



HTML5 / CSS3

Whitepaper

haase & martin
Neue Medien Dresden

Inhalt

HTML 5

Allgemeines	3
Unterscheide zu vorherigen Standards	3
Geänderte und undefinierte Elemente	3
<i>Diese Elemente sollen mehr ihrer ursprünglichen Bedeutung entsprechen</i> <i>Dokumententypenangabe</i>	
Neue Elemente	4
<i>Elemente die Bereiche kennzeichnen / Strukturelemente</i>	
<i>Semantische Elemente</i>	
<i>Formularelemente</i>	
<i>Multimediaelemente</i>	
<i>Welcher Browser unterstützt welchen Video-Codec?</i>	
<i>Welcher Browser unterstützt welchen Audio-Codec?</i>	
Application Programming Interface (API)	11

CSS 3

Neue CSS3 Selektoren – Möglichkeit HTML-Elemente anzusprechen	12
<i>Attribut-Selektoren</i>	
<i>Strukturelle Pseudo-Klassen</i>	
<i>Negations-Pseudo-Klassen</i>	
<i>Pseudo-Elemente</i>	
<i>:target-Pseudoklasse</i>	
<i>Pseudoklassen für UI-Element-Zustände</i>	
<i>Generelle Geschwister-Kombinatoren</i>	
Media Query	14
<i>Eigenschaften (allgemein)</i>	
<i>Auswahl von Eigenschaften mit kurzer Erläuterung</i>	
<i>Wo wird es benutzt?</i>	
Funktionelle Notationen	16
CSS3 Eigenschaften	17
<i>Border</i>	
<i>Background</i>	
<i>Color</i>	
<i>Texteffekte</i>	
<i>User-Interface</i>	
<i>Mehrspaltiges Layout</i>	
<i>Webfonts</i>	
<i>Transformation</i>	
<i>3D Transformation</i>	
Transition und Keyframe Animationen	26
<i>Transition</i>	
<i>Keyframe-Animation</i>	
CSS3 im mobilen Internet	28
Quellen	31

HTML 5

Allgemeines

HTML5 tritt die Nachfolge von HTML4 an. Die Dokumentbeschreibungsstandards HTML4.01, XHTML1.0 und DOM HTML Level 2 werden ersetzt und neue Funktionalitäten werden integriert.

Im Mai 2011 erhielt HTML5 beim W3C den „Last Call“. Der Status „Last Call“ bedeutet dass HTML5 faktisch bereits einen fertigen Zustand angenommen hat. Was mit einem RC (Release Candidate) vergleichbar ist. Die meisten Browser haben es bereits, wenn auch unvollständig, implementiert.

Unterschiede zu vorherigen Standards

Im Allgemeinen wurden die Elemente vollständig (bis auf wenige Ausnahmen) aus HTML4 übernommen. Dazu zählen unter anderem:

- `<frameset>` ist veraltet und entspricht nicht mehr dem Standard
- Tags zum Gestalten, wie ``, `<big>`, `<center>`, `<s>`, ... - alle Formatierungen sollen über CSS erfolgen
- Tags, deren Bedeutung durch andere Elemente gegeben ist, wie `<acronym>`, `<applet>`, `<dir>`

Geänderte und undefinierte Elemente

Diese Elemente sollen mehr ihrer ursprünglichen Bedeutung entsprechen

<code><i></code>	Text der in der Regel kursiv geschrieben wird, wie Gedanken, veränderte Stimmung, ...
<code></code>	Text der stilistisch hervorgehoben werden soll aber keine höhere Wichtigkeit hat wie der umgebende Text
<code></code>	Textbetonung wird verstärkt durch die Verschachtelung des Elements
<code><hr></code>	Thematischer Umbruch zwischen Absätzen in einem Text
<code><small></code>	Kennzeichnung von Kleingedrucktem, wie Rechtshinweise oder zusätzliche Preisauszeichnungen
<code></code>	Auszeichnung eines Textes mit sehr hoher Bedeutung

Dokumententypenangabe:

```
<!DOCTYPE html>
```

- sehr vereinfacht
- Groß- und Kleinschreibung spielen keine Rolle

HTML5 ist abwärtskompatibel und somit bleibt das Dokument valide, da es jeder Browser verstehen kann.

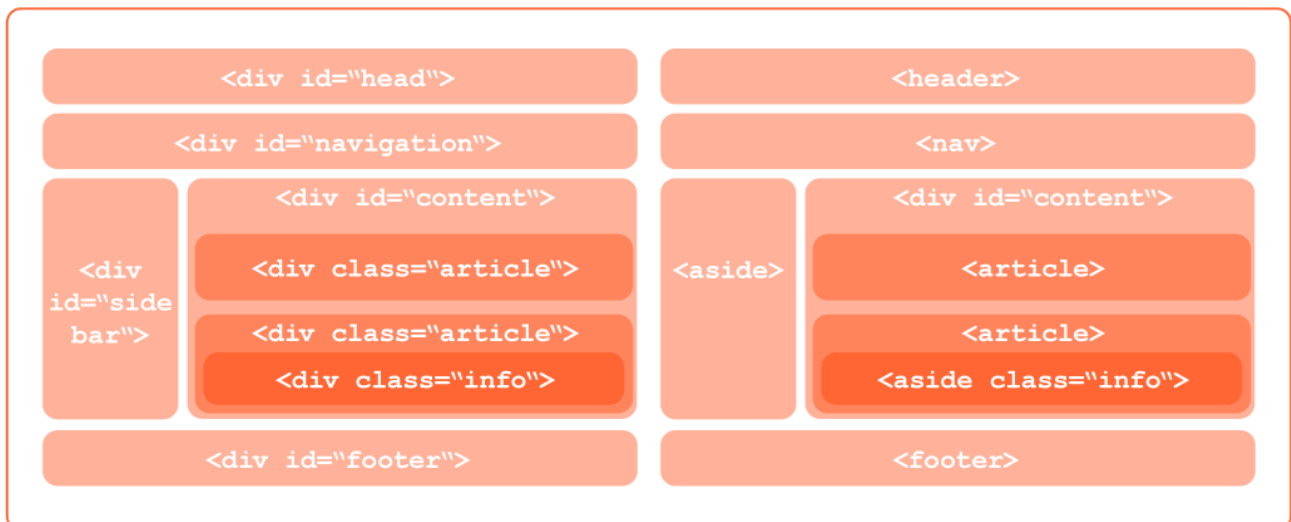
Neue Elemente

Die neuen Elemente werden von älteren Browsern (z.B. Internet Explorer 6) nicht erkannt. Um diese Elemente zu erkennen und darzustellen wird JavaScript benötigt.

