

## 8 CSS – Cascading Style Sheets

Bisher sehen unsere Webseiten natürlich noch sehr langweilig aus, die Gestaltung übernimmt der Browser, weitere Gestaltungen haben wir noch nicht hinzugefügt. Dies geschieht jetzt mit CSS, Cascading Style Sheets.



### 8.1 Sinn und Zweck von Stylesheets

Stylesheets sind eine unmittelbare Ergänzung zu HTML. Es handelt sich dabei um eine Sprache zur Definition von Formateigenschaften einzelner HTML-Elemente. Mit Hilfe von Stylesheets können Sie beispielsweise bestimmen, dass alle Überschriften 1. Ordnung einen großen Schriftgrad aufweisen, in der Schriftart Helvetica, aber nicht fett erscheinen und mit einem Abstand von 1,75 Zentimeter zum darauffolgenden Absatz versehen werden. Angaben dieser Art sind mit reinem HTML nicht möglich.

Das ist aber nur der Anfang. Stylesheets bieten noch viel mehr Möglichkeiten. So können Sie HTML-Elemente – egal ob Textabsätze, Listen oder Formulare – mit einer eigenen Hintergrundfarbe, einem eigenen Hintergrundbild (Wallpaper) oder mit diversen Rahmen ausstatten. Sie können Elemente im Anzeigefenster des WWW-Browsers positionieren und Abstände definieren. Für Print-Layouts stehen Möglichkeiten zur Definition von Seiten-Layout und Textflusskontrolle bereit. Für die akustische Wiedergabe von Web-Seiten gibt es ein ganzes Arsenal an Eigenschaften, um die künstliche Sprachausgabe zu steuern. Einige CSS-Eigenschaften sind auch in der Lage, das Anzeigefenster des Browsers zu beeinflussen, so etwa das Aussehen des Cursors.

## 8.2 Einbinden von CSS

Wie können die CSS, also die Formatdefinitionen mit dem HTML Code „verbunden“ werden? Hierzu gibt es **drei** Möglichkeiten:

Bezug zum HTML-Code	Aussehen im Code	Vorteile
<b>zentral in einer separaten CSS-Datei</b>	<pre>&lt;title&gt;Titel der Datei&lt;/title&gt; &lt;link rel="stylesheet" href="styles.css"&gt; &lt;/head&gt;</pre> <p>Die CSS werden in einer externen Datei mit dem Namen styles.css abgelegt.</p>	<p>Klare Trennung von Inhalt und Layout</p> <p>Die CSS können für mehrere HTML-Dateien verwendet werden</p>
<b>Zentral für eine HTML-Datei im Kopf (head)</b>	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Titel der Datei&lt;/title&gt; &lt;style&gt; &lt;!-- ... Hier werden die Formate definiert, hier stehen jetzt die CSS ... --&gt; &lt;/style&gt; &lt;/head&gt; &lt;body&gt;Hier stehen die Inhalte &lt;/body&gt; &lt;/html&gt;</pre>	<p>Der Bezug zu dem HTML Code ist deutlich, diese Anweisungen werden nur in dieser HTML-Datei verwendet.</p>
<b>Direkt im HTML-Tag</b>	<pre>&lt;p style="font- family:Garamond"&gt;Text in Garamond&lt;/p&gt;</pre>	<p>Zusammenhang wird unmittelbar klar.</p> <p>Nachteil: Code wird relative unübersichtlich.</p>

Alle drei Arten können miteinander verbunden werden.

Die Stylesheet Angaben werden jeweils überschrieben, wenn mehrere Angaben zum gleichen Tag eingebunden werden.

## 8.3 Aufbau der CSS

Beispiel:

```
h1 { font-family: Calibri, Verdana, Arial, sans-serif;
      font-size: 18px;
      color: red;
    }
```

Selektor: Wofür soll die Formatdefinition gelten? Dies können Tags, aber auch Klassen oder id's sein.

Wichtige Syntax:

- die eigentliche Formatdefinition steht in geschweiften Klammern
- es können mehrere Definitionen angegeben werden
- diese werden jeweils mit ; abgeschlossen
- zwischen Eigenschaft und Wert steht jeweils :

Welche Eigenschaft soll verändert werden?

Angabe des jeweiligen Wertes.

Hier geschieht also folgendes:

Alle Überschriften der Kategorie 1 sollen die Schriftart Verdana verwenden, falls diese nicht vorhanden ist, die Schriftart Arial, wenn diese nicht vorhanden ist, die vom Browser eingestellte Schrift ohne Serifen.

