

Webdesign mit HTML und CSS

HTML



CSS



Inhaltsverzeichnis

Inhaltsverzeichnis	2
HTML (Hypertext Markup Language)	3
Webbrowser	3
Aufbau einer HTML-Seite	4
Überschriften und Absätze	6
Markierungen und Hervorhebungen	6
Bereiche	7
Hyperlinks	8
Fotos und Grafiken	10
Zitate und Quellenangaben	12
Whitespace	12
Metainformationen	13
Sonderzeichen	14
Listen	15
Tabellen	16
Trennlinien	17
Audio und Video	18
Untertitel für Audio und Video	20
Links	21
CSS (Cascading Style Sheets)	22
Aufbau von Stylesheets	22
Selektoren	24
Pseudo-Klassen	25
Pseudo-Elemente	26
Spezifität	26
Farben	27
Text	29
Webschriften	30
Hintergründe und Farbverläufe	31
Einheiten	32
Box-Model	32
Rahmen	34
Tabellen	35
Listen	35
Cursor, Content und Counter	36
Flexbox	37
Schatteneffekte	39
Filter-Effekte und Transformationen	39
Transitionen	40
Animationen	41
Nesting	42
Weitere At-Rules	42
Custom Properties	43
Vendor Prefixes	43
Berechnungen	44
Links	45

HTML (Hypertext Markup Language)

Die Hypertext Markup Language (HTML) ist eine textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten. HTML-Dokumente sind die Grundlage des World Wide Web und werden von Webbrowsern dargestellt. Neben den vom Browser angezeigten Inhalten können HTML-Dateien zusätzliche Angaben in Form von Metainformationen enthalten, z. B. über die im Text verwendeten Sprachen, den Autor oder den zusammengefassten Inhalt des Textes.

HTML wird vom World Wide Web Consortium (W3C) und der Web Hypertext Application Technology Working Group (WHATWG) weiterentwickelt. Die letzte vom W3C veröffentlichte Version ist seit dem 14. Dezember 2017 HTML 5.2, seitdem wird HTML in Form eines "Living Standard" von der WHATWG weiterentwickelt. Auch die Extensible Hypertext Markup Language (XHTML) wird durch HTML5 ersetzt.

HTML dient als Auszeichnungssprache dazu, einen Text semantisch zu strukturieren, nicht aber zu formatieren. Die visuelle Darstellung ist nicht Teil der HTML-Spezifikationen und wird durch den Webbrowser und Gestaltungsvorlagen wie Cascading Style Sheets (CSS) bestimmt.

Webbrowser

Webbrowser	Version	Rendering-Engine
⌚ Google Chrome    	≥ 119	WebKit (≤ 27), Blink
⌚ Mozilla Firefox    	≥ 117	Gecko
⌚ Microsoft Edge    ⌚ Microsoft Internet Explorer 	≥ 119	Trident (≤ 11), EdgeHTML (≤ 44), Blink
⌚ Apple Safari 	≥ 17	WebKit

Tabelle: Webbrowser

Aufbau einer HTML-Seite

HTML-Seiten setzen sich zusammen aus dem eigentlichen Inhalt und Markup, sogenannten Tags. Ein Tag besteht aus einer Typ-Bezeichnung und aus keinem, einem oder mehreren Attributen zwischen einem <-Zeichen am Anfang und einem >-Zeichen am Ende. Es gibt einteilige und zweiteilige Tags. Zweiteilige Tags haben einen öffnen Tag der alle Attribute aufnimmt und einen schließenden Tag welcher durch ein / -Zeichen vor der Typ-Bezeichnung gekennzeichnet wird. Ein zweiteiliger Tag kann Text oder weitere Tags enthalten. Ein Dokument mit verschachtelten Tags muss wohlgeformt (well-formed) sein. Das bedeutet, dass der Tag welcher als erstes geöffnet wurde auch als letztes erst geschlossen werden darf. Sowohl innerhalb eines Tags als auch zwischen mehreren Tags dürfen beliebig viele Whitespaces sein.

Attribute gibt es in zwei Varianten. Flag-Attribute bestehen nur aus ihrer Bezeichnung und schalten ein Feature ein bzw. aus. Key-Value-Attributen wird durch das =-Zeichen ein Wert zugewiesen. Der Wert muss von einfachen oder doppletten Hochkommas (' oder ") umschlossen sein und darf keinen Zeilenumbruch enthalten.

Alles zwischen <!-- und --> wird als Kommentar behandelt und vom Webbrowser ignoriert.

