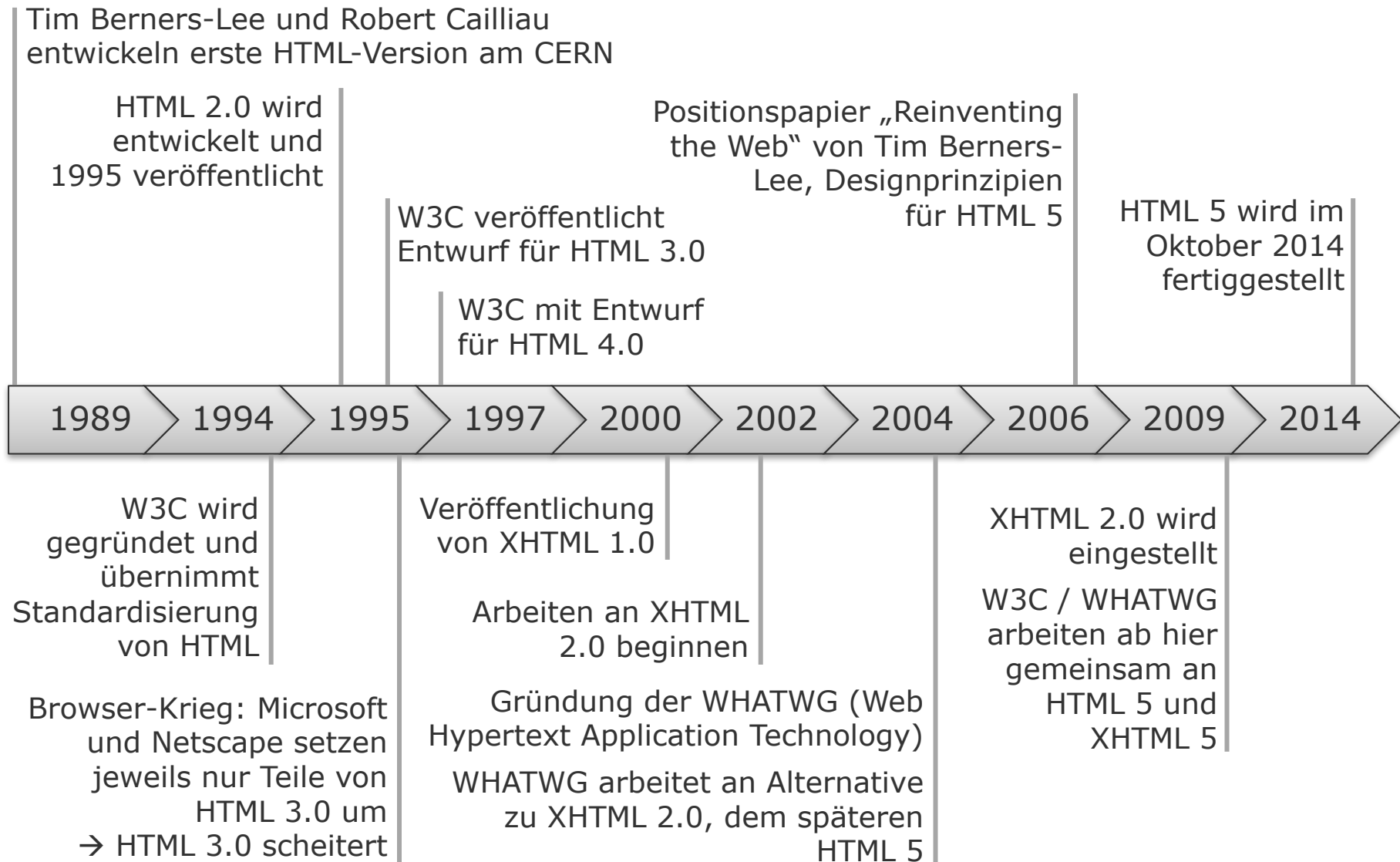




openHPI
HTML 5

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Kurze Geschichte von HTML: von HTML 1.0 über XHTML zu HTML 5



HTML 5 – Ziele und Paradigmen

Nachdem HTML-Arbeitsgruppe des W3C zerbrach, konzentrierte sich die WHATWG auf die Entwicklung eines praxisorientierten Standards

- Neuer Standard sollte gemeinsam von Nutzern, Autoren, Designern, Browser-Herstellern, ... entwickelt werden
- **Entwicklungsparadigma:** „Nutzer vor Autoren vor Entwicklern vor Spezifikationsdesignern vor „reiner Lehre“
- HTML 5 solle
 - Stück für Stück entwickelt werden
 - abwärtskompatibel sein
 - einfach zu benutzen sein
 - noch stärker auf Trennung zwischen Struktur und Layout achten

HTML 5 – Was ist neu?

