



# WebComponents

Bausteine des modernen Web

# Thomas Kruse

- Entwickler, Trainer, Berater
  - [www.trion.de](http://www.trion.de)
  - @everflux
- Java User Group Münster
- Frontend Freunde Münster



# Karsten Sitterberg

- Entwickler, Trainer, Berater
  - [sitterberg.com](http://sitterberg.com)
  - @kakulty
- Java User Group Münster
- Frontend Freunde Münster



# Revolution of the web

- HTML (+ Forms)
- Applets / Flash
- Ajax
- HTML5
- Reichhaltige Frontend-Anwendungen

# Komponenten

Perfekt für UI

The screenshot displays the Postbank online banking interface. At the top, the Postbank logo and slogan "Eine Bank fürs Leben." are on the left. A yellow banner contains the text "DEMOKONTO". On the right, the user's name "PETRA PFIFFIG" is shown with a plus icon, followed by account details: "Postbank Giro plus: 9999999999, BLZ: 20010020" and "Sitzungsende in 08:00". A "Banking beenden" button is also present.

Below the banner is a navigation bar with tabs: "Kontenübersicht", "Umsätze", "Auftragslisten" (highlighted), "Service", "Nachrichtenbox" (with a notification badge "1"), and "Einstellungen".

The main content area shows the heading "Auftragslisten für Konto: Petra Pffiffig, Konto: 9999999999, BLZ: 20010020". Below this are four tabs: "Überweisungen" (highlighted), "Auslandsüberweisungen", "Daueraufträge", and "Sparaufträge".

Under the "Überweisungen" tab, there is a "Neue Überweisung" button. Below it is a table header with columns: "Buchungstermin", "Empfänger", "Betrag €", "Herkunft", "Status", and a search field with an "X" icon.

The table content shows a message: "Es liegen für das Girokonto 9999999999 keine eingestellten Überweisungen vor."

# Komponenten

Perfekt für UI

The screenshot shows the Postbank online banking interface. The top navigation bar is yellow and contains the Postbank logo, a 'DEMOKONTO' badge, the user name 'PETRA PFIFFIG', account details, and a 'Banking beenden' button. The main navigation bar is light blue and contains links for 'Kontenübersicht', 'Umsätze', 'Auftragslisten' (highlighted), 'Service', 'Nachrichtenbox', and 'Einstellungen'. Below the navigation, there is a message: 'Auftragslisten für Konto: Petra Pfiffig, Konto: 9999999999, BLZ: 20010020'. A secondary navigation bar contains buttons for 'Überweisungen', 'Auslandsüberweisungen', 'Daueraufträge', and 'Sparaufträge'. The main content area features a 'Neue Überweisung' button and a table with columns: 'Buchungstermin', 'Empfänger', 'Betrag €', 'Herkunft', and 'Status'. The table is currently empty, displaying the message: 'Es liegen für das Girokonto 9999999999 keine eingestellten Überweisungen vor.'

The screenshot shows the Postbank online banking interface, similar to the first one but with the 'Daueraufträge' (Direct Debits) section active. The top navigation bar is yellow and contains the Postbank logo, a 'DEMOKONTO' badge, the user name 'PETRA PFIFFIG', account details, and a 'Banking beenden' button. The main navigation bar is light blue and contains links for 'Kontenübersicht', 'Umsätze', 'Auftragslisten', 'Service', 'Nachrichtenbox', and 'Einstellungen'. Below the navigation, there is a message: 'Auftragslisten für Konto: Petra Pfiffig, Konto: 9999999999, BLZ: 20010020'. A secondary navigation bar contains buttons for 'Überweisungen', 'Auslandsüberweisungen', 'Daueraufträge' (highlighted), and 'Sparaufträge'. The main content area features a 'Neuer Dauerauftrag' button and a table with columns: 'Empfänger', 'Betrag €', 'Verwendungszweck', 'Herkunft', and 'Nächste Abbuchung'. The table is currently empty, displaying the message: 'Es liegen für das Girokonto 9999999999 keine Daueraufträge vor.'

# Komponenten

Perfekt für UI

The screenshot displays the Postbank online banking interface. At the top, the Postbank logo and tagline 'Eine Bank fürs Leben.' are on the left. A yellow banner contains the user's name 'PETRA PFIFFIG' with a plus icon, the account details 'Postbank Giro plus: 9999999999, BLZ: 20010020', the session end time 'Sitzungsende in 08:00', and a 'Banking beenden' button. A 'DEMOKONTO' badge is also present. Below the banner is a navigation bar with buttons for 'Kontenübersicht', 'Umsätze', 'Auftragslisten' (highlighted), 'Service', 'Nachrichtenbox' (with a '1' notification), and 'Einstellungen'. The main content area shows 'Auftragslisten für Konto: Petra Pfiffig, Konto: 9999999999, BLZ: 20010020'. A row of buttons includes 'Überweisungen' (highlighted), 'Auslandsüberweisungen', 'Daueraufträge', and 'Sparaufträge'. Below this is a 'Neue Überweisung' button. At the bottom, a table header for transactions is shown with columns: 'Buchungstermin', 'Empfänger', 'Betrag €', 'Herkunft', 'Status', and a search icon. A message below the table states: 'Es liegen für das Girokonto 9999999999 keine eingestellten Überweisungen vor.'