HTML-Dateien haben die Endung .html oder .htm.

```
<tag>  
  
<tag>INHALT</tag>  
  
<tag attribut="WERT">INHALT</tag>  
  
<tag attribut1="WERT" attribut2="WERT" attribut3>  
  
  <tag attribut>  
  
</tag>  
  
<!-- KOMMENTAR-->
```

Codebeispiel: HTML-Tags

Das erste Element in jedem HTML-Dokument muss <!DOCTYPE html> sein. Damit wird dem Browser mitgeteilt, dass es sich um ein HTML5-Dokument handelt.

Das Grundgerüst einer HTML-Seite sieht folgendermaßen aus. Seit HTML5 sind die Tags `html`, `head` und `body` optional.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Seitentitel</title>
  </head>
  <body>
    <!-- Inhalt -->
  </body>
</html>

```

Codebeispiel: Grundgerüst einer HTML-Seite

Tag	Beschreibung
<body></body>	Enthält den sichtbaren Seiteninhalt
<head></head>	Kopfteil enthält nicht sichtbare Metadaten
<html></html>	Wurzel einer HTML-Seite
<title></title>	Legt den Titel der Seite im Browserfenster oder für Favoriten fest

Tabelle: Tags für die Grundstruktur

Welche Attribute für einen Tag gültig sind ist vom jeweiligen Tag abhängig. Es gibt allerdings eine Auswahl globaler Attribute die auf jeden Tag anwendbar sind.

Attribut	Beschreibung	Wert
class	Ein oder mehrere Klassennamen	Alphanumerischer Wert ohne Whitespaces oder mehrere Klassen getrennt durch Leerzeichen
hidden	Element ausblenden	☒
id	Eindeutiger Name	Muss ein eindeutiger, alphanumerischer Wert ohne Whitespaces sein
lang	Sprache des Elements	de, en, ...
style	CSS Properties	CSS Properties durch Semikolon getrennt
title	Erzeugt einen Tooltip	Freitext

Tabelle: Globale Attribute

Überschriften und Absätze

Tag	Beschreibung
<code><h1></h1>, <h2></h2>, <h3></h3>, <h4></h4>, <h5></h5>, <h6></h6></code>	Überschriften
<code><p></p></code>	Legt einen Absatz fest

Tabelle: Tags zum Festlegen von Überschriften und Absätzen

Markierungen und Hervorhebungen

Tag	Beschreibung
<code>, </code>	Hervorgehoben (Fett)
<code><code></code></code>	Programmcode
<code></code>	Gelöschter Text (Durchgestrichen)
<code><i></i></code>	Hervorgehoben (Kursiv)
<code><ins></ins></code>	Eingefügter Text
<code><kbd></kbd></code>	Tastatureingabe oder Tastenkürzel
<code><mark></mark></code>	Markierung
<code><s></s></code>	Hervorgehoben (Durchgestrichen)
<code><small></small></code>	Kleingedrucktes
<code></code>	Markierung ohne besondere Semantik
<code><sub></sub></code>	Text tiefgestellt
<code><sup></sup></code>	Text hochgestellt
<code><u></u></code>	Hervorgehoben (Unterstrichen)