Welche Browser diese Elemente unterstützen, kann unter <http://www.findmebyip.com/litmus/> oder <http://html5test.com/> nachgelesen werden. Eine umfangreiche Zusammenfassung, was mobile Browser derzeit leisten können, ist unter <http://mobilehtml5.org/> zu finden. Diese tabellarische Übersicht wurde von Maximiliano Firtman zusammengetragen.

Elemente die Bereiche kennzeichnen / Strukturelemente

Standardstruktur die auf „fast“ jeder Webseite verwendet wird. Deswegen wurden viele dieser IDs als Elemente übernommen.



<article>	Zeichnet einen inhaltlich eigenständigen Teil einer Seite aus. Inhalt sollte auch außerhalb der Seite einen Sinn ergeben, wie z.B. ein Forum-Post, ein Magazin- oder Zeitungsartikel, ein Blogbeitrag oder ein Leserkommentar.
<aside>	Erläuternde oder weiterführende Inhalte, die die Hauptinhalte ergänzen. innerhalb eines <article>-Elements: ein Block, der nicht mit dem Inhalt in Beziehung steht, aber nicht Teil des Artikels sein muss, z.B. Infobox außerhalb des <article>-Elements: ein Block, der inhaltlich mit der Webseite in Beziehung steht, z.B. Sidebar
<footer>	Fußbereich einer Seite, es sind auch mehrere <footer> pro Seite möglich (kann auch Element eines Artikels sein, z.B. Datum der Veröffentlichung). Er beinhaltet Informationen z.B. über Autor, urheberrechtliche Daten, <pre data-bbox="547 949 1414 1301"><article> Inhalt der Arikels <footer> Infos zum Artikel (Autor, Veröffentlichungsdatum, ...) </footer> </article> <footer> Infos zur Webseite (haftungsausschluss, Sitemap, Kontakt, ...) </footer></pre>
<header>	Definiert den Kopfbereich einer Webseite oder eines Sektionselements. Gruppe von einführenden oder navigatorischen Hilfen, z.B. Logo, Inhaltsangabe, Kopf eines Abschnitts mit <h1>-<h6> Elementen oder <hgroup> <pre data-bbox="547 1576 1414 2018"><header> Header mit Logo, Inhaltsangabe, ... </header> <articel> <header> <hgroup> <h1>Titel des Artikels </h1> <h2>Untertitel</h2> </hgroup> <header> <p> ... </p> </articel></pre>

<code><hgroup></code>	Gruppe einer <code><h1></code> - <code><h6></code> Elementenreihe, wie Titel & Untertitel.
<code><nav></code>	Einheit mit wesentlichen Navigationspunkten, entweder Links zu anderen Seiten oder Sprunganker innerhalb der selben Seite.
<code><section></code>	Gruppe von Elementen mit thematischer Zusammengehörigkeit, z.B. einzelne Register oder einzelne Bereiche einer Webseite, wie Einleitung, Newsbereich oder Kontaktinformationen. Kein semantisch leeres, bedeutungsloses Element – es ist kein Container, um Elemente fürs Styling zusammenzufassen, dazu sollte weiterhin <code>div</code> verwendet werden.
<code><figure></code>	Eine geschlossene Einheit von Inhalten, die mit dem Artikel in Verbindung steht aber vom Artikel entfernt werden kann, ohne ihn inhaltlich zu beeinträchtigen, z.B. Diagramme, Illustrationen, Tabelle <pre><figure> <figcaption> Bildunterschrift </figcaption> </figure></pre>
<code><figcaption></code>	Bildunterschrift, Legende, Untertitel, etc. für das <code><figure></code> -Element

Einige Elemente (`<section>`, `<nav>`, `<article>`, `<aside>`) bringen eine Funktion in der Kombination mit den Überschriftenelementen `<h1>` bis `<h6>` mit. Die Überschriftenhierarchie wird nicht anhand der Überschriftenelemente bestimmt, sondern auch anhand der Position innerhalb der neuen Elemente.

Semantische Elemente

<code><details></code>	Wird zusammen mit <code><summary></code> -Element genutzt, beschreibt Details und ist standardmäßig nicht sichtbar.
<code><mark></code>	Markiert eine Textstelle, um den Leser darauf aufmerksam zu machen (kann per CSS individuell gestaltet werden).
<code><meter></code>	Einsetzbar für Messwerte und numerische Angaben, z.B. Festplattennutzung, Relevanz eines Suchergebnisses, etc. Nicht verwendbar für beliebige Angaben, die nicht in Bereiche eingeordnet werden können.

```
<p> Your score is: <meter> 2 out of 10 </meter></p>

<meter> 80% </meter>
<meter> 3 /4 </meter>

<p>
  Christmast is in
  <meter value="30" min="1" max="366"
    title="days">
    30 days!
  </meter>
</p>
```

`<progress>`

Markiert einen Fortschritt einer Aufgabe. Bei einem unbestimmten Fortschritt ist unklar wie viel Arbeit noch besteht, bevor eine Aufgabe beendet werden kann, z.B. das Programm wartet auf eine Antwort vom Server. Ein bestimmter Fortschritt ist ein numerischer Wert zwischen 0 und einem gegebenen Maximum.

`<summary>`

Kopfbereich für das `<details>`-Element

```
<details>
  <summary>HTML 5 </summary>
  This document teaches you everything you have to
  learn about HTML 5.
</details>
```

`<time>`

Wird für die Angabe von Uhrzeit und/oder Datum verwendet. Die eingetragene Zeit wird vom Browser in unterschiedlichen Zeitzonen automatisch umgerechnet.

```
Das Eröffnungsspiel ist am <time datetime="2011-06-26">26.Juni</time>.
```

```
Wir öffnen um <time>10:00</time> jeden Tag.
```

```
<time datetime="2011-07-21T02:00+09:00">
20:00Uhr, Morgen, Tokyoer Zeit
</time>
```

```
Veröffentlicht am:
<time atetime="2011-07-20" pubdate>
  20.      Juli
</time>
```

`<ruby>`

Ruby Annotation (chinesische Zeichen): wird in Ostasien verwendet, um die Aussprache anzuzeigen. Wird immer mit `<rt>`- oder `<rp>`-Element verwendet.