**Postbank**  
Eine Bank fürs Leben.

**DEMOKONTO**

PETRA PFIFFIG +  
Postbank Giro plus: 9999999999, BLZ: 20010020  
Sitzungsende in 08:00

Banking beenden

Kontenübersicht | Umsätze | **Auftragslisten** | Service | Nachrichtenbox <sup>1</sup> | Einstellungen

Auftragslisten für Konto: **Petra Pfiffig, Konto: 9999999999, BLZ: 20010020**

Überweisungen | Auslandsüberweisungen | Daueraufträge | Sparaufträge

**Neue Überweisung**

Buchungstermin	Empfänger	Betrag €	Herkunft	Status	
----------------	-----------	----------	----------	--------	--

Es liegen für das Girokonto 9999999999 keine eingestellten Überweisungen vor.

# Komponenten

The screenshot shows a Postbank online banking interface. The top header includes the Postbank logo and the text 'Eine Bank fürs Leben.' and 'DEMOKONTO'. The user's name 'PETRA PFIFFIG' and account details 'Postbank Giro plus: 9999999999, BLZ: 20010020' are displayed, along with a 'Banking beenden' button. A navigation bar contains 'Kontenübersicht', 'Umsätze', 'Auftragslisten', 'Service', 'Nachrichtenbox', and 'Einstellungen'. A message box indicates 'Auftragslisten für Konto: Petra Pfiffig, Konto: 9999999999, BLZ: 20010020'. Below this, there are tabs for 'Überweisungen', 'Auslandsüberweisungen', 'Daueraufträge', and 'Sparaufträge'. The 'Auslandsüberweisungen' section is active, showing a 'Neue Auslandsüberweisung' button and a table of transactions.

Buchungstermin	Empfänger	Betrag	Herkunft	Status	
04.07.2016	Peter Pfiffig	256,12 EUR	Unbekannt	legt vor	
31.03.2016	Susi Sorglos	1.500,00 EUR	Unbekannt	legt vor	🗑️
26.03.2016	Maria Pfahl	256,12 EUR	Unbekannt	legt vor	🗑️



# Herausforderungen im Browser

- Serverseitig: Viele Komponenten Frameworks (Wicket, JSF, ...)
- Im Browser existiert kein natives Komponenten/Templating System
  - Import von JavaScript: `<script>`
  - Import von CSS: `<style>`

