

Webseitengestaltung mit CSS

HTML (Hyper Text Markup Language) bleibt die Sprache, in der eine Webseite geschrieben ist. Die Inhalte werden nachwievor in ein HTML-Dokument geschrieben und mittels HTML-Tags strukturiert. Meta-Tags beinhalten die Informationen für Suchmaschinen. Bis zu diesem Punkt ändert sich durch die Ergänzung mit CSS nichts an der Arbeit bei der Webseitengestaltung.

Dynamisches HTML (engl. "Dynamic HTML" oder abgekürzt "**DHTML**") ist eine Erfindung von Marktstrategen, sagen Kritiker. In der Tat ist Dynamisches HTML keine klassische HTML-Erweiterung in Gestalt neuer HTML-Elemente. Es ist auch keine neue Sprache. Dynamisches HTML ist vielmehr der Sammelbegriff für verschiedene Lösungen, um dem Autor einer Web-Seite zu ermöglichen, Elemente der Webseite während der Anzeige dynamisch zu ändern, sei es automatisch oder durch Einwirken des Anwenders.

CSS (Cascading Stylesheets) ist eine unmittelbare Ergänzung zu HTML. Es handelt sich dabei um eine Sprache zur Definition von Formateigenschaften des HTML-Dokuments und einzelner HTML-Elemente. Sie können:

- CSS für ein einzelnes HTML-Tag anwenden,
- im Kopf einer HTML-Datei zentrale Definitionen zum Aussehen aller Tags einer Art
- oder Sie können Ihre Stylesheet-Definitionen sogar in separaten Dateien notieren.

Die Stylesheet-Dateien können Sie in beliebig vielen HTML-Dateien referenzieren. Auf diese Weise können Sie für große Projekte einheitliche Layouts entwerfen. Mit ein paar kleinen Änderungen in einer zentralen Stylesheet-Datei können Sie dann für Hunderte von HTML-Dateien ein anderes Aussehen bewirken.

Neben CSS gibt es auch andere Sprachen zur Gestaltung einer durch eine weitere Sprache vorgegebene Struktur. Bekannte Stil-Sprachen sind: DSSSL, XSL und XSLT. Sie sind für die Webseitengestaltung aber nicht von Interesse.

Genau wie HTML wird auch CSS vom W3-Konsortium normiert. Es handelt sich also um einen von Firmen unabhängigen, offen dokumentierten und frei verwendbaren Standard. Für CSS gibt es genau wie für HTML eine Arbeitsgruppe beim W3-Konsortium, die sich um die Weiterentwicklung der Sprache kümmert und sich dabei dem Regelwerk zur Entstehung der so genannten Recommendations (=Empfehlungen) des W3-Konsortiums unterzieht.

Die Version 1.0 von CSS wurde bereits im Jahr 1996 herausgegeben. 1998 kam die Version 2.0 heraus, die bis 2011 zu **CSS Level 2 Revision 1 (CSS 2.1)** erweitert wurde. Auf der Website des W3-Konsortiums sind alle Befehle von CSS 2.1 aufgelistet:

<http://www.w3.org/TR/CSS2/>

An der Version 3.0 wird noch gearbeitet. Seit dem 29. September 2011 hat die Sammlung der Selektoren der Generation 3.0 den Status einer Empfehlung. Selektoren sind aber nur ein Teil des gesamten Sprachumfangs. Die Selektoren des Sprachstandards CSS 3 sind auf der Website des W3-Konsortiums nachzulesen:

<http://www.w3.org/TR/css3-selectors/>

Bei HTML gibt es gerade den Sprung von verschiedenen Versionen (HTML 4.01; XHTML 1.1; im Entstehen begriffenes XHTML 2.0) zu einem – hoffentlich – wieder allgemein gültigen HTML 5.

Doch alle Überlegungen zu aktuellen und neuesten Sprachstandards darf nicht die Gedanken an den Besucher einer Website überlagern, der mit einem eventuell einige Jahre alten Computer, auf dem ein lange nicht aktualisierter Browser installiert ist. Man muss also unbedingt eine Website mit HTML und CSS auf verschiedenen Browsern (Internet Explorer, Mozilla Firefox, Google Chrome, Safari, usw.) testen. Dieses Verfahren nennt sich **Cross Browser Check**.