`<rt>`

eigentliche Information

`<rp>`

Kennzeichnet was angezeigt werden soll, wenn der Browser das Zeichen nicht kennt.

Formularelemente

`<command>`

Ist ein Radiobutton, eine Checkbox oder eine Standard Schaltfläche und muss sich innerhalb des Menü-Items befinden.

```
<menu>
  <command="command">
    Click Me!
  </command>
</menu>
```

`<detailist>`

Ermöglicht es Vervollständigungsvorschläge anzugeben und wird immer im Zusammenhang mit dem `<input>`-Element benutzt.

```
<input list="Obst" />
<deailist id="Obst">
  <option value="Apfel">
  <option value="Birne">
  <option value="Pflaume">
  <option value="Kirsche">
</detailist>
```

`<keygen>`

Zur Erstellung von Identifikations- Schlüsselpaaren.

Verschiedene Einstellungen können über Attribute vorgenommen werden:

- `autofocus` - Der Focus zum automatischen Laden kann auf "enable" oder "disable" gestellt werden.
- `challenge` - Mit dem Parameter "challenge" wird die Nutzung der Schlüssel vorgeschrieben, wenn der öffentliche Schlüssel übertragen wurde.
- `disabled` = Der Parameter "disabled" deaktiviert die Schreibrechte des Nutzers für das Formularfeld mit dem öffentlichen Schlüssel.
- `form` = Das Feld mit dem Schlüssel wird mit diesem Parameter dem angegebenen Formular hinzugefügt.
- `keytype` = Hier wird der Verschlüsselungstyp "rsa" als Parameter für die Erstellung eines RSA Schlüssels angegeben.
- `name` = Der Feldname des Schlüsselfelds wird als Parameter angegeben.
- Außerdem kommen Universalattribute, wie `class`, `lang`, `tabindex`, etc. oder Eventattribute , wie `onabort`, `onchange`, etc verwendet werden.

```
<form action="http://www.test.html">
  <keygen keytype="rsa" name="securekey" challenge="challenge">
  <input name="securekey" value="Schlüssel" />
</form>
```

`<output>`

Gegenstück zum `<input>`-Element, z.B. für Ausgabe, Ergebnis einer Rechnung

Multimediaelemente

`<audio>`

Einbinden von Musik und/oder Audio-Streams

<code>autoplay</code>	Startet Player, wenn vollständig geladen
<code>controls</code>	Kontrollelemente ein- bzw. ausblenden
<code>loop</code>	immer wieder abspielen
<code>preload</code>	Startet das Laden, wenn Webseite geladen ist
<code>src</code>	Quellpfad, Quelldatei

```
<audio controls="controls">
  <source src="horse.ogg" type="audio/ogg" />
  <source src="horse.mp3" type="audio/mp3" />
  Your browser does not support the audio element.
</audio>
```

`<canvas>`

Leinwand (Container/freie Fläche), um Bitmaps per JavaScript hinein zu zeichnen

<code>height</code>	Definiert Höhe des <code><canvas></code> -Elements
<code>width</code>	Definiert Breite des <code><canvas></code> -Elements

```
<canvas id="myCanvas"></canvas>

<script type="text/javascript">
  var canvas=document.getElementById('myCanvas');
  var ctx=canvas.getContext('2d');
  ctx.fillStyle='#FF0000';
  ctx.fillRect(0,0,80,100);
</script>
```

Eine ausführliche Beschreibung gibt es vom Webdesigner, Frontendentwickler und Buchautor Peter Kröner unter:
<http://www.peterkroener.de/eine-kleine-canvas-einfuehrung/>.

`<embed>`

Ein allgemeiner Behälter zum Einfügen externer Inhalte, der nicht mit dem `audio`-, `img`- oder `video`-Element realisiert werden kann. Zur Darstellung dieser Inhalte wird normalerweise ein Browser-Plugin erforderlich sein. Verwendet wird es beispielsweise für Flash-movies oder Quicktime Movies.

`<source>`

Quellen für Medien-Elemente, wie `<audio>` und `<video>`

<video>

Element zum Einbinden von Filmen und/oder Video-Streams.

audio	Default-Zustand
autoplay	Startet Player, wenn vollständig geladen
controls	Kontrollelemente ein- bzw. ausblenden
height	Definiert Höhe
loop	immer wieder abspielen
poster	Anzeige eines Bild, bevor Play gedrückt wurde
preload	Startet Laden, wenn Webseite geladen
src	Quellpfad, Quelldatei
width	Definiert Breite

Welcher Browser unterstützt welchen Video-Codec?

BS	Win										Mac				
	Firefox		Safari	Internet Explorer						Chrome	Opera	Firefox	Safari		Opera
Version	3.6	4	5	6	7	8	9	10	10	11.1	4	5	11.1		
ogg/theora	✓	✓							✓	✓	✓			✓	70%
H.264			✓				✓	✓	✓			✓			38%
WebM		✓							✓	✓	✓			✓	31%

Quelle: <http://www.findmebyip.com/litmus/> Stand: 12.10.2011

Welcher Browser unterstützt welchen Audio-Codec?

BS	Win										Mac				
	Firefox		Safari	Internet Explorer						Chrome	Opera	Firefox	Safari		Opera
Versinon	3.6	4	5	6	7	8	9	10	10	11.1	4	5	11.1		
ogg/vorbis	✓	✓							✓	✓	✓			✓	71%
mp3			✓				✓	✓	✓			✓			38%
wav	✓	✓	✓						✓	✓	✓	✓	✓	✓	69%
aac			✓				✓	✓	✓			✓			38%

Quelle: <http://www.findmebyip.com/litmus/> Stand: 12.10.2011

Eine Auflistung aller HTML5 Tags findet sich unter: <http://w3schools.com/html5/>

Application Programming Interface (API)

APIs sind nicht wirklich Teil der HTML5-Spezifikation. Sie werden aber mit HTML5 in Zusammenhang gebracht.