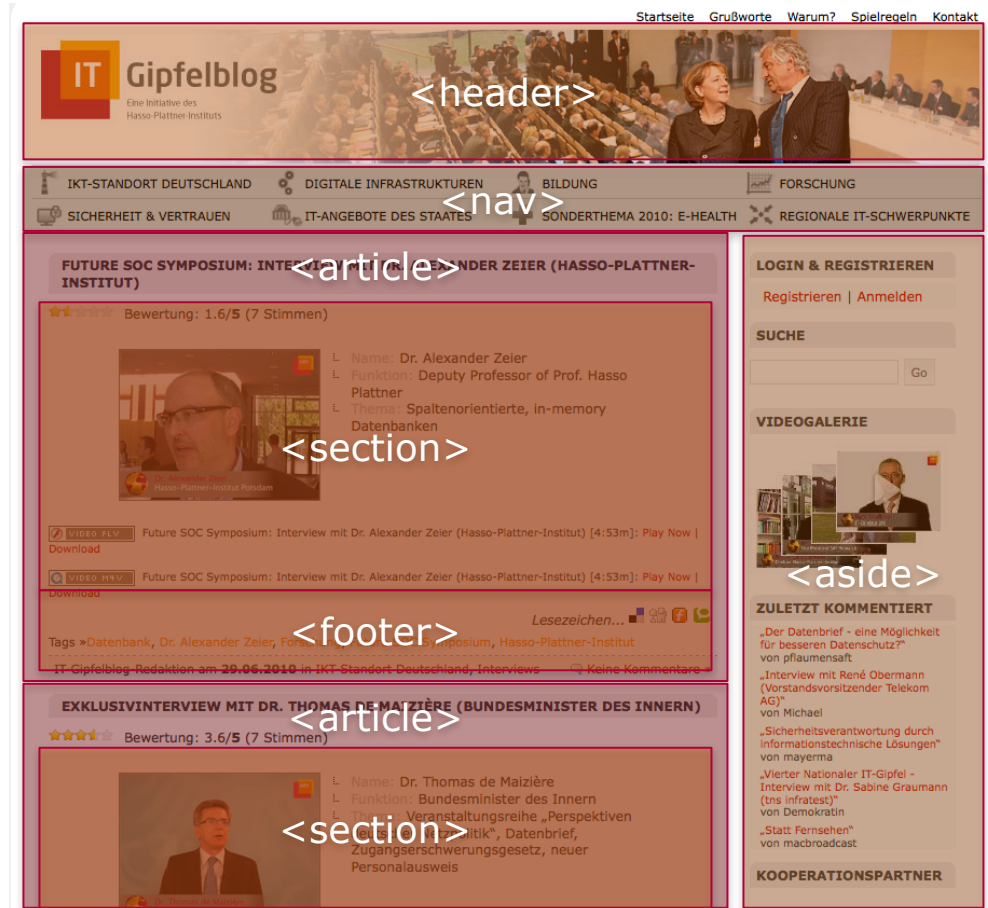
- HTML 5 basiert nicht länger auf SGML
 - Keine Document Type Definition (DTD)
 - Automatische Validierung (Syntaxprüfung) mit bestehenden Tools nicht mehr möglich, spezielle Werkzeuge werden benötigt
- HTML 5 führt Reihe „**sprechender**“ **Strukturelemente** ein, die eine bessere Strukturierung erlaubt
- HTML 5 bietet Reihe neuer Elemente für **Multimedia-Inhalte**
- Für **Formulare** gibt es viele neue Tags und Attribute
- **<canvas>**-Element ist Zeichenfläche für Animationen
- SVG (Scalable Vector Graphics) und MathML können direkt in HTML-Dokumenten verwendet werden
- Viele Elemente und Attribute für die Formatierung werden nicht mehr unterstützt

Semantische Elemente (1/2)

