit X-Tags, Bricks von Mozilla
- Status: Developer-Preview, v.0.4.2
- Alle modernen Browser, inkl. IE 10

# Polymer

## Deklaration eines Elements

```
<polymer-element name="tag-name" constructor="TagName">
  <template>
    <!-- shadow DOM -->
  </template>
  <script>
    Polymer({
      // Properties and methods
    });
  </script>
</polymer-element>
```

# Polymer

## Data-Binding

- Two-Way Data-Binding
- Erweiterung der DOM-API
- Ein Model in Polymer ist das Element selbst

# **Polymer**

## **Beispiel für Data-Binding**

Beispiel...

# Custom-Elemente mit Angular

- AngularJS ist ein Application-Framework
- Dependency Injection, Module und MVC
- Two-Way Data-Binding
- Templating direkt in HTML
- Erweiterung von HTML mit dem Konzept der Direktiven

# Direktiven

- Marker im DOM
- Behaviour und Daten mit DOM-Nodes verknüpfen
- Verwendung als HTML-Attribut oder als
- CSS-Klasse oder als
- HTML-Kommentar oder als
- HTML-Tag



# Built-In Direktiven

```
<!-- ng-model: data-binding -->  
<input ng-model="user.name" type="text" placeholder="Your name...">  
  
<!-- ng-controller: attach controller function to view -->  
<div ng-controller="ApplicationCtrl" id="header">...</div>  
  
<!-- ng-include: fetch and including an HTML fragment -->  
<div ng-include="'view/header.tpl.html'"></div>
```

# Custom-Direktiven

## Definiton Object

```
.directive('myDirective', function (myService) {  
  // Returning the directive's "Definition Object" as a public API  
  return {  
    // execution order; multiple directives for one element  
    priority: 0,  
    // tied to 'priority'; can stop directive's execution  
    terminal: false,  
    // inline template, string or function  
    template: '<div><h3>{{title}}</h3></div>',  
    // include template file, string or function  
    templateUrl: 'myDirective.html',  
    // if true, defining element will be replaced  
    replace: false,  
    // perform tasks before directive is linked to a scope  
    compile: function (element, attributes, transclude) {},  
    // manipulate the DOM; main place of the directive's logic  
    link: function ($scope, $element, $attrs) {},  
    // inherit parent scope (false, true); isolated scope: {}  
    scope: false,  
    // binds the directive to a controller function; string or function  
    controller: function ($scope, $element, $attrs) {},  
    // tied to 'controller'; makes an other directives controller available  
    require: 'myOtherDirective',  
    // applies the parent scope to a part of the directive's template  
    transclude: true  
  };  
});
```

# Custom-Direktiven

## Ein Beispiel

Sample App: The Shoutbox

# **Polymer + Direktiven**

**Passt das zusammen?**

**Dankeschön!**  
**Fragen?**

# Kontakt



**Mail:** [norbert.schmidt@mayflower.de](mailto:norbert.schmidt@mayflower.de)

**GitHub:** <https://github.com/nosch>

**Twitter:** @noschm