Literatur-Tipps

Lehrgänge zum Selbststudium

HTML 4 für Dummies

Ed Tittel & Mary Burmeister
mitp-Verlag
ca. 12,00 €

Webdesign für Dummies

Lisa Lopuck
mitp-Verlag
ca. 30,00 €

Einstieg in CSS – Grundlagen und Praxis

Elisabeth Wetsch
Galileo Computing
ca. 20,00 €

CSS – Das umfassende Handbuch (behandelt CSS 2 und CSS 2.1)

Eric A. Meyer
O'Reilly Verlag
ca. 40,00 €

Das Beste an HTML & CSS: Best Practices für standardkonformes Webdesign

Ben Henick & Jorgen Lang
O'Reilly Verlag
ca. 35,00 €

Kurzreferenzen (Taschenbücher, Online)

HTML – kurz & gut

Jennifer Niederst

CSS – kurz & gut

Eric A. Meyer

beide: O'Reilly Verlag
jeweils unter 15,00 €

SelfHTML Version 8.1.2

SelfHTML e.V. (ehemals Stefan Münz)

Kostenlos unter
<http://de.selfhtml.org/>

Tutorials, Validatoren, Paradebeispiel (Online)

HTML-World

arejo GmbH
<http://www.html-world.de/>

Stil mit {stil}

Ulrike Häbler
<http://www.mediaevent.de/css/>

bjoernsworld

Björn Höhrmann
<http://www.bjoernsworld.de/css/>

The Styleworks

Klaus Langenberg
<http://www.thestyleworks.de/>

Validatoren

<http://validator.de.selfhtml.org/>
<http://validator.w3.org/>
<http://jigsaw.w3.org/css-validator/>

Der CSS Zengarden

Dave Shea
<http://csszengarden.com/>

Allgemeine Informationen zum Computer und zum Internet

Duden – Basiswissen Schule

Angewandte Informatik

Dr. Lutz Engelmann (Hrsg.)
Dudenverlag
21,00 €

Taschenbuch Multimedia

Peter A. Henning
Fachbuchverlag Leipzig
ca. 20,00 €

Cascading Stylesheets (CSS)

Die Einbettung von CSS in ein HTML-Dokument

<pre><head> ... <style type="text/css"> <!-- Selektor {Attribut: Wert;} ... --> </style> ... </head></pre>	<pre><body> ... <p style="Attribut: Wert;..."> ... </p> ... </body></pre>
als Inhalt des Style-Tags, umschlossen von Kommentarzeichen	als Spezifikation in das entsprechende HTML-Tag
<pre><head> ... <style type="text/css"> @import url(datei.css); </style> ... </head></pre>	<pre><head> ... <link rel="stylesheet" type="text/css" href="datei.css"> ... </head></pre>
als Import innerhalb des Style-Tags	als Link auf ein externes CSS-Dokument

Bei Änderung/Ergänzung von Spezifikationen:
title="ein Name" — Sheet wird verwendet, bis vom Besucher ein anderes gewählt wird.
rel="alternate stylesheet" — Sheet wird nur verwendet, wenn es vom Besucher gewählt wird.

Syntax einer Regel

Jede Anweisung besteht aus einem **Selektor**, der sich auf ein HTML-Tag bezieht, und einer **Deklaration** in geschwungenen Klammern. Die Deklaration besteht aus mindestens einem **Attribut** und nach einem Doppelpunkt dem zugewiesenen **Wert**. Die Attribute werden mit Semikolon voneinander getrennt.

```
h1 {color: red; text-align: center; } /* Kommentar */
```

Mehreren Selektoren kann das gleiche Format zugewiesen werden.

```
h1, h2, p {color: green; }
```

Selektoren können kontextsensitiv verknüpft werden. Im folgenden Beispiel wird nur der gefettete Bereich des Absatzes blau dargestellt (Verkettung oder Nachfahren-Selektor).

```
p b {color: blue; }
```

Hier wird nur ein Absatz, der direkt auf eine Tabelle folgt, in Arial geschrieben (Geschwister-Selektor).

```
Table + p {font-family: Arial; }
```

Durch die HTML-Spezifikation **class** können verschiedene Selektoren zusammengefasst oder ein bestimmter Selektor isoliert werden.

```
In HTML: <h1>Warnung</h1><h1 class="achtung">Gefahr!</h1>
          <p class="achtung">Vorsicht vor dem Hund!</p>
```

```
In CSS:  h1 {text-align: center; } für alle h1-Tags
          h1.achtung {color: red; } für das h1-Tag der Klasse „achtung“
          .achtung {background: yellow; } für alle Tags der Klasse „achtung“
```