Offline Storage	Daten einer Webseite können auf dem Computer oder Smartphone lokal gespeichert werden. Diese Seite kann auch offline genutzt werden.
Local Storage	Lokales Abspeichern von Schlüssel- und Wertepaaren. Ermöglicht das Ablegen von wesentlich größeren Datenmengen als bei Cookies. Webapps können Daten vorab in den lokalen Speicher laden (z.B. blitzschnelles Anzeigen beim Wechsel auf die nächste Seite.)
Geolocation	Zugriff auf Positionsdaten über GPS oder mittels WiFi und IP Auswertungen über einen Google Service.
App Caching	Legt fest daß der Browser die Daten, einer Webseite, solange nicht neu herunterlädt, bis sich der Inhalt geändert hat. Somit wird der Start der mobilen Webanwendung beschleunigt.
Forms	Gib die Möglichkeit einer lokalen Validierung der eingegebenen Daten an ohne daß sie an den Server übermittelt werden.

weitere APIs:

- Drag&Drop
- History
- Inline Editing
- Messaging
- Selectors
- Server Events
- Web Sockets
- Web Workers

Ausführliche Informationen und Codebeispiele in Safari Developer Center (kostenlose Anmeldung ist notwendig).

CSS 3

Neue CSS3 Selektoren – Möglichkeit HTML-Elemente anzusprechen

Attribut-Selektoren

<code>a[href^="http://"]</code>	Diese Definition trifft beispielsweise für alle <code><a></code> -Elemente zu, bei denen das <code>href</code> -Attribut mit „http://“ beginnt.
<code>img[src\$=".gif"]</code>	Diese Definition trifft beispielsweise für alle <code></code> -Elemente zu, bei denen die Zeichenkette mit „.gif“ endet.
<code>a[title*="wikipedia"]</code>	Diese Definition trifft beispielsweise für alle <code><a></code> -Elemente zu, bei denen die Zeichenkette „wikipedia“ enthält.

Strukturelle Pseudo-Klassen

<code>:root</code>	Trifft auf Hauptelement zu.
<code>tr:nth-child /</code> <code>tr:nth-child(even)</code>	Trifft auf alle <code><tr></code> -Elemente zu, die das 2. Kind ihres Elternelements sind. Bezieht sich in diesem Beispiel auf gerade und ungerade Tabellenzeilen: <code>tr:nth-child(2n+1) / tr:nth-child(odd)</code> .
<code>tr:nth-last-child</code> <code>(-n+2)</code>	Trifft auf die zwei letzten Zeilen einer Tabelle zu.
<code>img:nth-of-type(2n)</code>	Trifft auf jedes zweite <code></code> -Element zu.
<code>li:last-child</code>	Trifft auf das letzte <code></code> -Element einer Liste zu.

Negations-Pseudo-Klassen

<code>video:not([autoplay])</code>	Alle Videos, die nicht das Attribut „autoplay“ besitzen.
<code>*:not(.error)</code>	Trifft auf alle <code><div></code> -Elemente zu, deren Klasse nicht „error“ lautet.
<code>html *:not</code> <code>(:link):not(:visited)</code>	Trifft auf alle HTML-Elemente zu, außer auf Links.

Pseudo-Elemente

<code>p:first-line</code>	Trifft auf die erste Zeile eines Absatzes zu.
<code>p.newChaper::first-letter</code>	Trifft auf den ersten Buchstaben eines Absatzes der Klasse „new-Chaper“ zu.
<code>.Note::before{content: „Beachte: „}</code>	Trifft auf alle Elemente der Klasse „Note“ zu und fügt deren Inhalt ein „Beachte:“ hinzu.
<code>::after</code>	Trifft auf Elemente nach dem Inhalt zu.

:target-Pseudoklasse

Manche URLs beziehen sich auf eine bestimmte Stelle auf der Seite, z.B. `.../index.html#section2`. Dieser Teil kann mit der `:target`-Pseudoklasse angesprochen werden.

<code>p:target{color:red}</code>	Trifft auf alle <code><p></code> -Elemente, die ein Zielelement eines Aufrufs sind zu.
----------------------------------	--

Pseudoklassen für UI-Element-Zustände

<code>input:enabled</code>	Trifft auf alle <code><input></code> -Elemente zu, die nicht „deaktiviert“ sind.
<code>input:disabled</code>	Trifft auf alle <code><input></code> -Elemente zu, die „deaktiviert“ sind.
<code>input:checked</code>	Trifft auf „angehakte“ Radiobuttons zu.

Generelle Geschwister-Kombinatoren

<code>h1 ~ p</code>	<p>Zwei Elemente die das selbe Elternelement haben.</p> <p>Trifft auf das zweite Element zu, welches dem ersten Element nicht unmittelbar voran gehen muss.</p> <p>z.B. Alle <code><p></code>-Elemente, denen ein <code><h1></code>-Element auf selber Ebene voran geht.</p> <pre><h1> Titel </h1> <div> ... </div> <p> Absatz </p></pre>
---------------------	---

Media Query

Besteht aus einem Medium-Typ und mindestens einem weiteren Ausdruck, der den Geltungsbereich einschränkt.

z.B.: `media="only screen and (color)"` - Diese Bestimmung trifft nur für Farbbildschirme zu.

Ermöglicht die Anzeige von Inhalten entsprechend der Beschaffenheit der Ausgabegeräte, ohne daß der Inhalt verändert wird.