- Meistverwendetes strukturierendes Block-Element in HTML 4.0 ist `<div>`-Tag
- HTML 5 führt dafür eine Reihe „sprechender Strukturelemente“, sogenannte **semantische Elemente** (Elemente mit „Bedeutung“) ein, z.B.
 - `<section>`
 - `<article>`
 - `<header>`, `<footer>`
 - `<aside>`
 - `<nav>`
 - ...
- Maschinen (z.B. Suchmaschinen) können so leichter identifizieren, welche Teile des Dokuments wichtig sind oder welche für die Navigation verwendet werden

Beispiel

```
...  
<header>...</header>  
<nav>...</nav>  
<article id="1">  
  <header>...</header>  
  <section>...</section>  
  <footer>...</footer>  
</article>  
<article id="2">  
  ...  
</article>  
...  
<aside>...</aside>  
<footer>...</footer>  
...
```



Multimedia-Elemente (1/3)

- Audio und Video konnten bisher nur mittels Browser-Plugins in HTML-Dokumenten verwendet werden, z.B. Windows Media Player, Quicktime, VideoLAN client, Adobe Flash, usw.
- Neue Tags für **<audio>** und **<video>** erlauben es, Multimedia-Dateien direkt mit dem Browser abzuspielen
- Wie alle Elemente sind auch die Multimedia-Tags Teil des
→ Document Object Models (DOM)
 - Bedienelemente (Play, Pause, Lautstärke, ...) können mit CSS gestaltet werden



Chrome



Safari



Firefox

- Komplexere Multimedia-Steuerung (z.B. Sprungmarken, Playlisten, usw.) können mit JavaScript realisiert werden

Multimedia-Elemente (2/3)

- Betriebssysteme bzw. Browser müssen für Wiedergabe sogenannte **Codecs** (Verfahren für Kodierung/Dekodierung) unterstützen
- Vorgesehene Codecs für HTML 5 sind:
 - Audio: mp3, wma, AAC, Ogg Vorbis
 - Video: H.264, Ogg Theora, WebM
- Alternative Quellen (z.B. für unterschiedliche Codecs oder Bitraten) können mit **<source>**-Tag angegeben werden
- (Früher) **problematisch**: Standard legt nicht fest, welche Codecs unterstützt werden müssen, Browser-Hersteller handhaben das unterschiedlich, z.B. bei Video-Codecs:
 - Internet Explorer und Safari unterstützen nur H.264
 - Chrome, Firefox und Opera unterstützen alle Codecs
 - Früher mussten Inhaltenanbieter mehrere Codecs bereitstellen, heute reicht in der Regel H.264 aus

Multimedia-Elemente (3/3)

Beispiel: <audio>-Tag mit verschiedenen Quellen

```
<audio>
  <source src="/audio/track1.wma" type="audio/x-ms-wma">
  <source src="/audio/track1.mp3" type="audio/mpeg">
  <p>Wenn Sie diesen Text sehen, unterstützt Ihr Browser
    den HTML5 Audio-Tag nicht</p>
</audio>
```

- ❑ Der Browser spielt das erste <source>-Element ab, dessen Codec er versteht
- ❑ Unbekannte Tags ignoriert ein Browser, daher ist es wichtig, einen Hinweis für ältere Browser hinzuzufügen

Formulare (1/4)

- HTML 5-Formulare basieren auf Standard Web Forms 2.0
 - Abwärtskompatibel zu Web Forms 1.0 (gehört zu HTML 4.0)
- Gegenüber bisherigen HTML-Formularen gibt es einige Neuerungen, z.B.
 - Input-Elemente können auch außerhalb von Formularen stehen und diesen mit dem **form**-Attribut zugeordnet werden
 - `<input type="checkbox" form="registrierung"> AGB akzeptiert`
 - Neue Elemente: `<datalist>`, `<keygen>`, `<output>`
 - Neue Input-Typen, z.B. für Zahlen, Zeit und Datum, Farbwerte
 - Neue Attribute für Formularelemente, z.B.
 - für Eingabevalidierung: `min`, `max`, `required`, ...
 - mit Platzhalter
 - ...