<code><var></var></code>	Variable

Tabelle: Tags zum Markieren und Hervorheben von Text

Bereiche

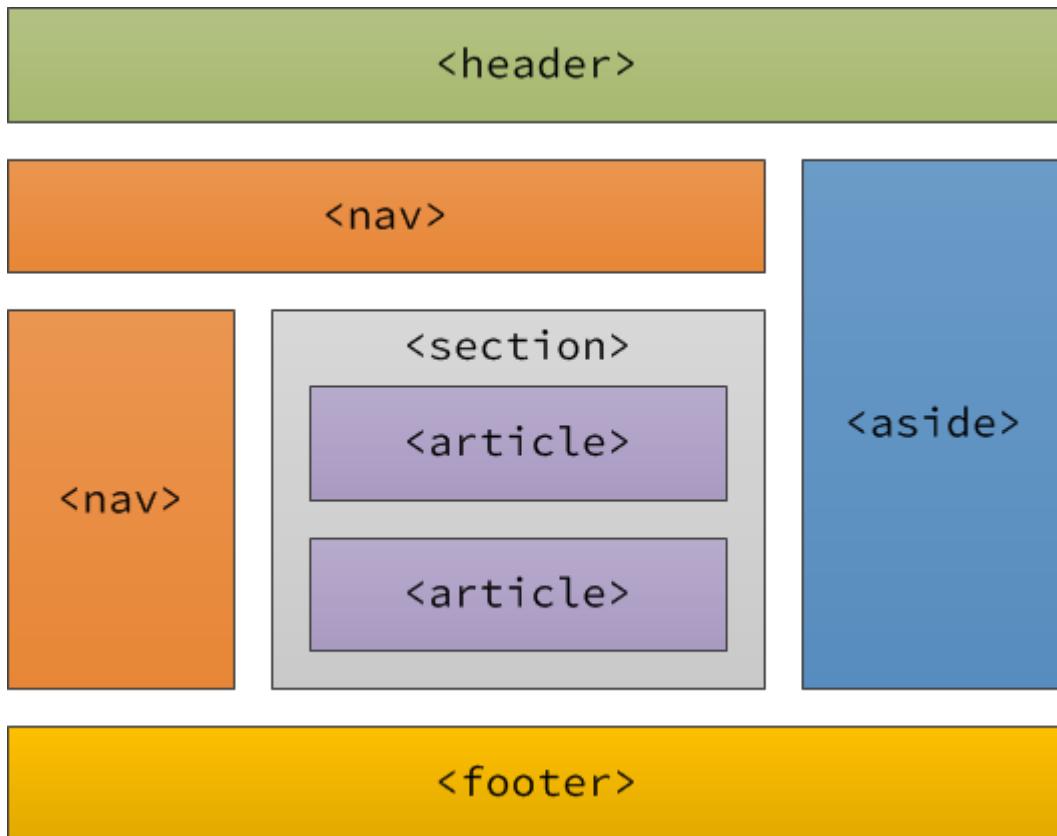


Abbildung: The Semantic Layout

Tag	Beschreibung
<article></article>	Artikel
<aside></aside>	Seitenleiste
<div></div>	Bereich ohne besondere Semantik (division)
<figcaption></figcaption>	Titel für figure
<figure></figure>	Abbildung mit Titel
<footer></footer>	Fußelement
<header></header>	Kopfelement
<main></main>	Hauptbereich
<nav></nav>	Navigationselement
<section></section>	Teilbereich

Tabelle: Tags zur Aufteilung in Bereiche

Hyperlinks

Ein Hyperlink (sinngemäß elektronischer Verweis), kurz Link, ist ein Querverweis in einem Hypertext, der funktional einen Sprung zu einem anderen elektronischen Dokument oder an eine andere Stelle innerhalb eines Dokuments ermöglicht. Wenn der Hyperlink ausgeführt wird, wird automatisch das darin angegebene Ziel aufgerufen. Im Allgemeinen wird der Begriff auf das World Wide Web bezogen, in dem Hyperlinks ein Kernbestandteil sind. Inhaltlich entspricht das Konzept von Hyperlinks Querverweisen in der gedruckten Literatur, deren Ziel dort allerdings in der Regel manuell aufgesucht werden muss.

Quelle: Wikipedia (<https://de.wikipedia.org/wiki/Hyperlink>)

Links können auf Ziele innerhalb einer Seite gesetzt werden. Dabei wird in das `href` Attribut das `#`-Zeichen gefolgt von der Id eines Tags geschrieben. Beim Klick auf den Link sollte der Browser zu dem Element mit der entsprechenden Id springen.

Ein Uniform Resource Locator (URL) identifiziert und lokalisiert eine Ressource, wie z. B. eine Website über die zu verwendende Zugriffsmethode (z. B. das verwendete Netzwerkprotokoll wie HTTP oder FTP) und den Ort (engl. `location`) der Ressource in Computernetzwerken. URLs sind eine Unterart der generellen Identifikationsbezeichnung mittels Uniform Resource Identifier (URIs). Da URLs die erste und häufigste Art von URIs darstellen, werden die Begriffe häufig synonym verwendet. Im allgemeinen Sprachgebrauch werden URLs auch als Internetadresse oder Webadresse bezeichnet.

Quelle: Wikipedia (https://de.wikipedia.org/wiki/Uniform_Resource_Locator)

Eine **URL** setzt sich zusammen aus dem verwendeten **Protocol** gefolgt von der optionalen Angabe eines **Benutzernamens mit Passwort**, dem **Rechnernamen** als voll qualifizierten Domain-Namen (FQDN = Fully Qualified Domain Name) mit optionaler Portnummer, ein **Pfad** auf dem **Rechner** und abschließend dem **Dateinamen**.