Eigenschaften (allgemein):

- width
- height
- device-height
- device-width
- orientation
- aspect-ratio
- device-aspect-ratio
- color
- color-index
- monochrome
- resolution
- scan
- grid

Auswahl von Eigenschaften mit kurzer Erläuterung:

width	<code>@media print and (min-width: 25cm)</code> <code>@media screen and (min-width: 400px) and (max-width: 700px)</code> <code>@media handheld and (min-width: 20em), screen and (min-width: 20em)</code>
orientation	Die Ausrichtung ist <code>portrait</code> , wenn Höhe größer oder gleich der Breite ist, sonst ist sie <code>landscape</code> . <code>@media all an (orientation:portrait)</code> <code>@media all an (orientation:landscape)</code>
device-aspect-ratio	Seitenverhältnis <code>@media screen and (device-aspect-ratio: 16/9)</code> <code>@media screen and (device-aspect-ratio: 32/18)</code> <code>@media screen and (device-aspect-ratio: 1280/720)</code>
resolution	Auflösung <code>@media print and (min-resolution: 300dpi)</code> <code>@media print and (min-resolution: 118dpcm)</code>

Wo wird es benutzt?

HTML	<pre><link rel="stylesheet" media="screen and (color), projection and (color)" rel="stylesheet" href="example.css"></pre>
XML	<pre><?xml-stylesheet media="screen and (color), projection and (color)" rel="stylesheet" href="example.css" ?></pre>
@import	<pre>@import url(example.css) screen and (color), projection and (color);</pre>
@media	<pre>@media screen and (color), projection and (color) { ... }</pre>

Funktionelle Notationen

`attr()`

Zugriff auf einen Wert innerhalb eines Attributs.

```
<stock>
  <wood length="12" />
  <wood length="5" />
  <metal length="19" />
  <wood length="4" />
</stock>
stock::before {
  ...
  ...
}
stock > * {
  display: block;
  width: attr(length, em); /* default 0 */
  height: 1em;
  .
  .
  .
}
```

`counter()`

Benennung einer Variablen, welche durch `counter-increment` erhöht werden kann und durch `counter-reset` zurückgesetzt werden kann. So können beispielsweise Kapitel durchnummeriert werden.

```
p{
  counter-increment: par-num
}

h1 {
  counter-reset: par-num
}

p:before {
  content: counter(par-num, upper-roman) "."
}
```

`calc()`

Diese Funktion `calc()` dient z.B. zur Spaltenberechnung, mit Abstand zwischen den Spalten.


```
div.spalten {
  float: left;
  margin: 1em;
  width: calc(100%/3-2*1em);
}
```

CSS3 Eigenschaften

Das CSS3 befindet sich noch in der Entwicklung, doch die Browser implementieren bereits die CSS-Eigenschaften. Um kenntlich zu machen daß diese Eigenschaften noch nicht standardisiert sind setzen die Browser ihren Präfix davor.

-khtml-	KHTML: Konqueror
-webkit-	WebKit: Safari, Chrome, Arora, Mobile Safari, etc.
-moz-	Gecko: Firefox, Mozilla, SeaMonkey, etc.
-o-	Presto: Opera
-ms-	Trident: Microsoft Internet Explorer

Border

border-radius	<p>Definition von Abgerundeten Ecken.</p> <pre>border-radius: 5px;</pre> <p>Jede Ecke kann individuell abgerundet werden.</p> <pre>border-top-left-radius – linke obere Ecke border-top-right-radius – rechte obere Ecke border-bottom-left-radius – linke untere Ecke border-bottom-right-radius – rechte untere Ecke</pre>
border-colors	<p>Es ist jetzt möglich mehrere Farben zu vergeben, somit kann ein Verlauf erzeugt werden.</p> <pre>border: 8px solid #00; border-colors: #ff6633 #ff7141 #ff7c50 #ff865e #ff916c #ff9b7a #ffa689 #ffb197;</pre>  <p>Die border-colors-Eigenschaft kann auch auf verschiedene Seiten angewendet werden.</p>

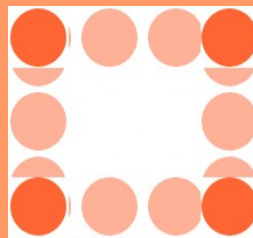
border-image

Diese Eigenschaft definiert als Rahmen ein Bild.

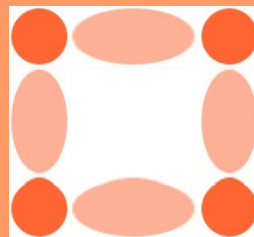
```
border-image:  
  border-top-image  
  border-right-image  
  border-bottom-image  
  border-left-image  
border-corner-image:  
  border-top-left-image  
  border-top-right-image  
  border-bottom-right-image  
  border-bottom-left-image
```

Des Weiteren ist es möglich ein Bild als Rahmen zu zerschneiden, dabei gibt es folgende Optionen: `stretch`, `repeat` und `round`.

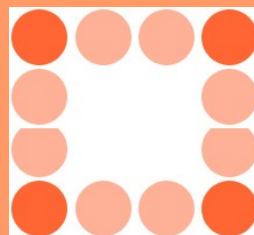
`repeat` – Das Bild wird einfach wiederholt.



`stretch` – Die Kantenstücke werden auf die volle Breite bzw. Höhe gestreckt.



`round` – die Kantenstücke werden etwas verzogen damit nichts vom dem Bild abgeschnitten werden muss.



shadow

Legt einen Schatten für ein HTML-Element fest.

```
box-shadow: x-Verschiebung y-Verschiebung Weichzeichengrad Farbe;
```

```
box-shadow: 10px 10px 5px black;
```

Background

background

Mit CSS3 können mehrere Hintergrundbilder für ein Element verwendet werden. Diese werden durch Komma getrennt.

```
background: url(body-top.gif) top left no-repeat,  
url(banner.gif) top 11px no-repeat;
```

Dadurch entstehen kleinere Datenmengen und somit eine schnellere Downloadrate.

background-size

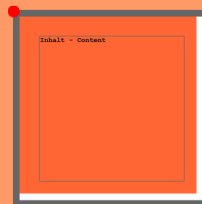
Ermöglicht das Skalieren eines Hintergrundbildes.

```
background-size: 412px 320px;
```

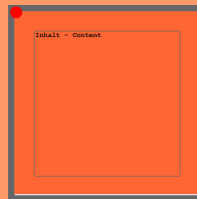
background-origin

Definiert den Ursprung für background-position.