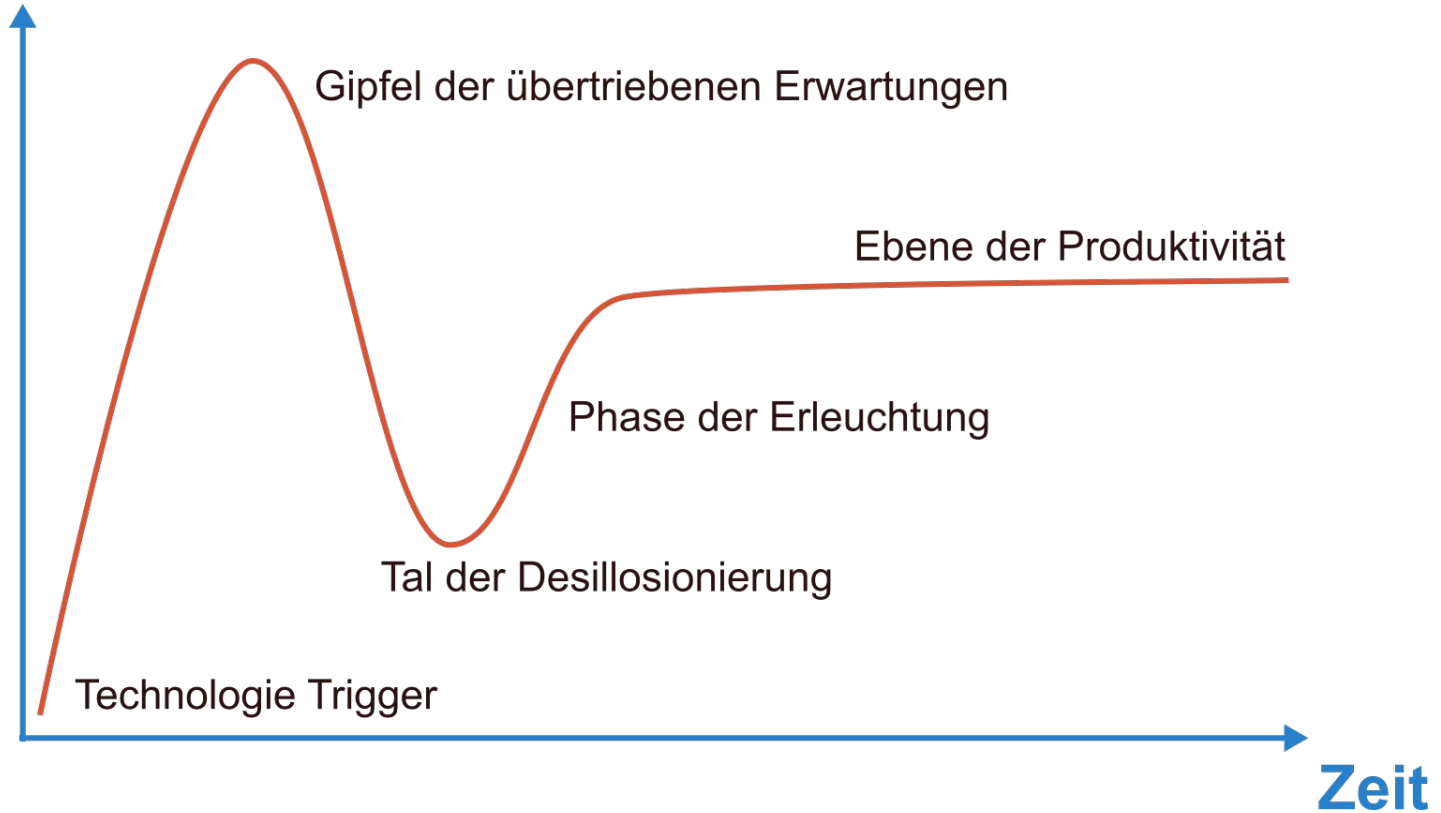
# Demo

Probleme mit gemischten Komponenten

# WebComponents

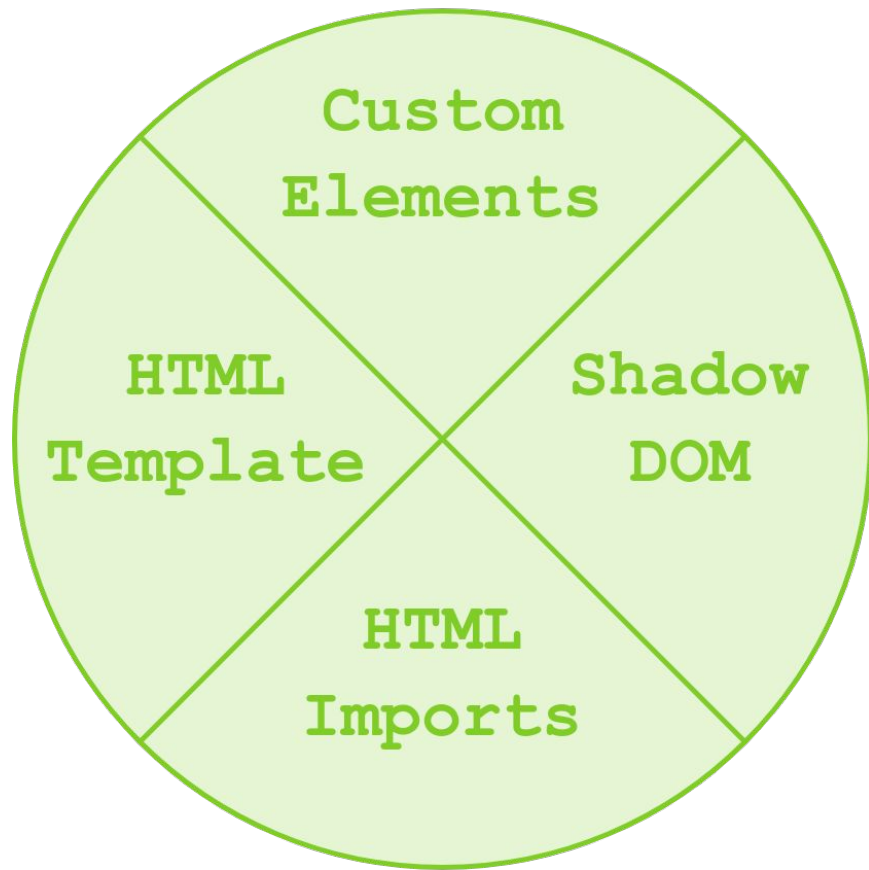
Hilfe oder Hype?