```
http://usr:pwd@example.org:80/demo/example.html?param=val#id
```

Codebeispiel: Zusammensetzung einer URL

```
<a href="https://www.google.de/" target="_blank">Google</a>
```

```
<iframe name="f1" width="480" height="320" sandbox></iframe>
```

Codebeispiel: Anchor-Tag

		Attribut		
Tag	Beschreibung	Bezeichnung	Beschreibung	Werte
<a>	Hyperlink zur einer anderen Seite oder Seitenteil	href	Dateipfad oder URL	../pfad/, http://domain.tld/, mailto:name@domain.tld, #ID
		target	Bereich/Fenster in dem das Ziel angezeigt werden soll	NAME, _blank, _self, _parent, _top
		download	Ziel zum Download anbieten	DATE/NAME
<iframe></iframe>	Eine andere Seite als Unterelement einbinden	height	Höhe in Pixeln	Bsp: 300
		name	Name zur Referenzierung durch target Attribut	NAME
		sandbox	Schränkt Berechtigungen ein	allow-forms, allow-popups, allow-same-origin, allow-scripts, allow-top-navigation
		src	Dateipfad oder URL	../pfad/, http://domain.tld/
		width	Breite in Pixeln	Bsp: 400

Tabelle: Links und Frames

Fotos und Grafiken

```


<picture>
  <source srcset="picture.webp" type="image/webp">
  
</picture>
```

Codebeispiel: Einbindung von Fotos und Grafiken

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
	Bindet externe Grafikdatei ein	alt	Alternativbeschreibung des Bildes (erforderlich)	Freitext
		src	Dateipfad oder URL	URI
		srcset	Dateipfad oder URL	URI
		width, height	Breite und/oder Höhe mit der die Grafik angezeigt werden soll in Pixeln	Bsp.: 320
<picture>	Bindet Grafik mit mehreren Quellen ein	⊗		
<source>	Eine von mehreren Grafikdateien	media	Quelle auf ein bestimmtes Medium einschränken	MEDIAQUERY, all, screen, ...
		srcset	Dateipfad oder URL	URI
		type	Grafikformat	image/jpeg, ...
		width, height	Breite und/oder Höhe mit der die Grafik angezeigt werden soll in Pixeln	Bsp.: 320

Tabelle: Tags für Fotos und Grafiken

Vektorgrafiken bestehen aus mathematischen Formeln für Formen und Linien und sind dadurch beliebig skalierbar, ohne an Qualität zu verlieren. Sie eignen sich für Logos und einfache Designs. **Rastergrafiken** hingegen bestehen aus Pixeln, einer festen Anzahl von Bildpunkten, und eignen sich für komplexe Bilder wie Fotos, verlieren aber an Qualität, wenn sie vergrößert werden.

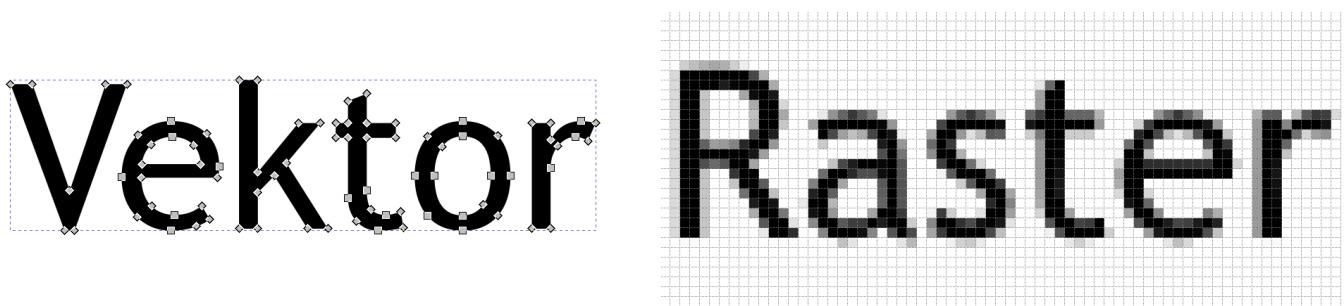


Abbildung: Vektor- vs. Raster-Grafikformat

Alle unten aufgeführten Grafikformate, ausgenommen das JPEG File Interchange Format, können auch zur Darstellung kurzer Animationen verwendet werden. In dem Fall sind mehrere Einzelbilder in der Grafikdatei enthalten, welche nacheinander in einer Endlosschleife angezeigt werden. Scalable Vector Graphics können ebenfalls animiert sein, enthalten dann aber keine Einzelbilder sondern Animationslogik.

Bezeichnung	Dateiendung	Media Type	Farbtiefe	Komprimierung
AV1 Image File Format (AVIF)	.avif	image/avif	48-Bit-RGBA	Verlustfrei, Verlustbehaftet
Graphics Interchange Format (GIF)	.gif	image/gif	8-Bit (256 Farben)	Verlustfrei
JPEG File Interchange Format (JFIF)	.jpg, .jpeg, .jpe	image/jpeg	24-Bit-RGB	Verlustbehaftet
Portable Network Graphics (PNG)	.png	image/png	32-Bit-RGBA	Verlustfrei
Scalable Vector Graphics (SVG)	.svg	image/svg+xml	32-Bit-RGBA	Keine (Vektorgrafik)
WebP File Format	.webp	image/webp	32-Bit-RGBA	Verlustfrei, Verlustbehaftet

Tabelle: Grafikformate

Zitate und Quellenangaben