Pseudo-Klassen-Selektoren

Speziell für die Darstellung der unterschiedlichen Zustände von Links (in HTML mit `` erzeugt) stellt CSS einige Pseudo-Klassen bereit. D.h. sie müssen im HTML-Tag nicht benannt werden.

```
a:link {color: blue; }           a:visited {color: purple; }
a:active {color: red; }         a:hover {color: green; }}
```

Auch für andere Elemente gibt es Pseudoklassen (und Pseudoelemente), z.B.:

```
:focus {color: blue;} schreibt Elemente, die den Fokus erhalten, blau
:target {color: yellow;} schreibt das Ziel des geklickten Links gelb
:first-letter {color: red;} schreibt den ersten Buchstaben des Absatzes rot
:first-line {color: green;} schreibt die erste Zeile des Absatzes grün
```

Vererbung und Spezifität

Bei der Kaskadierung der Stildefinitionen kommt es auf zwei wichtige Bestandteile an; die Vererbung und die Spezifität. Man kann sich eine HTML-Datei wie eine Matryoschka vorstellen: `body` ist die Mutter von `table`, welche die Mutter `tr`-Tags ist, und diese wieder der `td`-Tags. Wenn wir `body` mit der Eigenschaft `color:blue` belegen, erben alle Kinder von `body`, also `table`, `tr` und `td` diese Eigenschaft. Einige Eigenschaften werden allerdings nicht vererbt, beispielsweise: `background`, `border`, `margin` und `padding`.

Die Spezifität (wie speziell sich eine Stildefinition auf ein Tag bezieht und angewendet wird) errechnet sich aus dieser Tabelle:

Art des Selektors	Spezifität
Universeller Selektor [*] und Kombinatoren [Verkettungen]	0,0,0,0
Selektoren für Elemente und Pseudoelemente [p]	0,0,0,1
Selektoren für Klassen und Pseudoklassen [.]	0,0,1,0
ID-Selektor [#]	0,1,0,0
Eingebettete Stildefinitionen [style="..."]	1,0,0,0

Werte und ihre Definitionen

Den Attributen der Elemente müssen Werte zugewiesen werden. Dies können Schlüsselwörter sein (wie `red`, `bold`, `center`), Maßangaben, Prozentwerte, Hexadezimalcode oder URLs.

Größen können absolut in Zentimeter, Millimeter, Inch aber auch in Punkt und Pica und in Pixel angegeben werden. Die Standardschrift eines Geschäftsbriefs liegt bei 10–12 Punkt. Die Angaben in `cm`, `mm`, `in` werden nur sehr ungenau umgesetzt. Pixel werden von den meisten Browsern als Bildschirmpixel interpretiert (z.B. 1027*768 Auflösung).

```
10mm = 1cm = 0,394in
1in = 2,54cm = 25,4mm

12pt (Punkt) = 1 pc (Pica) [= 1/6 Inch]
72pt = 6pc [=1 Inch]
```

Sie können die Größe eines Elements auch relativ angeben. Dafür stehen Ihnen Prozentangaben zur Verfügung oder die Einheit `em`. Prozent können positiv und negativ vorkommen und im Betrag auch deutlich über 100 hinausgehen. Sie verhalten sich immer relativ zu anderen Werten.

```
{line-height: 150%} setzt den Zeilenabstand auf das anderthalbfache der Zeilenhöhe
{line-height: 1.5em} erzielt dasselbe Ergebnis
```

Farbangaben können Sie mit Schlüsselwörtern (die 16 VGA-Farben auf Englisch) machen, mit Prozenten (0–100) oder in Hexadezimalcode (0–F oder 0–255). Die beiden letzten Methoden arbeiten mit den RGB-Kanälen. Bei Prozentangaben sind Dezimalwerte erlaubt (z.B. 75.5%).

```
#FF0000 = rgb(255,0,0) = rgb(100%,0%,0%) = red
#0000FF = rgb(0,0,255) = rgb(0%,0%,100%) = blue
```

Der Hintergrund

Wenn Sie einer Datei ein Hintergrundbild zuweisen, denken Sie an die Lesbarkeit des Textes und an die Dateigröße. Relative Links werden relativ zur CSS-Datei nicht zur HTML-Datei interpretiert.