# Aufmerksamkeit

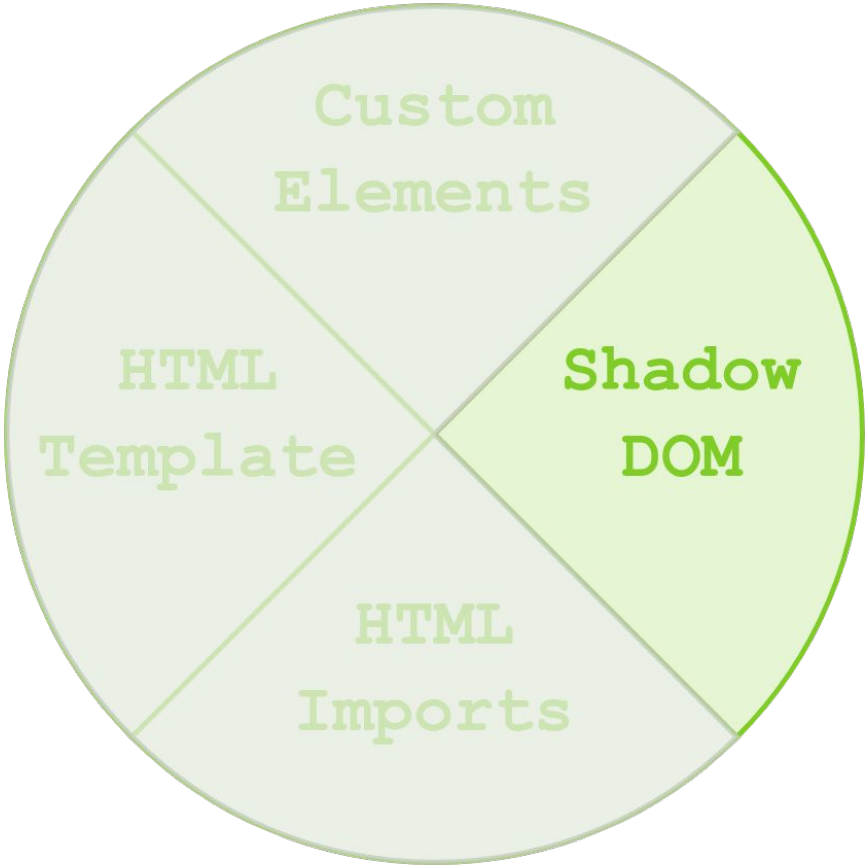


Gartner Hype-Kurve

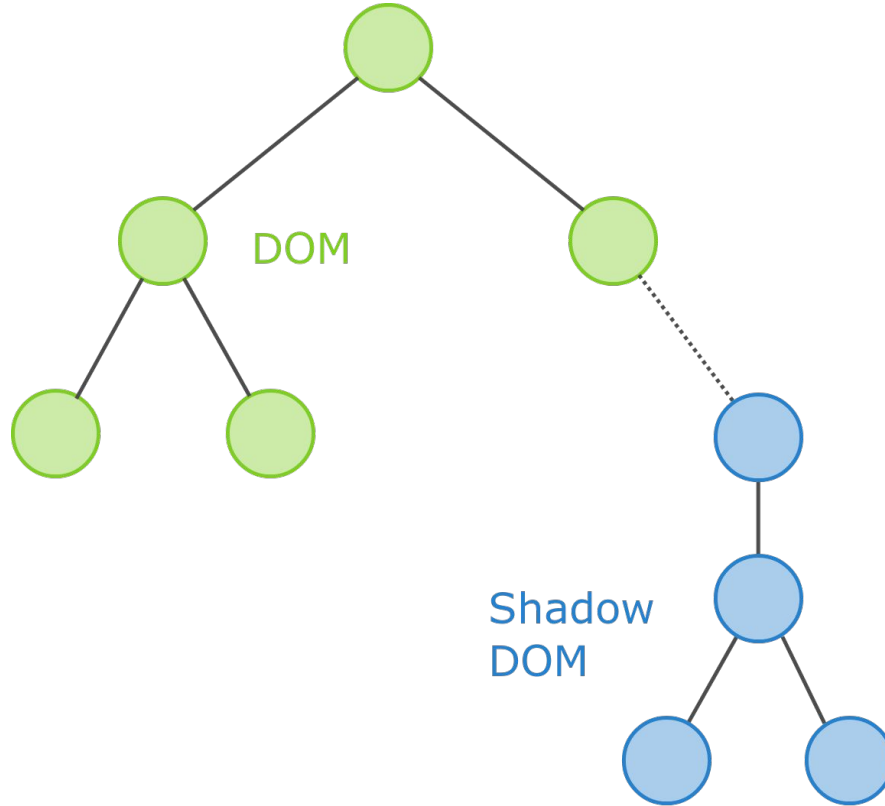
# Web Components



# Web Components

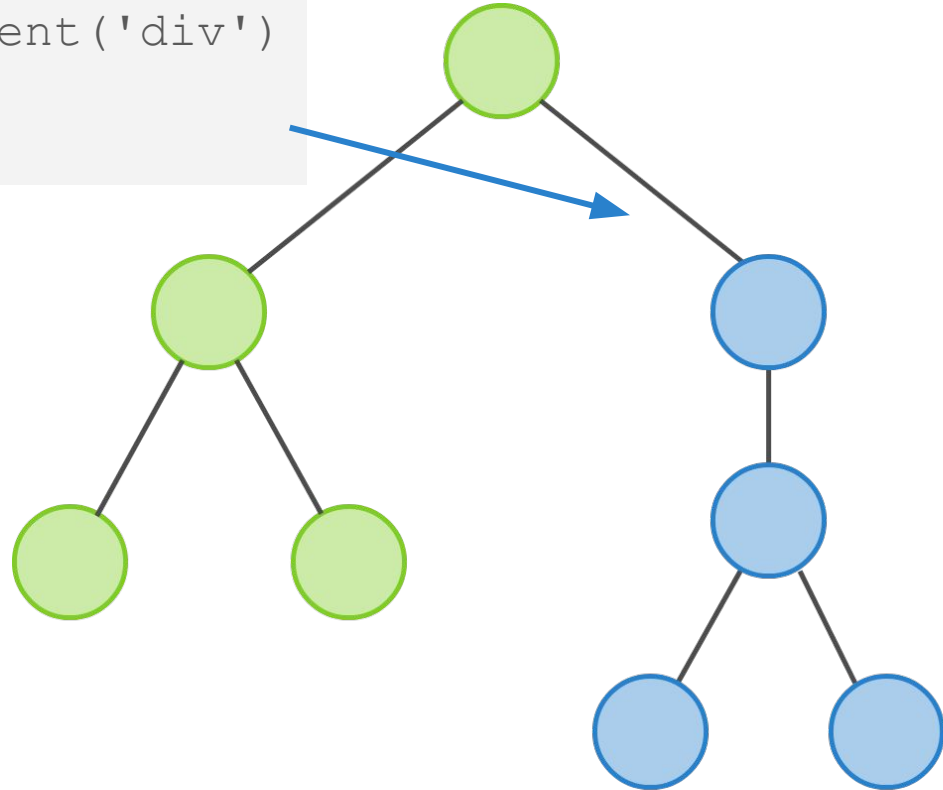