Formulare (2/4)

Beispiele für neue Features

- Zahlenbereiche können mit Slider ausgewählt werden

```
<input type="range" min="0" max="10">
```



- Mit dem <datalist>-Element können Vorschläge (mit Auto-Vervollständigung) zu Texteingabefeldern ergänzt werden

```
<input type="text" list="browsers">
```

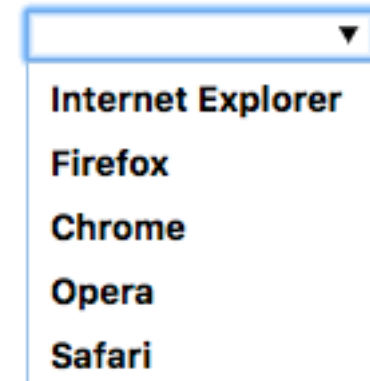
```
<datalist name="browsers">
```

```
  <option>Internet Explorer</option>
```

```
  <option>Firefox</option>
```

```
  ...
```

```
</datalist>
```



- Eingabefelder nur für numerische Werte (mit dem **value**-Attribut und Up-/Runter-Pfeilen)

```
<input type="number" value="100">
```



Vorbelegung von
Formularfeldern mit
dem **value**-Attribut

Formulare (3/4)

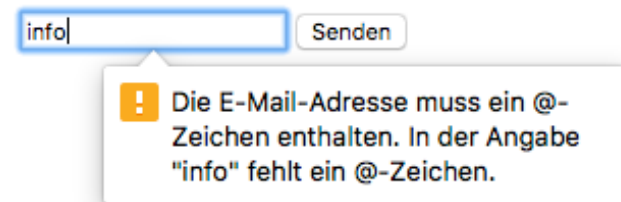
Beispiele für neue Features

- Eingabefelder mit Eingabevalidierung, z.B. für Email-Adressen

```
<input type="email">
```

```
<input type="url">
```

Bei fehlerhaften Eingaben wird Formular nicht gesendet



- Farbauswahl mit „Color Picker“
(nicht für jeden Browser)

```
<input type="color">
```



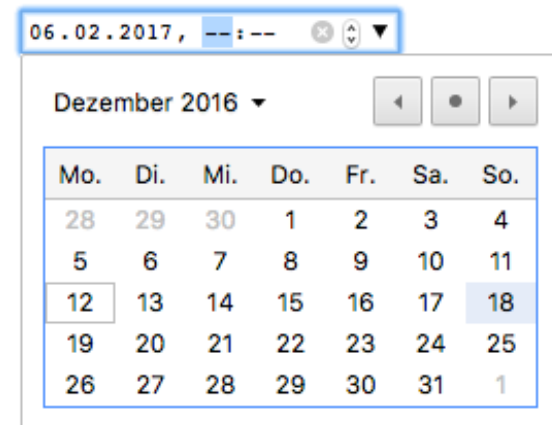
Formulare (4/4)

Beispiele für neue Features

■ Eingabefelder für Zeit und Datum, z.B.

`<input type="datetime-local">`

- analog: date, time, month, week
- Auswahl-Dialog für manche Browser



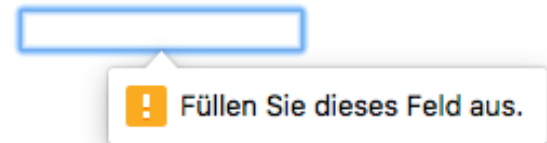
06.02.2017, --:--

Dezember 2016

Mo.	Di.	Mi.	Do.	Fr.	Sa.	So.
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

■ Attribute für **Eingaberestriktionen**

- max, min: höchste/niedrigste Werte
- required: Feld muss ausgefüllt werden
- pattern: regulärer Ausdruck für Eingabemuster



! Füllen Sie dieses Feld aus.

■ Weitere neue Attribute (Auswahl)

- placeholder: Hinweis auf erwartete Eingabe

`<input type="email" placeholder="Ihre E-Mail-Adresse">`

- multiple: Auswahl mehrerer Dateien für Formular-Upload



Ihre E-Mail-Adresse

Zeichenfläche – Canvas (1/2)