<code>body {background-image: url(bild.jpg); }</code>	
<code>background-attachment: WORT;</code>	bestimmt die Scrolleigenschaft des Hintergrundes scroll fixed Standardwert: scroll
<code>background-color: FARBE;</code>	benennt eine Hintergrundfarbe Standardwert: transparent
<code>background-image: URL;</code>	benennt ein Hintergrundbild Standardwert: none
<code>background-position: ZAHL</code>	Position des (ersten) Hintergrundbildes Werte: left center right top center bottom 0–100% 0–100% Standardwert: 0% 0%
<code>background-repeat: WORT</code>	Wiederholungsart des Hintergrundbildes Werte: repeat repeat-x repeat-y no-repeat Standardwert: repeat
<code>background: WORT ZAHL URL etc.</code>	Kurzschriftvariante für die <code>background</code> -Attribute Man kann auch einige oder alle Attribute in einer Kurzschriftvariante zusammenfassen.

Schriftformate

Um Schriften zu formatieren, gibt es eine ganze Reihe an Eigenschaften bzw. Attributen. Die wichtigsten sind hier aufgeführt:

<code>font-family: SCHRIFTNAME</code>	definiert eine Schriftart oder -familie z.B.: Arial, Garamond, Courier New, Lucida Console Standardwert: Browserabhängig
<code>font-size: WORT ZAHL</code>	setzt die Schriftgröße in % oder pt, pc, in, cm, em oder: xx-small ... medium ... xx-large oder: smaller larger Standardwert: medium, 1em, 100%
<code>font-style: WORT</code>	definiert eine Schrifteigenschaft italic oblique normal Standardwert: normal
<code>font-weight: WORT ZAHL</code>	definiert die Stärke der Schrift 100–900 normal bold bolder lighter Standardwert: 400 normal
<code>letter-spacing: ZAHL</code>	definiert einen Zeichenabstand em, px (negative Werte: enger) Standardwert: normal 0
<code>line-height: ZAHL</code>	definiert die Höhe der Zeilenbox em, % relativ zur <code>font-size</code> keine negativen Werte Standardwert: normal 1
<code>word-spacing: ZAHL</code>	definiert einen Wortabstand em, px (negative Werte: enger) Standardwert: normal 0
<code>font: ZAHL WORT etc.</code>	Kurzschriftvariante für die <code>font</code> -Attribute

Man kann auch mehrere Attribute in einer Kurzschriftvariante zusammenfassen. Es müssen aber mindestens `font-size` und `font-family` in ihr bestimmt sein. Will man darin auch die Zeilenhöhe angeben, muss sie mit einem Slash [/] getrennt direkt nach der Schriftgrößenangabe kommen.

Da wir nicht wissen, welche Schriften auf dem Rechner unseres Besuchers installiert sind, bietet es sich an, nicht nur eine Schriftart anzugeben, sondern von Komma getrennt eine Alternativschrift und danach auch noch den Namen einer generischen Schriftfamilie.

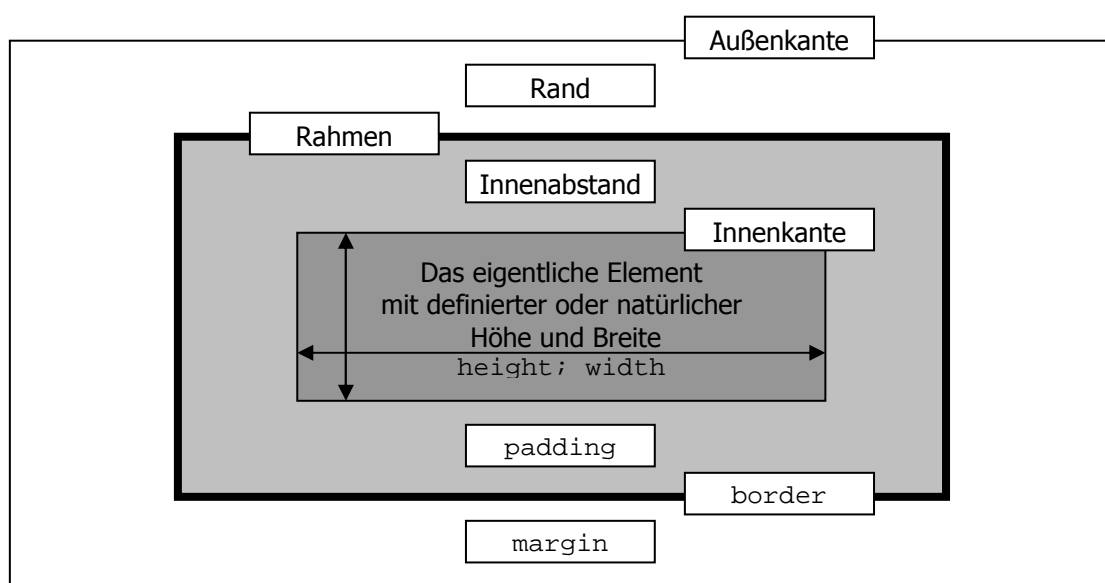
<code>serif</code>	Browser verwendet seine Standard-Antiquaschriftart (mit Serifen/Füßchen)
<code>sans-serif</code>	Browser verwendet seine Standard-Grotesqueschriftart (ohne Serifen/Füßchen)
<code>cursive</code>	Browser verwendet seine voreingestellte Kunstschrift- oder kalligrafische Schriftart
<code>monospace</code>	Browser verwendet eine nicht proportionale Schriftart
<code>fantasy</code>	Browser verwendet eine ausgefallene Schriftart. Das könnte auch Wingdings oder Webdings sein. Vorsicht bei Verwendung dieser Schriftartfamilie!