# Shadow DOM



# Herkömmlich

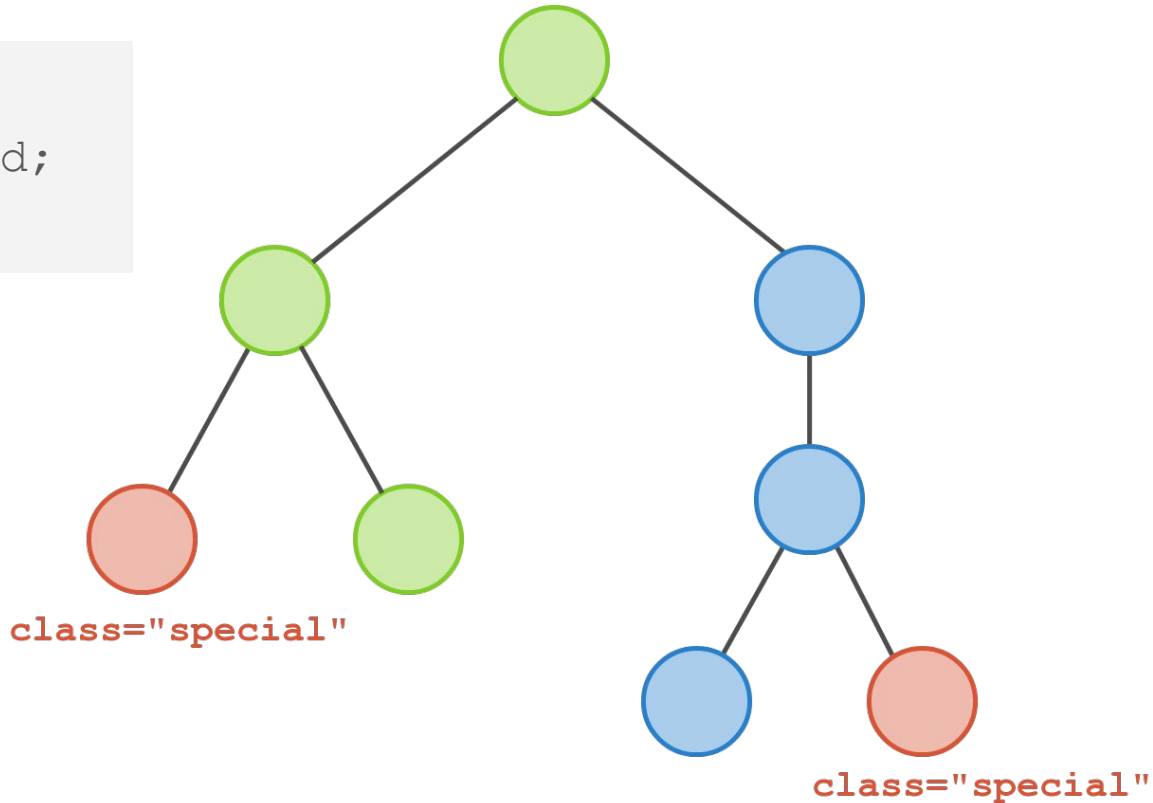
```
const el = document.createElement('div')  
document.body.appendChild(el)
```