Alle Überschriften der Kategorie 1 sollen außerdem die Schriftgröße 18px benutzen und rot dargestellt werden.

## Überschrift 1

Weit hinten, hinter den Wortbergen, fern der Länder Vok  
Konsonantien leben die Blindtexte. Abgeschieden wohnen  
Buchstabhausen an der Küste des Semantik, eines großen  
kleines Bächlein namens Duden fließt durch ihren Ort und  
den nötigen Regelialien. Es ist ein paradiesmatisches Lan

### 8.4 Erste Gestaltung mit CSS

→ Um die CSS aus dem Vorlagenordner mit unserer HTML-Datei zu verknüpfen, fügen wir oben im head folgende Zeile hinzu:

```
<link rel="stylesheet" href="css/styles.css">
```

- Erstellen Sie in Ihrem Ordner einen neuen Unterordner namens css und erstellen Sie darin eine Datei, die Sie styles.css nennen.
- Darin schreiben Sie folgende Codezeilen und speichern diese.
- Betrachten Sie das Ergebnis im Browser.

#### 8.4.1 Quellcode der CSS

```
body {  
    font-family: Calibri, Arial, sans-serif;  
}  
/* Typografie */  
h1,  
h2,  
h3 {  
    text-transform: uppercase;  
    text-align: center;  
    margin-bottom: 5px;  
    margin-top: 20px;  
    color: #0165a3;  
}  
h1 {  
    font-size: 1.8em;  
}  
h2 {  
    font-size: 1.6em;  
}  
img {  
    width: 100%;  
}
```

## 8.4.2 Erläuterung der Inhalte

CSS	Bedeutung / Auswirkung
<code>body {</code>	
<code>font-family: Calibri, Arial, sans-</code>	
<code>serif;</code>	
<code>}</code>	
<code>text-transform: uppercase;</code>	
<code>text-align: center;</code>	
<code>margin-bottom: 5px;</code>	
<code>margin-top: 20px;</code>	
<code>color: #0165a3;</code>	
<code>font-size: 1.8em;</code>	
<code>img { width:100%; }</code>	

## 8.5 Einheiten in CSS

Einheit	Bedeutung	Verwendung, Einsatz
<b>px</b>	Pixel	picture Element, absolute Angabe in Bezug auf das Ausgabegerät
<b>em</b>	em steht für die Breite eines Gevierts des Druckwesens. ungefähr – die Breite des Großbuchstabens M	relative Einheit bezogen auf die Schriftgröße des Elternelements 1.5em eineinhalb mal so groß wie die Grundschriftgröße Wird für die Schriften, aber auch für Breiten von Containern verwendet
<b>%</b>	Prozent	relative Einheit, wird für Breiten von Containern verwendet
<b>rem</b>	root em	Relativ zur Größe des Wurzel (root)-Elementes
<b>vw</b> <b>vh</b>	viewport width viewport height	Dabei entspricht eine Einheit vw oder vh gleich 1% der Bildschirm-Breite oder -Höhe. Eine Box mit der Höhe von 30vh nimmt wie erwartet die Höhe von 30% der verfügbaren Bildschirm-Höhe ein. Gut prinzipiell für den Einsatz von Schriftgrößen im responsiven Design:  <pre>h1 {   font-size: 2.7vw; }</pre>

## 8.6 Das verwendete Box-Modell

Die Breite und auch die Abstände von Elementen zueinander bringen den Webdeveloper regelmäßig ins Schwitzen. Grund dafür ist das sogenannte Box-Modell.

Ein Element hat demnach eine bestimmte Breite oder Höhe.

Dem Element kann ein sichtbarer Rand hinzugefügt werden (**border**).

Zwischen border und Inhalt kann ein Innenabstand sein (**padding**)

Der Außenabstand als Abstand zu den anderen Elementen nennt man **margin**.

Das Beispiel zeigt die Abstände an einem Beispiel auf der Webseite und in der Ansicht der Entwicklerkonsole.

Abstände vorhanden

Wilde Natur mit herrlichen Ausblicken - das alles entdecken Sie "Sea & Travel". Wir stellen ihnen die schönsten Küsten in Nordeuropa vor und informieren über spannende Wanderungen und entlegene Strände und Aussichtspunkte.