Elementklassifizierung

- Block-Elemente
Absätze (`p`), Überschriften (`h1-h6`), Listen (`ul, ol`), Tabellen (`table`), (`div`) und (`body`)
Sie beginnen in einer neuen Zeile und dulden kein Folge-Element in dieser Zeile.
- Inline-Elemente
Anker (`a`), Text hervorhebungen (`em`), (`span`) sowie die meisten Grafiken (`img`)
Sie erzeugen keinen Zeilenumbruch vor oder nach dem Element.
- Listenpunkt-Elemente
Die Punkte in einer Aufzählung (`li`) dienen der Zuweisung eines Gliederungszeichens oder der Durchnummerierung

Box- und Rahmeneigenschaften

Alle Elemente erzeugen um sich ein virtuelles Rechteck (Box). Die Formatierung eines Elements wirkt sich auf die gesamte Box aus.



- Der Hintergrund eines Elements (Grafik oder Farbe) erstreckt sich über Inhaltsbereich und Innenabstand bis zum Rahmen.
- Die Attribute `padding`, `border` und `margin` können so für das gesamte Element oder speziell für den rechten (`border-right`), linken (`-left`), oberen (`-top`) oder unteren (`-bottom`) Bereich bestimmt werden; bzw. `border-top-color`, `border-top-style` und `border-top-width`.
- Höhe (`height`), Breite (`width`) und der Rand (`margin`) dürfen auf den Wert `auto` gesetzt werden. Die Ränder dürfen zudem noch negative Werte annehmen.
- Die Innenabstände (`padding`) und der Rahmen (`border`) können keine negativen Werte annehmen. Sie sind standardmäßig auf 0 gesetzt.

<code>border: STYLE WIDTH COLOR</code>	Kurzschriftvariante für die <code>border</code> -Attribute
<code>border-color: FARBE {1,4}</code>	definiert die Element-Rahmen-Farbe Farbe des Elements
<code>border-style: WORT {1,4}</code>	definiert den Element-Rahmen-Stil Werte: <code>none dotted dashed solid double groove ridge inset outset</code> Standardwerte: <code>none</code>
<code>border-width: ZAHL {1,4}</code>	definiert die Element-Rahmen-Dicke Werte: <code>px</code>
<code>height: ZAHL auto</code>	definiert die Höhe eines Objekts in <code>px</code> Standardwert: <code>auto</code>
<code>margin: % ZAHL {1,4}</code>	definiert den Rand Standardwert: <code>0</code>
<code>padding: % ZAHL {1,4}</code>	definiert den Innenabstand Standardwert: <code>0</code>
<code>width: ZAHL auto</code>	definiert die Breite des Elements Standardwert: <code>auto</code>

Medienspezifische Stylesheets

Für verschiedene Ausgabemedien können auch unterschiedliche Stylesheets vorgehalten werden. Bei der Einbindung der Stylesheets muss die Abfrage nach dem Medium [`media`] ergänzt werden:

```
<link rel="stylesheet" media="print" href="example.css" />
```

oder:

```
@import url("example.css") screen;
```

Die Auswahlmöglichkeiten der Medien haben sich in den CSS-Generationen stark verändert. Sinnvoll sind aber durchaus: `braille`, `handheld`, `print`, `projection`, `screen`, `all` [ist Standard]

Auf folgende CSS-Attribute soll der Browser nur zurückgreifen, wenn sie für die Ausgabe über den Drucker benötigt werden.