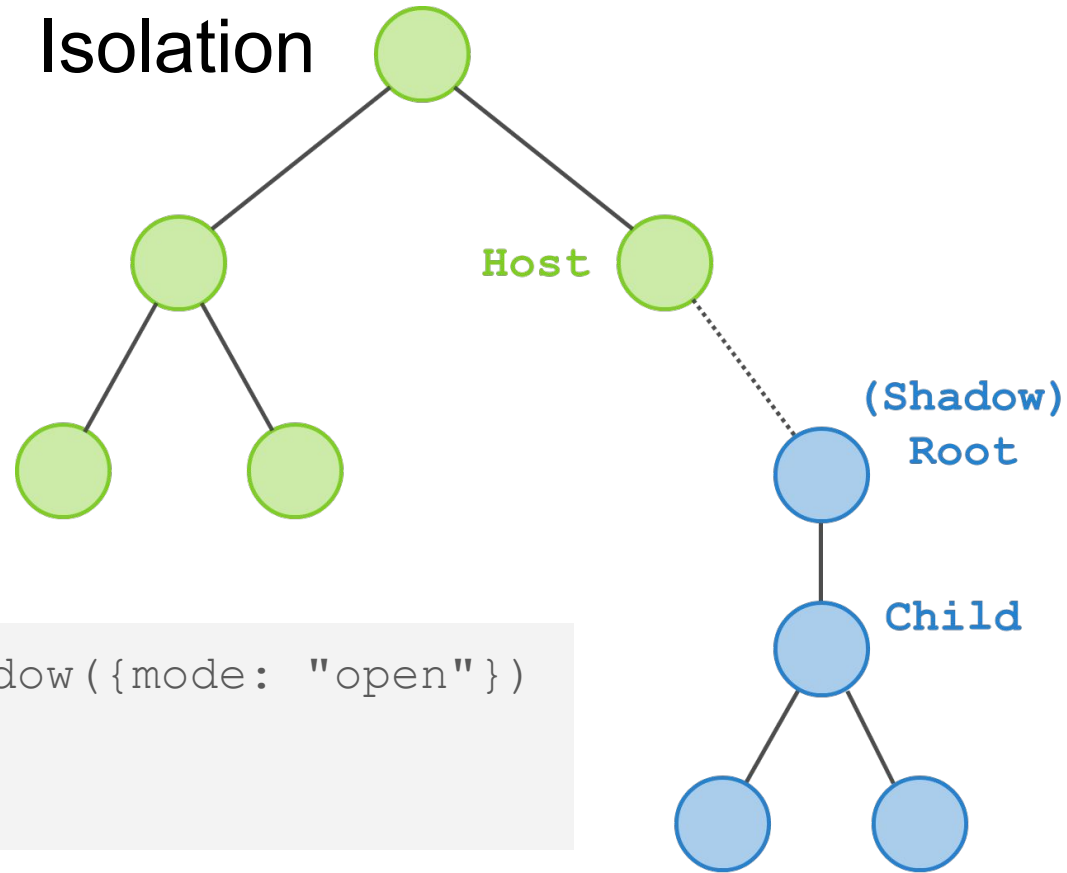


# Problem: Überschreiben der Styles

```
.special {  
  background-color: red;  
}
```



# Lösung: Shadow-DOM Isolation

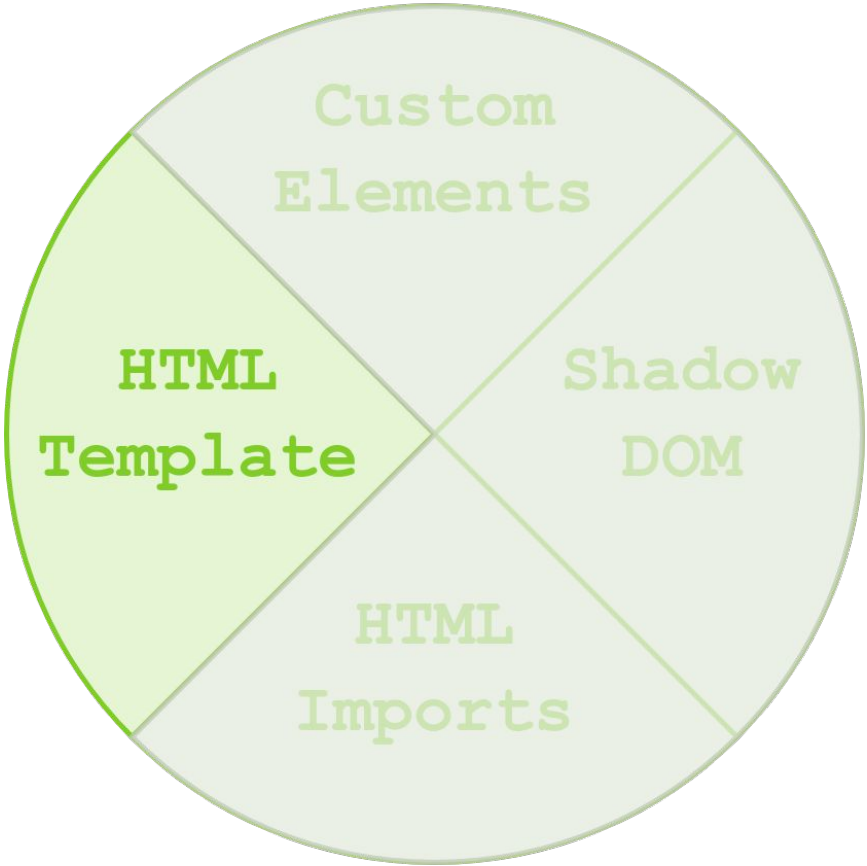


```
const root = host.attachShadow({mode: "open"})  
root.appendChild(child)
```

# Demo

Shadow DOM

# Web Components

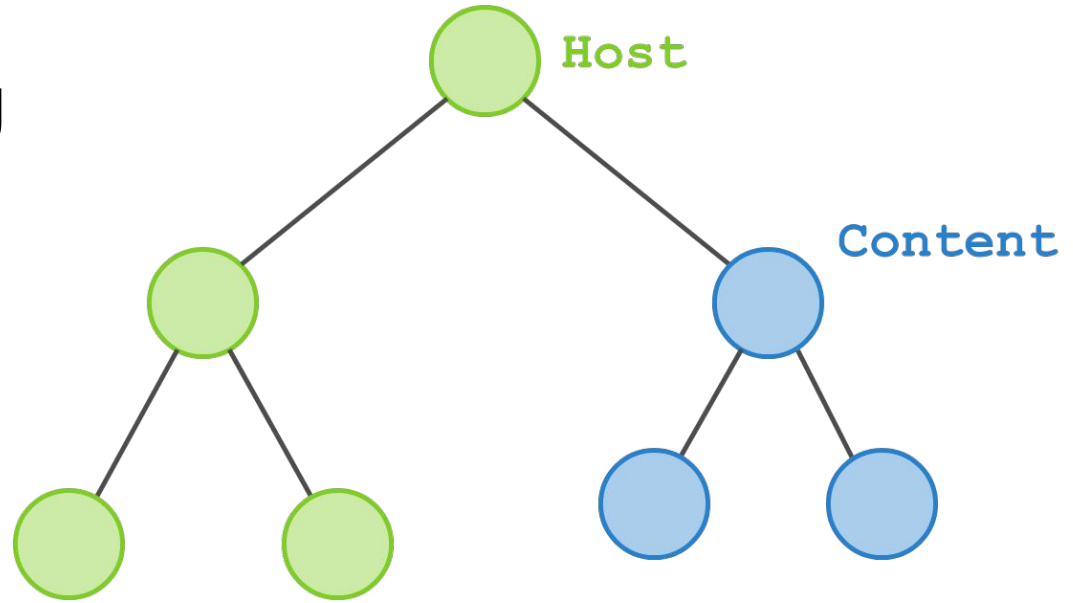


# Template

- Kleine, wiederverwendbare HTML-Schnipsel
- Werden beim initialen Laden **nicht** gerendert
- Müssen durch ein Script in den DOM eingefügt werden
- Beispiel eines Template:

```
<template id="demo">
  <div>
    Ich bin ein Template-Div
  </div>
</template>
```

