

Doing Web Apps

HTML5 2

Autor: Rüdiger Marwein
Letzte Änderung: 2014-10-24
Version: 1.1

Dieses Dokument darf – mit Nennung des Autoren - frei vervielfältigt, verändert und weitergegeben werden.

Der Inhalt ist sorgfältig recherchiert, mit dem Dokument ist jedoch keinerlei Garantie auf Fehlerfreiheit gewährleistet.

Dieser Inhalt ist unter einem Creative Commons Namensnennung Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by/2.0/de/> oder schicken Sie einen Brief an Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Inhaltsverzeichnis

1. Bilder.....	3
2. Container-Elemente.....	3
3. Die Attribute „id“ und „class“	4
4. Document Object Model.....	5
5. Arbeiten mit dem DOM.....	5

1. Bilder

Bilder werden über das `` Tag eingebunden. Pflicht-Attribut ist hierfür die URL des einzubindenden Bildes: „src“.

Eine Bild-URL ist ein relativer oder absoluter Pfad oder eine vollständige URL.

```



```

Es lassen sich nicht alle Bildformate in eine Webseite einbetten. Bildformate, die von allen Browsern unterstützt werden sind:

- jpg
- gif
- png
- svg

Weitere wichtige Attribute des `` Tags sind

width	Breite	Zahl wie „231“ (implizit Pixel) oder prozentual wie „50%“
height	Höhe	Zahl wie „231“ (implizit Pixel) oder prozentual wie „50%“
alt	Alternativ-Text	Für Suchmaschinen und Blinde, beschreibt den Bildinhalt
title	Titel-Text	Erscheint in Tooltip.
align	Ausrichtung am Text	„left“ oder „right“. Mit „left“ wird das Bild links angeordnet und entsprechend rechts vom Inhalt umflossen.

Breite und Höhe verhalten sich proportional, wenn man nur eines der beiden Attribute angibt. Gibt man Höhe und Breite an und missachtet die Proportionen (aspect-ratio), wird das Bild gestaucht oder gestreckt.

```

```

2. Container-Elemente

Es wird zwischen Inline-Elementen und Blocklevel-Elementen unterschieden.

Inline-Elemente sind beispielsweise ``, `` und ``.

Blocklevel Elemente nehmen je die volle zur Verfügung stehende Breite ein und forcieren einen Umbruch zum nächsten Element. Beispiele sind `<p>`, ``, ``, `<form>`, `<h1>` - `<h5>`. Sie lassen sich als rechteckige Box beschreiben.

Innerhalb eines `<p>` Tags sind keine weiteren Blocklevel-Elemente erlaubt.

Mit Container-Elementen wird die Struktur einer Webseite definiert. HTML5 sieht hierfür unter anderem diese Tags vor:

<code><div></code>	Formatloser Container mit beliebigem Einsatzzweck
<code><header></code>	Container für den Kopfbereich der Webseite
<code><footer></code>	Container für den Fußbereich der Webseite

2.Container-Elemente

<article>	Container für einen geschlossenen Inhalts-Block, bspw einer News, einer Stellenausschreibung oder einer Informations-Seite.
<aside>	Container für Zusatzinformation
<nav>	Container, der eine Navigation beinhaltet.
<section>	Container für einen gekapselten Bereich.

Die HTML5-Boilerplate beschreibt einen exemplarischen Aufbau einer HTML5 Seite mit einigen der o.g. Tags wie folgt (reduziert):

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8" />
  <title>Der Titel der Webseite</title>
</head>
<body>
  <div id="container">
    <header>
      Hier steht die Kopfzeile
    </header>
    <div id="main">
      Hier steht der Haupt-Inhalt
    </div>
    <footer>
      Hier steht die Fußzeile
    </footer>
  </div>
</body>
</html>
```

3.Die Attribute „id“ und „class“

An jedem HTML-Element können die Attribute „id“ und „class“ vergeben werden. Diese schließen die Lücke zur besonderen Gestaltung mit CSS. Es ist dann nicht nur möglich auf Tag-Ebene zu gestalten sondern gezielt ausgezeichnete Elemente.

Wird das Attribut „id“ vergeben, muss es in einem Dokument einmalig mit seinem Namen sein. Der Name ist frei wählbar. Zur Verwendung in CSS sollte er jedoch nur aus Buchstaben, Zahlen und Unterstrichen oder Minuszeichen bestehen.