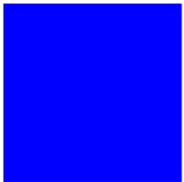
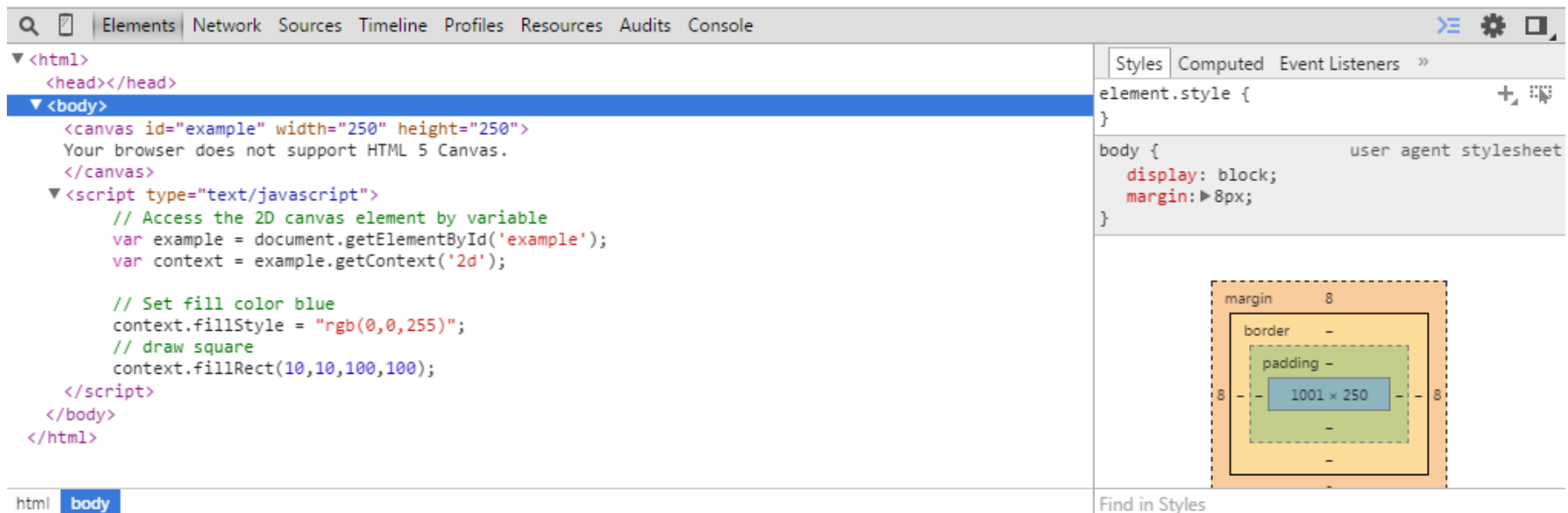
- **<canvas>-Tag** bietet Zeichenfläche für Bitmap-Grafiken
- Ursprünglich von Apple für WebKit-Browser entwickelt, mittlerweile in den WHATWG-Standard integriert
- Zeichnen auf den Canvas mit JavaScript (JS), Animation der Zeichnungen ebenfalls mit JS möglich
- Unterstützte Zeichenelemente z.B.
 - Linien
 - Rechtecke
 - Kreise
 - Bögen
 - Bézier-Kurven
 - Farbverläufe
 - ...



<http://canvasrider.com>

Zeichenfläche – Canvas (2/2)

Beispiel: Zeichnen eines blauen Quadrats

The screenshot shows a web browser's developer tools interface. The 'Elements' panel on the left displays the HTML structure:

```
<html>
  <head></head>
  <body>
    <canvas id="example" width="250" height="250">
      Your browser does not support HTML 5 Canvas.
    </canvas>
    <script type="text/javascript">
      // Access the 2D canvas element by variable
      var example = document.getElementById('example');
      var context = example.getContext('2d');

      // Set fill color blue
      context.fillStyle = "rgb(0,0,255)";
      // draw square
      context.fillRect(10,10,100,100);
    </script>
  </body>
</html>
```

The 'Styles' panel on the right shows the default user agent styles for the body element:

```
body {
  display: block;
  margin: 8px;
}
```

Below the styles, a box model diagram is visible, showing a blue square (1001 x 250) with a green padding area and an orange margin area. The diagram indicates a margin of 8px and padding of 8px.

Weitere Features und Browserunterstützung

- HTML 5 führt eine ganze Reihe weitere Features ein, z.B.
 - neue Tags wie z.B. `<time>`, `<meter>` oder `<progress>`
 - komplette Übersicht: <http://www.w3schools.com/tags/> oder <http://www.selfhtml5.org/>
- HTML 5 streicht Reihe von Tags aus HTML 4.01, vor allem im Bereich der Formatierung, z.B.
 - `<center>`, `<font>`, `<u>`, `<frame>`, `<frameset>`, ...
- Ein großer Teil der HTML 5-Features werden heute von aktuellen Versionen der gängigen Browser unterstützt – aber noch läuft nicht alles überall
 - Übersicht: <http://caniuse.com/>