# Template: Einbindung



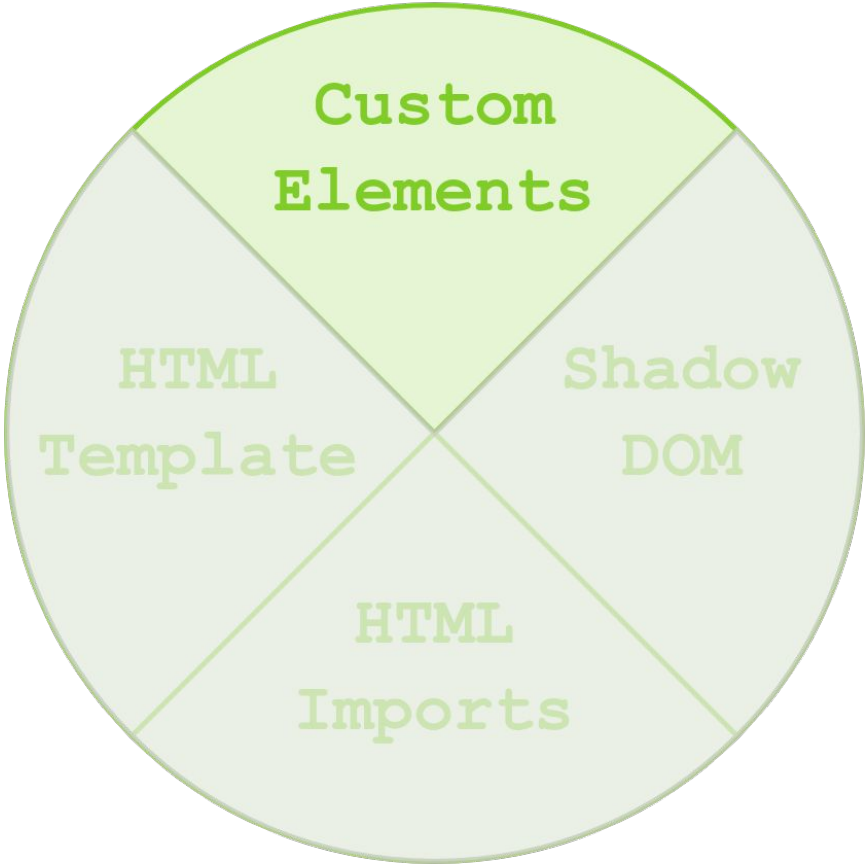
```
const tmplContent = document.getElementById('demo').content;  
const contentNode = document.importNode(tmplContent, true);  
host.appendChild(contentNode);
```

↑  
**Deep-Copy**

# Demo

Template

# Web Components

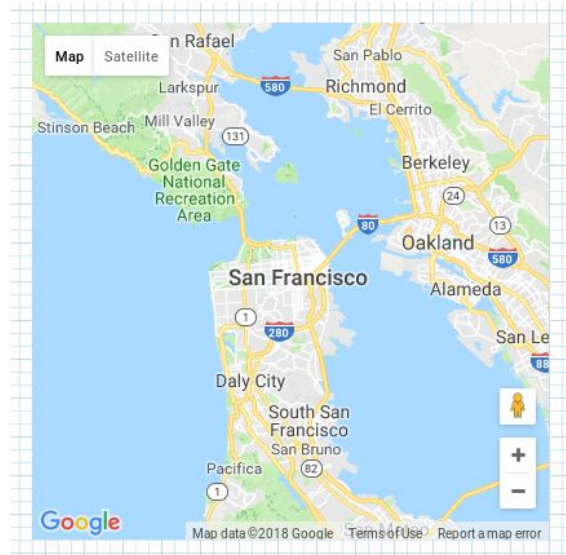




# Custom Elements

- Bieten die Möglichkeit, eigene HTML-Elemente zu definieren
- Oder bestehende Elemente zu erweitern
- Erben alle von `HTMLElement`

```
<body>  
  <google-map></google-map>  
  ...  
</body>
```



# Custom Element

```
class MyComponent extends HTMLElement {  
  constructor() {  
    super();  
    const content = document.createElement('section');  
    content.innerHTML = `    this.appendChild(content);  
  }  
}  
  
window.customElements.define('my-component', MyComponent);
```

# Custom Element zur Laufzeit

```
<my-component>  
  <section>  
    <div>Ich bin eine Komponente</div>  
  </section>  
</my-component>
```

# Demo

Basic Custom Element

# Lifecycle

- Custom Elements durchlaufen verschiedene Phasen
  - Lebenszyklus
- Erzeugung: `constructor()`
- In den DOM eingefügt: `connectedCallback()`
- In anderes Dokument eingehängt: `adoptedCallback()`
- Update der Attribute von außen: `attributeChangedCallback()`
- Aus dem DOM entfernt: `disconnectedCallback()`

# API

- Durch Attribute können Werte in die Komponente gereicht werden
- Attribute müssen deklariert werden
  - `observedAttributes()`

```
static get observedAttributes() {  
  return ['bgcolor', 'textcolor'];  
}
```

- Es können die Attribute `bgcolor` und `textcolor` ausgelesen werden

# API

- Um auf Änderungen der Attribute zu reagieren
  - `attributeChangedCallback()`
- Callback besitzt Parameter
  - **name**: Name des Attributs, das geändert wurde
  - **oldValue**: Alter Wert des Attributs
  - **newValue**: Neuer Wert des Attributs

```
attributeChangedCallback(name, oldValue, newValue) {  
  console.log('attribute changed');  
  updateElement();  
}
```

# API - Custom Events

- Durch Custom Events kann die Komponente mit der “Außenwelt” kommunizieren

```
const smileEvent = new CustomEvent('smile', {  
  bubbles: true,  
  detail: { text: 'Hurray!' }  
});  
  
this.dispatchEvent(smileEvent);
```