### UNSERE ANGEBOTE

• Ferienhäuser direkt am Meer

Ansicht mit der Entwicklerkonsole in Chrome



Stark übertriebenes padding und margin:



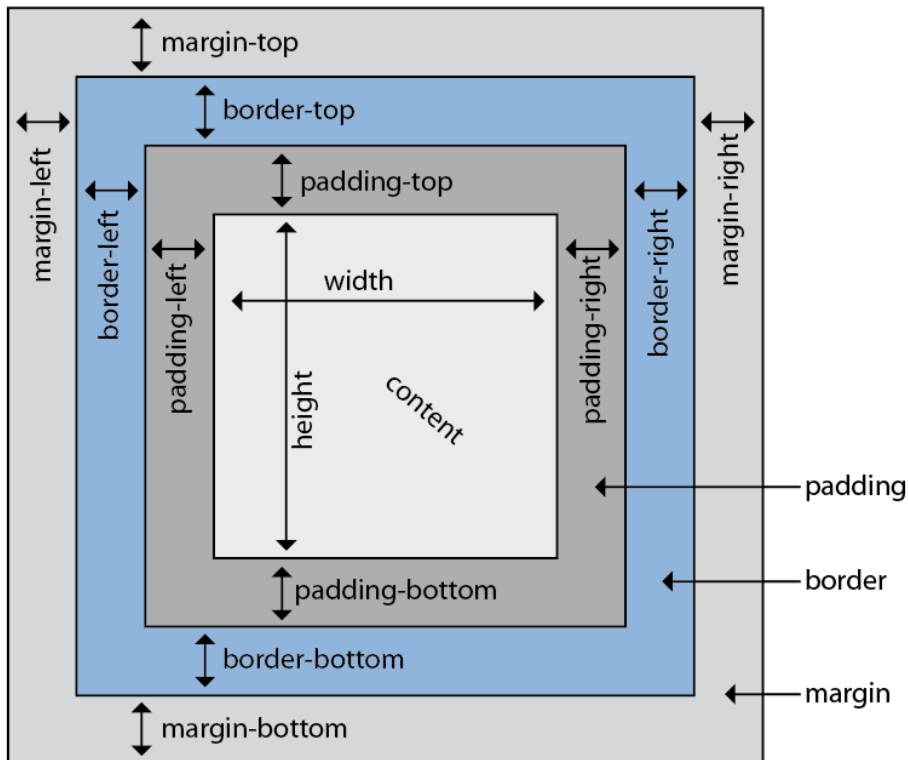
Dieses Box-Modell nennt man auch **content-box**

Alle Abstände können auch einzeln angesprochen werden. Die Gesamtbreite (Ausdehnung) eines Elements ergibt sich aus der width, dem padding, dem border und dem margin. Ein folgendermaßen definierter Container ist insgesamt 132px breit.

```
div {  
  width: 100px;  
  padding: 5px;  
  border: 1px solid #000;  
  margin: 10px; }
```

Außenabstand – margin  
Sichtbarer Rahmen – border  
Innenabstand – padding

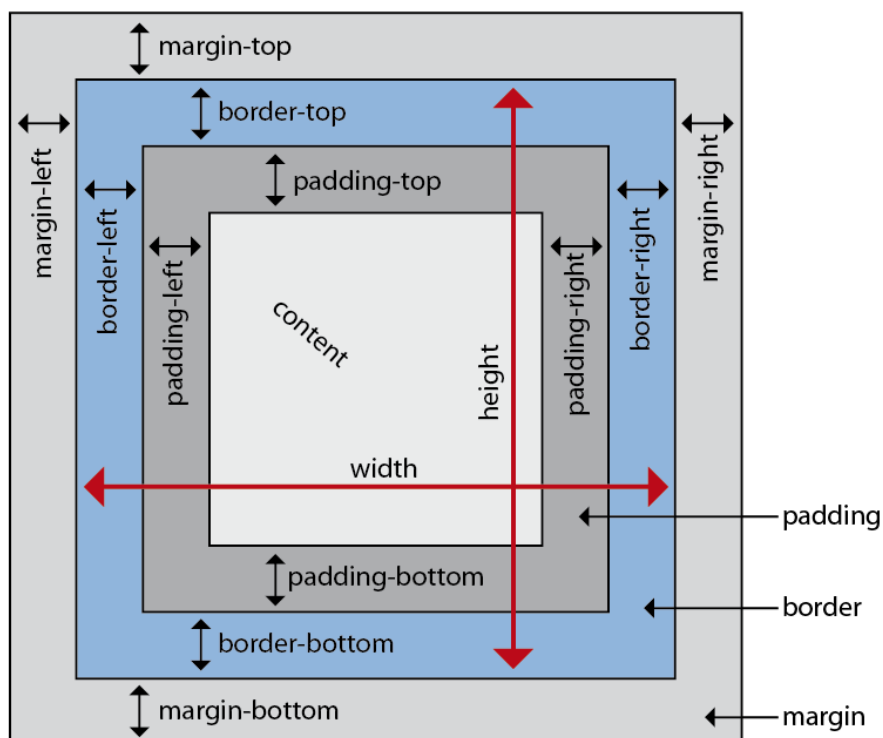
## 8.6.1 Darstellung des "Standard"-Box-Models box-sizing: content-box



## 8.6.2 Alternatives Box-Model box-sizing: border-box

Wenn man `box-sizing: border-box` hinzufügt, wird die Ausdehnung (z. B. `width`) einschließlich border und padding berechnet.