<code>orphans: ZAHL</code>	Zeilenanzahl, die mind. am unteren Seitenrand verbleiben muss; Standardwert: 2
<code>page-break-after/before: Wort</code>	Wunschstelle für Seitenumbrüche beim Druck nicht ausdrücklich zu verbieten, aber zu erzwingen Werte: <code>auto</code> , <code>always</code> , <code>avoid</code> Standardwert: <code>auto</code>

widows: ZAHL Zeilenanzahl, die mind. am oberen Seitenrand verbleiben muss; Standardwert: 2

Auch Grafiken, deren Größenangabe (z.B. 600 Pixel Breite) für den Bildschirm gesetzt sind, werden mit einem auf 600 dpi eingestellten Drucker nur 2,5 cm breit sein. In den meisten Fällen wird die Ausgabe in diesem Fall skaliert, man darf aber nicht davon ausgehen.

Referenz weiterer CSS-Attribute (Auswahl)

bottom: ZAHL	Positioniert Objekte von unten, wenn <code>position</code> nicht <code>static</code> ist. Werte: Prozent, Pixel; Standardwert: auto
color: FARBE	Vordergrundfarbe des Elements Standardwert: Browserabhängig
display: WORT	Definiert Darstellungsart; Vorsicht! Werte: inline, block, list-item ...
float: WORT	Definiert ein umflossenes Objekt Werte: left, right; Standard: none
left: ZAHL	Positioniert Objekte von links, wenn <code>position</code> nicht <code>static</code> ist. Werte: Prozent, Pixel; Standardwert: auto
list-style: TYPE POSITION IMAGE	Kurzschriftvariante für die <code>list-style</code> -Attribute definiert Listen und Aufzählungen
list-style-image: URL	definiert Listen und Aufzählungen Werte: url(*.gif) Standardwert: none
list-style-position: WORT	definiert Listen und Aufzählungen Werte: inside outside Standardwert: outside
list-style-type: WORT	definiert Listen und Aufzählungen Werte: disc circle square decimal lower-/upper-roman lower-/upper-alpha none Mehr sind vorgesehen aber nicht immer unterstützt Standardwert: disc
overflow: WORT	definiert Verhalten bei zu viel Inhalt Werte: visible, hidden, scroll, auto Standardwert: visible
position: WORT	Definiert das Schema, nachdem positioniert wird Werte und ihre Bedeutung: <code>static</code> – im Fluss der HTML-Datei <code>relative</code> – in Beziehung zum Elternelement <code>absolute</code> – [wird immer block] fest zum Elternelement, wenn es nicht <code>static</code> , sonst zu <code>body</code> <code>fixed</code> – [wird immer block] fest im Viewport (Fenster); scrollt nicht mit Standardwert: <code>static</code>

<code>right: ZAHL</code>	Positioniert Objekte von rechts, wenn <code>position</code> nicht <code>static</code> ist. Werte: Prozent, Pixel; Standardwert: <code>auto</code>
<code>text-align: WORT</code>	richtet Blockelemente aus Werte: <code>left right center justify</code> Standardwert: <code>left</code>
<code>text-decoration: WORT</code>	benennt einige Texteffekte Werte: <code>none under/overline line-through blink</code> Es werden nicht immer alle unterstützt Standardwert: <code>none</code> Standardwert <code><a></code> : <code>underline</code>
<code>text-indent: ZAHL</code>	definiert eine Einrückung der ersten Zeile Standardwert: <code>0</code>
<code>text-transform: WORT</code>	steuert die Groß- und Kleinschreibung Werte: <code>capitalize upper/lowercase none</code> Standardwert: <code>none</code>
<code>top: ZAHL</code>	Positioniert Objekte von oben, wenn <code>position</code> nicht <code>static</code> ist. Werte: Prozent, Pixel; Standardwert: <code>auto</code>
<code>vertical-align: % WORT</code>	definiert die vertikale Ausrichtung der Grundlinie eines Elements Werte: <code>sub (text-)top middle (text-)bottom</code> Standardwert: <code>baseline</code>
<code>z-index: ZAHL</code>	Positioniert sich überlagernde Objekte auf der z-Achse Standardwert: <code>auto</code>