# API - Custom Events

- Durch Custom Events kann die Komponente mit der “Außenwelt” kommunizieren

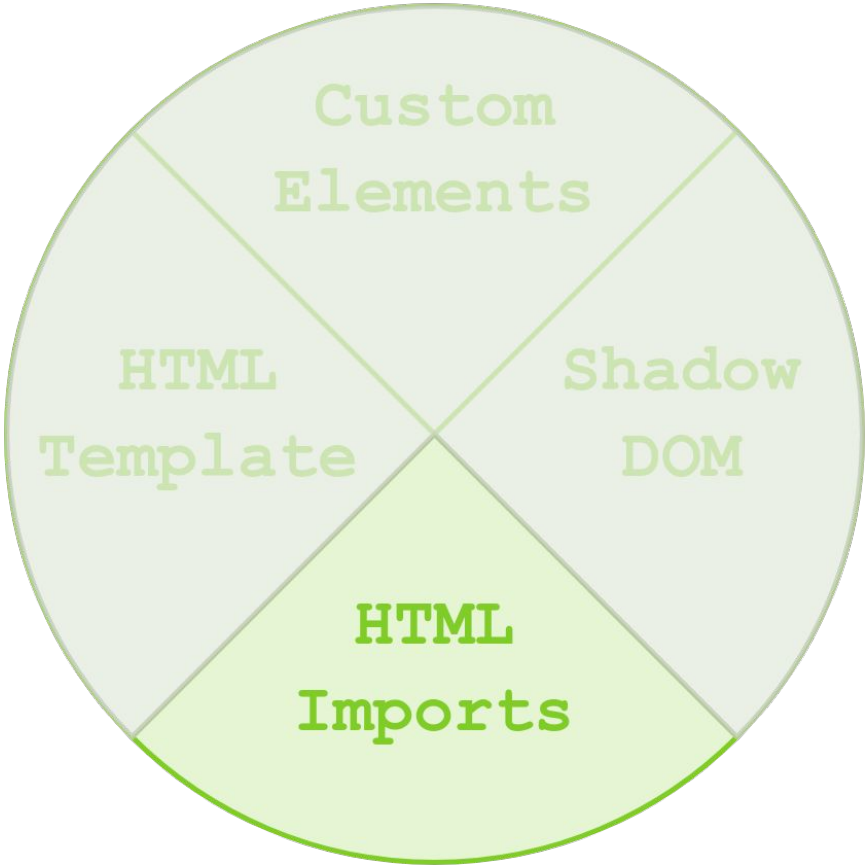
```
element.addEventListener (  
  'smile',  
  e => {  
    console.log(e.detail.text)  
  },  
  false  
);
```

**Listen to Event in  
Bubble-Phase**

# Demo

Custom Element

# Web Components



# HTML Imports

- HTML-Schnipsel in eigenen HTML-Dokumenten verwalten
- Jedes dieser Dokumente separat importierbar

```
<link rel="import" href="../my-template/my-template.html">
```

# Demo

Import

# Web Components v1

- Web Components offiziell released
- Keine schwerwiegenden API-Brüche
- Standardisierung über alle Browser
- HTML **Imports** deprecated!
  - Stattdessen: ES-Module

# Achtung

- HTML Imports sind im Prinzip deprecated
- Werden von Firefox, Safari und Microsoft nicht umgesetzt
  - Stattdessen soll auf ECMAScript-Module gesetzt werden
- Alternativ sind hierfür Polyfills möglich
  - `webcomponent.js`

## Browser support



CHROME



OPERA



SAFARI



FIREFOX



EDGE

### HTML TEMPLATES



STABLE



STABLE



STABLE



STABLE



STABLE

### CUSTOM ELEMENTS



STABLE



STABLE



STABLE



POLYFILL



DEVELOPING



POLYFILL



CONSIDERING

### SHADOW DOM



STABLE



STABLE



STABLE



POLYFILL



DEVELOPING



POLYFILL



CONSIDERING

### ES MODULES



STABLE



STABLE



STABLE



STABLE



STABLE

### HTML IMPORTS



STABLE



STABLE



POLYFILL



POLYFILL



POLYFILL



# Demo

Visual Designer

<http://polymer-designer.appspot.com/>

# Statische Verwendung

- Nicht immer müssen Komponenten dynamisch geladen werden
- Beispiel: Server Side Anwendungen (Rails, Spring MVC, JSF)
  - Feste Einbindung der Web Components ins gerenderte HTML
  - Etwa für bessere UX + Integrationszwecke

# Verwendung bei Anwendungs-Build

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <script src="bower_components/webcomponentsjs/webcomponents.js"></script>  
  </head>  
  <body fullbleed layout vertical>Hello World</body>  
</html>
```

# Erweiterungen

```
<head>
  <script src=„bower_components/webcomponentsjs/webcomponents.js“></script>
  <link rel="import" href="bower_components/core-header-panel/core-header-panel.html">
  <link rel="import" href="bower_components/core-toolbar/core-toolbar.html">
</head>
<body fullbleed layout vertical>
  <core-header-panel flex>
    <core-toolbar>Title</core-toolbar>
  </core-header-panel>
</body>
```

# Fazit

- Web Components eignen sich als Building-Blocks
  - Wiederverwendbar in unterschiedlichen Kontexten
  - Elemente bieten API, Verwendung mit Werkzeugen möglich
- Komponentenbibliotheken
  - <https://www.webcomponents.org/>
  - <https://www.primefaces.org/primeui/>
  - <https://www.telerik.com/kendo-ui>
- Toolkits / Framework-Support
  - Polymer (Google), X-Tag (Mozilla)
  - Angular Elements, Vue-custom-components, React
  - Vorteile vs. Vanilla WebComponent: Know-How, Tooling (Doc, Static Analyzer, ...), Libraries

# Danke.

Fragen?

@kakulty

[sitterberg.com](https://sitterberg.com)

@everflux

[trion.de](https://trion.de)