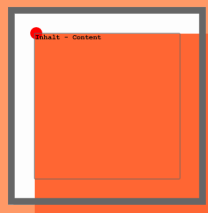
```
background-origin: border-box;  
Ursprung im Rahmen.
```



```
background-origin: padding-box;  
Ursprung an der Innenseite des Rahmen.
```



```
background-origin: content-box;  
Ursprung an der oberen linken Ecke des Inhaltsbereichs.
```

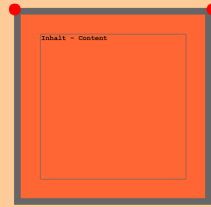


background-clip

Legt einen Hintergrundbereich fest.

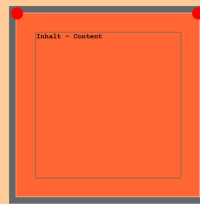
```
background-clip: border-box;
```

Darstellungsbereich ab Randaußenseite.



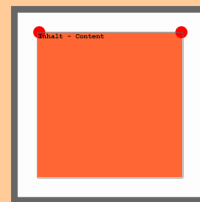
```
background-clip: padding-box;
```

Darstellungsbereich ab Randinnenseite.



```
background-clip: content-box;
```

Darstellungsbereich ab Inhaltsbereich.



Color

In CSS3 kann die Farbdefinition nicht nur über den Hexadezimalwert erfolgen, sondern auch über die RGB-Werte und die HSL-Werte. Des Weiteren ist es möglich den Alphawert der Farbe mit anzugeben.

alte Definitionsmöglichkeit	#800080
neue Definitionsmöglichkeiten	rgb(128, 0, 128) hsl(300, 100, 25)

opacity	Bestimmt die Dreckkraft eines Elementes (1 = 100% und 0 = 0%). <code>opacity: 0.8;</code>
gradient	Definiert Farbverläufe, wobei zwischen linearen und radialen Farbverläufen unterschieden wird. <code>div{ background: linear-gradient(red, blue); }</code> Mit verschiedenen Parametern kann, z.B. ein Startpunkt definiert werden. <code>div{ background: radial-gradient(bottom, red, blue); }</code> <code>div{ background: linear-gradient(-45deg, red, blue); }</code>

Texteffekte

text-shadow

Diese Eigenschaft definiert einen Textschatten, die Funktionalität ist gleich der `box-shadow`-Eigenschaft.

```
text-shadow: x-Verschiebung y-Verschiebung  
Weichzeichengrad Farbe;  
text-shadow: 6px 4px 10px black ;
```

text-overflow

Text der durch `overflow: hidden;` nicht angezeigt wird, kann jetzt durch diese Eigenschaft gekennzeichnet werden und fügt Auslassungspunkte an das Textende.

Werte:

`clip` – schneidet überstehenden Text ab

`ellipsis` – fügt Auslassungspunkt an, Punkte werden hinter letzten in die Zeile passenden Buchstaben gesetzt

`ellipsis-word` – fügt Auslassungspunkte an, Punkte werden hinter das letzte in die Zeile passende Wort gesetzt

`text-overflow-mode` – für `clip` | `ellipsis` | `ellipsis-word`

`text-overflow-ellipsis` – hat standardmäßig den String (“...“) | möglich ist auch `url()`

word-wrap

Lange Zeichenfolgen, wie eine lange URL, können in die nächste Zeile umgebrochen werden.

Werte:

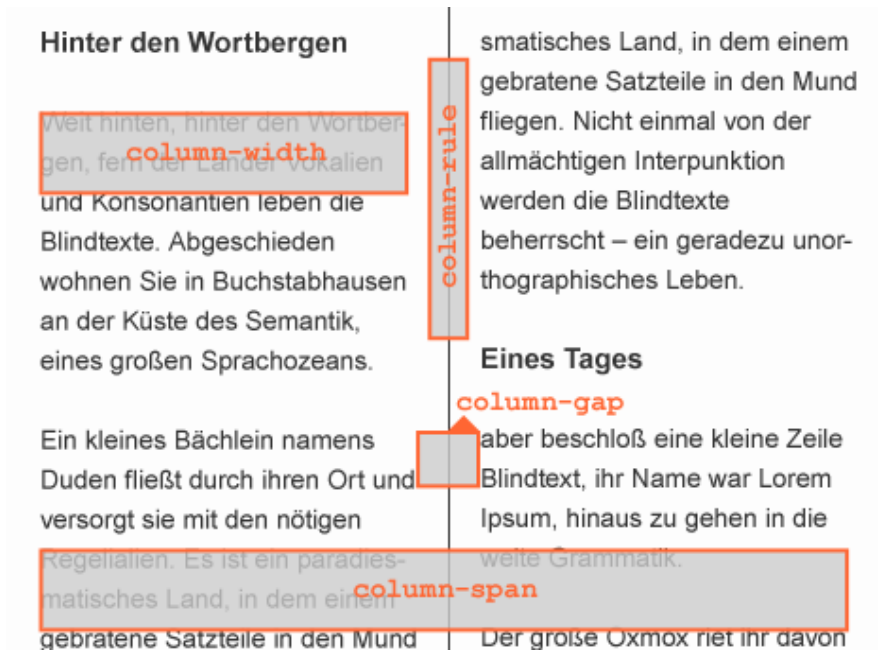
`break-word` – bricht das Wort um

`normal` – das Word wird nicht umgebrochen

User-Interface

<code>box-sizing</code>	<p>Definiert den Bezugspunkt der Breite einer Box.</p> <p>Werte:</p> <ul style="list-style-type: none"><code>border-box</code> – Breite bezieht sich auf die gesamte Box (mit Rahmen und Abständen)<code>content-box</code> – Breite bezieht sich nur auf den Inhalt<code>inherit</code> – Breite hat gleichen Bezugspunkt wie das Elternelement
<code>resize</code>	<p>Gibt dem Nutzer die Möglichkeit einzelne Elemente, wie Textareas zu vergrößern oder zu verkleinern.</p> <p>Um die Skalierbarkeit einzuschränken sollte <code>min-width, max-width</code> und <code>min-height, max-height</code> verwendet werden.</p> <p>Werte:</p> <ul style="list-style-type: none"><code>both</code> – Elemente in beide Richtungen zu verändern<code>horizontal</code> – Veränderung nur in horizontaler Richtung<code>vertical</code> – Veränderung nur in vertikaler Richtung
<code>outline</code>	<p>Wird verwendet um Aufmerksamkeit auf ein Element zu gelenken. Die Eigenschaft verschiebt nicht die Struktur, es überlagert alle anderen Elemente und steht somit an erster Stelle.</p> <pre>outline: 2px solid blue; outline-offset: 12px; (definiert den Abstand zum Element)</pre>

Mehrspaltiges Layout



column-width	Definiert die Breite einer Spalte.
column-gap	Definiert den Abstand zwischen den Spalten.
column-rule	Definiert eine Trennlinie zwischen den Spalten (Bestimmung wie die Eigenschaft border, Farbe, Art und Breite kann bestimmt werden).
column-count	Definiert die Anzahl der Spalten.
column-span	Definiert ein Element das sich über alle Spalten hinweg erstreckt.