```
<div id="current-page">Impressum</div>
```

Die „id“ kann auch als Sprungmarke zu einer Position in einer Webseite fungieren. Der Browser scrollt direkt an die Position des Elements (sofern bis zum Element gescrollt werden kann).

HTML: <h1 id="suche">Suche</h1>

URL: meineSeite.html#suche

Das Attribut „class“ gibt einem Element oder Bereich ein „Thema“. Der Name ist frei wählbar und darf mehrmals vorkommen. Zur Verwendung in CSS sollte er jedoch nur aus

Buchstaben, Zahlen und Unterstrichen oder Minuszeichen bestehen.

HTML: `<div class="produkt"> ... Produkt Eigenschaften ... </div>`

Die Namen (sowohl für „class“ als auch „id“) sollten so gewählt sein, dass sie das Thema widerspiegeln, nicht die konkrete Umsetzung. Auch sollte der Name nicht zu allgemein sein.

gut: produkt
schlecht: div-w320px
gut: teaser-title
schlecht: green-headline
gut: title-teaser
schlecht: box

4. Document Object Model

Die Tags mit ihren öffnenden und schließenden Tags bilden eine Baum-Struktur. Das Document Object Model (kurz DOM) ist ein Synonym für diesen Baum und dessen Äste bzw Knoten (Node).

Jeder Knoten im DOM hat bestimmte Eigenschaften und wird in seiner Ausprägung u.U. vom Eltern-Element beeinflusst.

```
<strong>Fett und <em>fett-kursiv</em></strong>
```

Mit Stylesheets werden Eigenschaften einzelner Knoten im DOM konfiguriert / gestaltet.

Mit JavaScript wiederum werden die Eigenschaften einzelner Knoten im DOM dynamisch geändert und die Struktur des Baumes durch neue Knoten oder Löschung von Knoten manipuliert.

5. Arbeiten mit dem DOM

Um sich ein Bild des DOM Baums zu machen, können in den verschiedenen Browsern unterschiedliche Werkzeuge verwendet werden:

Firefox → Rechtsklick in die Seite, Element untersuchen

DOM, CSS, Live-Änderungen HTML/CSS, JavaScript Debugging, Ausgabe-Konsole, Fehlerkonsole, geladene Dateien und HTTP-Header inkl Zeitmessung

Firefox Plugin „FireBug“

DOM, CSS, Live-Änderungen HTML/CSS, JavaScript Debugging, Ausgabe-Konsole, Fehlerkonsole, geladene Dateien und HTTP-Header inkl Zeitmessung, diverse Erweiterungen

Chrome → Rechtsklick in die Seite, Element überprüfen

DOM, CSS, Live-Änderungen HTML/CSS, JavaScript Debugging, Ausgabe-Konsole, geladene Dateien und HTTP-Header inkl Zeitmessung

Safari → Rechtsklick in die Seite, Element untersuchen

DOM, CSS, Live-Änderungen HTML/CSS, JavaScript Debugging, Ausgabe-

5.Arbeiten mit dem DOM

Konsole, geladene Dateien und HTTP-Header inkl Zeitmessung

Internet Explorer 9 → Entwickler-Tools

DOM, CSS, Minimal Live-Änderungen HTML/CSS, JavaScript Debugging, Ausgabe-Konsole, JavaScript-Profilier, geladene Dateien und HTTP-Header inkl Zeitmessung, Browser-Modus 7,8,9,10 einstellbar, Cookies und Cache deaktivieren

Opera → Rechtsklick -> Element untersuchen

DOM, CSS, Live-Änderungen HTML/CSS, JavaScript Debugging, Ausgabe-Konsole, geladene Dateien und HTTP-Header inkl Zeitmessung