`box-sizing: border-box`



## 8.7 CSS – Vererbung

Viele Eigenschaften, die in CSS angegeben werden, werden auch an die Kinder vererbt.

Wenn ich für den ganzen Body eine Schriftfarbe angebe, haben alle Kindelemente diese Schriftfarbe.

Bei Abständen wiederum ergibt das keinen Sinn. Wenn Sie z. B. einen Container haben, der einen Innen- und Außenabstand haben soll, will man meistens nicht, dass auch alle Kindelemente diesen Innen- und Außenabstand haben.

Beispiel

CSS	Aussehen normalerweise Schriftfarbe und Schriftart werden vererbt, Abstände und Rahmen des Articles werden nicht vererbt.
<pre>body {     font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;     color: #aa3333; } article {     border: 1px solid #aa3333;     margin-left: 20px;     padding: 10px; }</pre>	<h2>Überschrift</h2> <p>Weiterer Text Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit debitis aliquam tempore porro eius dolorum quibusdam, aperiam, neque voluptas autem voluptates saepe, in explicabo? Reprehenderit necessitatibus nulla sed unde provident!</p> <div> <h3>Absatz über Frösche</h3> <p>Text über Frösche Lorem ipsum dolor sit amet consectetur adipisicing elit. Id exercitationem repudiandae aut commodi rerum fugiat error repellat rem fugit consequatur. Natus quidem numquam id! Quia non recusandae accusantium quo officia.</p> </div>
So würde das Ergebnis aussehen, wenn auch die Abstände und Rahmen des articles vererbt würden.	<h2>Überschrift</h2> <p>Weiterer Text Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit debitis aliquam tempore porro eius dolorum quibusdam, aperiam, neque voluptas autem voluptates saepe, in explicabo? Reprehenderit necessitatibus nulla sed unde provident!</p> <div> <h3>Absatz über Frösche</h3> <div> <p>Text über Frösche Lorem ipsum dolor sit amet consectetur adipisicing elit. Id exercitationem repudiandae aut commodi rerum fugiat error repellat rem fugit consequatur. Natus quidem numquam id! Quia non recusandae accusantium quo officia.</p> </div> </div>

Erzwingen kann man Vererbung mit  
Inherit

z. B. border: inherit

## 8.8 CSS-Selektoren und Spezifität

HTML-Elemente können auf sehr unterschiedliche Art ausgewählt, selektiert werden.

### 8.8.1 Selektoren

#### Einfache Selektoren

Selektornamen	Beispiel
<b>Universalselektor</b>	* { margin:0; } gleichbedeutend mit { margin:0; }
<b>Type-selector</b>	p { padding:0; }
<b>Klassenselektor</b>	.floatleft { float:left; margin-right:5px; }
<b>ID-Selektor</b>	#wrapper { width:1140px; margin: 0 auto; }
<b>Gruppenselektor</b>	h1, h2, h3 { font-family: Cambria, Times, serif; } Mehrere Elemente werden angesprochen

#### Pseudoelemente und -klassen

Selektornamen	Beispiel
<b>Pseudoklassen</b>	Elemente werden aufgrund eines Status oder einer Position angesprochen  a:hover { text-decoration:underline; } Beim Rüberfahren mit der Maus erscheint ein Unterstrich.  li:last-child { border-right:none; } Wenn das Listenelement letztes Kindelement der Liste ist, wird der rechte Rand entfernt
<b>Pseudoelemente</b>	befinden sich nicht im DOM Baum, also nicht als HTML Element vorhanden p::first-letter { font-size:1.8em; } p::first-line { font-variant:small-caps; } Der erste Buchstabe (letter) wird größer gemacht (eine Art Initiale) die erste Zeile eines Absatzes wird in Kapitälchen dargestellt