Webfonts

Besondere Schriftarten mussten in älteren CSS-Versionen immer als Bild eingebunden werden. Mit CSS3 ist es möglich die Schriftart selbst mit einzubinden.

```
@font-face{
  font-family: Bezeichnung;
  src: url ('Quellpfad');
}

@font-face{
  font-family: MyHelvetica;
  src: local ("Helvetica Neue Bold"), local ("HelveticaNeue-Bold"),
      /* local - Überprüft ob Schrift auf Rechner vorhanden ist*/
      url (MgOpenModernaBold.ttf);
  font-weight: bold;
}
```

Vorteile:

- Skalierbarkeit: bei Vergrößerung keine Verschlechterung der Qualität
- schnellerer Download (Mobiles Web): Schriftdatei meist kleiner als ein Bild
- Barrierefrei: Lesbar von Screenreadern
- Suchmaschinenoptimiert: ist von Suchmaschinen lesbar

Transformation

Elemente können verschoben (`translate`), skaliert (`scale`), gedreht (`rotate`) und gekrümmt (`skew`) werden. Diese Eigenschaft kann auf alle Elemente angewendet werden.

```
div{
  transform:
    skew (30deg, -10deg)
    rotate (30deg)
    translate (100px, 20px)
    scale(0.8);
}
```

Der Transformationsursprung liegt standardmäßig im Mittelpunkt der Objektes, kann aber mit der Eigenschaft `transform-origin` selbst definiert werden.

3D Transformation

Besitzt die selben Eigenschaften wie `transform`, es muss lediglich hinter jeder Eigenschaft `3d` ergänzt und die dritte Koordinate (`z`) hinzugefügt werden.

```
translate3d(tx,ty,tz)
scale3d(sx,sy,sz)
roate3d(ax,ay,az)

transform-origin: x y z;

perspective (wert);           (Perspektive des 3D-Elements)
persective-origin             (Ursprung der Perspektive)
```

Weitere Eigenschaften und visuelle Beispiele unter: <http://www.webkit.org/blog/386/3d-transforms/>

Transition und Keyframe Animationen

Animationen helfen dem User die Zusammenhänge und das Verhalten des Systems besser zu erfassen. Wenn beispielsweise ein blinkender Cursor stehen bleibt, weiß der Nutzer daß das System noch nicht bereit ist.

Zudem Verbessern bewusst gewählte Animationen das Aussehen einer Webseite und mit CSS3 sind diese leicht und mit wenig Programmieraufwand zu realisieren.

Transition

<code>transition-properties</code>	Bestimmt welche CSS Eigenschaft animiert werden soll.
<code>transition-duration</code>	Bestimmt die Dauer der Animation.
<code>transition-timing-function</code>	Bestimmt die Art der Animation (<code>linear</code> , <code>ease-in</code> , <code>ease-out</code> , <code>ease-in-out</code>).
<code>transition-delay</code>	Bestimmt die Verzögerung vor dem Start.

Folgende Eigenschaften können animiert werden:

- Transform
- Schrift-Eigenschaften (font size, line, letter/word spacing,...)
- Farben
- Schatten
- Durchsichtigkeit
- ...

Keyframe-Animation

Kommen zum Einsatz wenn die Animation, nicht wie bei Transition nur einen Anfangs- und Endzustand, sondern auch einzelne Zwischenzustände hat.

<code>@-keyframes name</code> <code>{ ... }</code>	Anlegen eines Keyframes
<code>animation-name: name</code>	Verwenden der Animation
<code>animation-duration</code>	Dauer der Animation

animation-iteration-count	Anzahl der Animationsdurchläufe
animation-direction	Umkehren der Animation
animation-timing-funktion	Art der Animation (linear, ease-in, ease-out, ease-in-out)

Beispiel:

Button-Animation – Button in Normalgröße, verkleinert sich, wird größer und geht wieder in Normalgröße über

```
.sessionList.newFavorite{
    -webkit-animation-name: favorite;
    -webkit-animation-duration: 0.3s;
    -webkit-animation-iteration-count: 1;
}

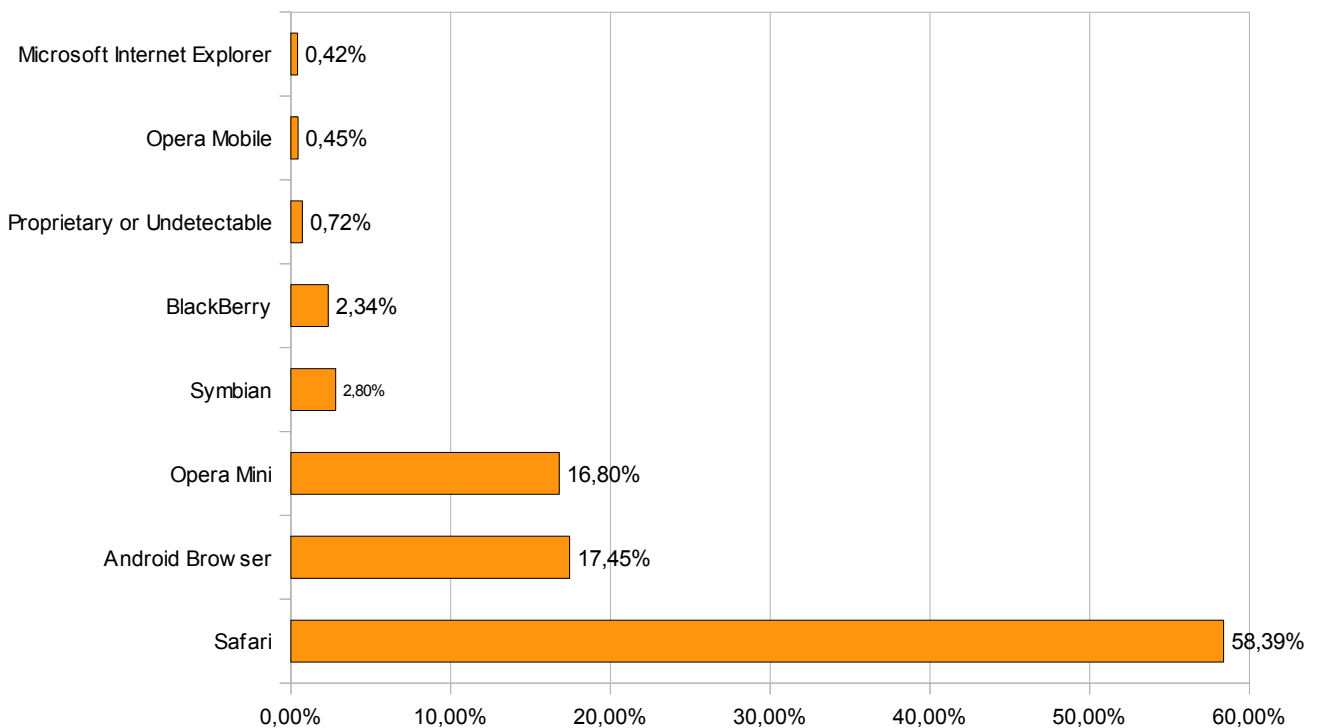
@-webkit-keyframes favorite{
    from {
        -webkit-transform: scale(1);
    }
    25%{
        -webkit-transform: scale(0.8);
        animation-timing-function: ease-out;
    }
    50%{
        -webkit-transform: scale(1.2);
        animation-timing-function: ease-in;
    }
    to{
        -webkit-transform: scale(1);
        animation-timing-function: ease-out;
    }
}
```

CSS3 im mobilen Internet

CSS3 bringt für das Mobile Web einige Vorteile. Durch die Einführung von Webfonts, Transformationen, 3D, Schatten, etc. können die Ladezeiten für Webseiten sehr verkürzt werden. Außerdem verringert sich der Programmieraufwand durch die CSS3 Animationen sehr.

Die Spezialisierung für mobile Webseiten ist WebKit denn sowohl iPhone als auch Android basieren darauf.

Marktanteile der Mobile- und Tablet-Browser



Quelle: Mobile/Tablet Browser Market Share (Calendar Q4, 2011),
<http://www.netmarketshare.com/browser-market-share.aspx?qprid=0&qpcustomd=1&qpct=2&qpmr=75&qptimeframe=Q>

box-reflect

Definiert eine Spiegelung unter einem HTML-Element.

```
-webkit-box-reflect: Richtung  
Abstand  
mask-box-image;
```

Richtung: **above, below, left, right**

Was angezeigt werden soll definiert `mask-box-image`.
Wenn es nicht angegeben wird, wird das Element 1:1 gespiegelt.

```
-webkit-box-reflect:below 1px -webkit-gradient  
(linear,left top, 0 100%,  
form(transparent),  
to(rgba(0,0,0,0.8)));
```



Lauftexte

Besonders für die keinen Displays der mobilen Endgeräte geeignet.

```
overflow-x: -webkit-marquee;
```

Wie der Lauftext sich bewegen soll, wird angegeben mit:

```
-webkit-marquee-direction: backwards;
```

ahead / up von unten nach oben

reverse / down von oben nach unten

brackwards / left von rechts nach links

forwards / right von links nach rechts

auto Voreinstellung

Wie oft sich der Lauftext bewegen soll, wird angegeben mit:

```
-webkit-marquee-style: alternate;
```

alternate – Schiebt den Text hin und her

none – Keine Bewegung

scroll – Wiederholt die vorgegebene Richtung

slide – Stoppt wenn der gesamte Text zu lesen ist oder die gegenüberliegenden Seite erreicht ist

Wie schnell sich der Lauftext bewegen soll, wird angegeben mit:

```
-webkit-marquee-speed: fast;
```

```
fast | normal | slow
```

```
Distanz / Zeit
```

Wie oft das Marquee ausgeführt werden soll, wird angegeben mit:

```
-webkit-marquee.increment: infinite;
```

```
Zahl | infinite - unendlich
```

Masken

Sind Elemente die über Objekte gelegt werden können, egal ob Text, Bild oder Video.

Eigenschaften sind analog zu Background:

```
-webkit-mask background
```

```
-webkit-mask-attachment background-attachment
```

```
-webkit-mask-clip background-clip
```

```
...
```

```
..
```

Arten von Masken:

```
url(image)
```

```
-webkit-gradient()
```

```
-webkit-canvas()
```

Quellen:

<http://de.wikipedia.org/wiki/HTML5>

<http://it-republik.de/jaxenter/artikel/HTML5-Das-mobile-Web-wird-erwachsen-3633.html>

<http://w3schools.com/html5/>

<http://www.findmebyip.com/litmus/>

<http://www.html5-portal.de/>

<http://www.webkrauts.de/2009/09/24/das-endoskelett-einer-webseite/>

<http://www.selfhtml5.org/>

<http://www.peterkroener.de/eine-kleine-canvas-einfuehrung/>

<http://webkompetenz.wikidot.com/docs:html-handbuch>

<http://www.html-lernen.de/category/webdesign/html/html5/>

<http://www.html-seminar.de/html-5-strukturierende-elemente.htm>

<http://technikwuerze.de/podcast/technikwuerze169/>

<http://css3-html5.de/alle-videos-in-einer-playlist/>

Herausgeber:

Haase & Martin GmbH – Neue Medien Dresden
Moritzburger Str. 27
01127 Dresden

Tel.: 0351 – 500 97 21

E-Mail: info@haaseundmartin.de

Web: www.haaseundmartin.de

Cover: unter Verwendung einer Grafik von Sharpshot – fotolia.com