Selektieren im Dom-Baum Selektorname	Beispiele
<b>Descendant Selector</b>	<code>nav a { display:block; }</code> Befindet sich irgendwo innerhalb eines nav-Elementes ein a, wird dies angesprochen
<b>Kind Selektor, Child Selector</b>  >	<code>p &gt; a { color:red; }</code> Nur a als ein direktes Kind von p wird angesprochen.
<b>Adjacent and General Sibling Combinator +</b>	<code>p { text-indent: 40px; margin-top: .25em; margin-bottom: .25em; }</code> <code>h1 + p, h2 + p { text-indent: 0; }</code> Ein Absatz, der direkt einer Überschrift folgt, hat keinen Einzug.
<b>Allgemeine Geschwister Selektor, general sibling combinator ~</b>	<code>div ~ p</code>  Wenn irgendwo ein p einem div folgt, wird dieses angesprochen

Attributselektoren Selektorstruktur	Beispiele
<b>E[attr]:</b>	Im öffnenden Tag des Elementes gibt es ein Attribut mit dem Namen attr
<b>E[attr=val]</b>	Gilt für Elemente E, die ein Attribut mit genauen Wert val haben.
<b>E[attr^=val]</b>	Gilt für Elemente E, die ein Attribut attr haben, dessen Wert mit val beginnt. <code>article a[href^=http]:before, article a[href^=https]:before</code>
<b>E[attr\$=val]</b>	für alle Elemente E, die ein Attribut attr haben, dessen Wert mit val endet. <code>article a[href\$='.pdf']</code>
<b>E[attr*=val]</b>	Gilt für alle Elemente E, die ein Attribut attr haben, dessen Wert irgendwo die Zeichenkette val hat.
<b>E[attr~=val]</b>	Gilt für alle Elemente E, die ein Attribut attr haben, in dem der Ausdruck val vorkommt. Dieser Ausdruck muss von weiteren durch mindestens ein Leerzeichen getrennt sein oder allein stehen.

## 8.8.2 Spezifität und Kaskade

Wie wir schon festgestellt haben, sind die Stylesheets kaskadierend, das heißt, Styles überschreiben sich, ein Stylesheet kann ein anderes überschreiben.

Wenn keine CSS definiert werden, wird nach dem User Agent Stylesheet (Browserstylesheet) formatiert, sobald wir eigene Styles hinzugefügt haben, gelten diese Regeln.

(Genau Erläuterung auch im Buch HTML und CSS ab S. 381)

CSS	Ergebnis
<p>Externes Stylesheet</p> <pre>h2 { color: blue; }</pre> <p>Styles im &lt;head&gt;</p> <pre>h2 {color:red; }</pre> <pre>&lt;h2 style="color: green"&gt;Absatz über Frösche&lt;/h2&gt;</pre> <p>...</p> <pre>&lt;h2&gt;Zweiter Absatz&lt;/h2&gt;</pre>	<p><b>Absatz über Frösche</b></p> <p>Text über Frösche Lorem ipsum dolor sit amet consectetur adipisicing elit. Id exercitationem repudiandae aut commodi rerum fugiat error repellat rem fugit consequatur. Natus quidem numquam id! Quia non recusandae accusantium quo officia.</p> <p><b>Zweiter Absatz</b></p> <p>Das interne Stylesheet überschreibt das externe, der Inline Style überschreibt beide.</p>

### Gewichtung der Selektoren - Spezifität

Style-Attribut	ID	Klasse, Pseudoklasse, Attribute	Elemente, Pseudoelemente
<b>A,</b>	<b>B,</b>	<b>C,</b>	<b>D</b>
Höchste Gewichtung	...	...	Niedrigste Gewichtung

Beispiele hierarchisch geordnet: oben geringste Spezifität, unten höchste

Beispiel	A	B	C	D	Spezifität	Beschreibung
<code>p { color: blue; }</code>	0	0	0	1	(0,0,0,1)	1 Typselektor
<code>div p {color: cyan;}</code>	0	0	0	2	(0,0,0,2)	2 Typselektoren
<code>article div div p { color: pink;}</code>	0	0	0	4	(0,0,0,4)	4 Typselektoren
<code>.farbe {color: green;}</code>	0	0	1	0	(0,0,1,0)	1 Klassenselektor
<code>p.farbe {color: violet;}</code>	0	0	1	1	(0,0,1,1)	1 Klassenselektor 1 Typselektor
<code>#absatz p {color: yellow;}</code>	0	1	0	1	(0,1,0,1)	1 ID-Selektor 1 Typselektor
<code>div#absatz p.farbe::first-letter</code>	0	1	1	3	(0,1,1,3)	1 ID-Selektor 2 Typselektoren 1 Pseudoelement
<code>&lt;p style="color: green"&gt;...</code>	1	0	0	0	(1,0,0,0)	1 Inline-Style