```
<blockquote cite="http://developer.mozilla.org">  
    Das ist ein Zitat vom Mozilla Developer Center.  
</blockquote>  
<cite>Mozilla Developer Center</cite>
```

Codebeispiel: Zitat als Block-Element mit Quellenangabe

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
<blockquote> </blockquote>	Erweiterndes Zitat	cite	Quelle als URL	URI
<cite></cite>	Referenz zu einer kreativen Arbeit	☒		
<q></q>	Kurzes Zitat innerhalb eines Textes	cite	Quelle als URL	URI

Tabelle: Tags für Zitate und Quellenangaben

Whitespace

Whitespace (auch **Leerraum** oder **Zwischenraumzeichen**) werden solche Zeichen genannt, welche in einem Text normalerweise nur durch Leerflächen dargestellt werden und dennoch (Speicher-)Platz in Anspruch nehmen. In HTML werde alle diese Zeichen, wie beispielsweise Zeilenumbrüche, auch mehrere aufeinanderfolgende Zeichen, immer zu einem einzelnen Leerzeichen zusammengefasst.

Tag	Beschreibung
 	Erzeugt einen Zeilenumbruch (line break)
<pre> </pre>	Whitespace im Quelltext bleibt erhalten (preformatted text)
<wbr>	Legt einen Punkt in einem Wort fest an dem ein Zeilenumbruch eingefügt werden kann (word break opportunity)

Tabelle: Tags zur Beeinflussung des Whitespace-Verhaltens

Metainformationen

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="author" content="Autorennname">
    <meta name="description" content="Seitenbeschreibung">
    <!-- Weiterleitung nach 3 Sec. -->
    <meta http-equiv="refresh" content="3; URL=/seite2.html">
    <title>Seitentitel</title>
  </head>
  <body>
    <h1>Überschrift<h1>
  </body>
</html>
```

Codebeispiel: HTML-Seite mit Metainformationen

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
<meta>	Metadaten zur Seite hinterlegen	charset	Bestimmt den Zeichensatz der Seite	UTF-8, ISO-8859-1, ASCII, ...
		content	Wert der Metainformation	WERT
		http-equiv	HTTP-Header	refresh, content-type, cache-control, ...
		name	Bestimmt welche Metainformation	author, description, keywords, viewport, ...

Tabelle: Metatags

Sonderzeichen

Zur Kodierung von Sonderzeichen gibt es sogenannte **HTML-Entities**. Diese bestehen aus einem &-Zeichen, einer Bezeichnung oder #-Zeichen gefolgt von einem Zeichencode und einem abschließendem ; -Zeichen.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="ISO-8859-1">
    <title>Spa&szlig; mit Zeichens&uuml;tzen</title>
  </head>
  <body>
    <h1>&Uuml;berschrift<h1>
  </body>
</html>
```

Codebeispiel: Seite mit Sonderzeichen als HTML-Entities

Entity	Zeichen	Beschreibung
&	&	Und-Zeichen
ä, Ä, ö, Ö, ü, Ü, ß	ä, Ä, ö, Ö, ü, Ü, ß	Umlaute
„, “, ‚, ‘, ”, ’	„, „, ‘, ’	Anführungszeichen
€, ¢, £, ¥	€, ¢, £, ¥	Währungszeichen
©, ®, ™	©, ®, ™	Copyright- und Trademark-Zeichen
<, >	<, >	Größer-/Kleinerzeichen
 		Nichtumbrechendes Leerzeichen
­	- (wenn ein Zeilenumbruch statt findet)	Bedingter Trennstrich

Tabelle: HTML-Entities

Listen

```
<ul>
  <li>One
    <ol>
      <li>First</li>
      <li>Second</li>
    </ol>
  </li>
  <li>Two
    <ol>
      <li>First</li>
      <li>Second</li>
    </ol>
  </li>
</ul>

<dl>
  <dt>Coffee</dt>
  <dd>black hot drink</dd>
  <dt>Milk</dt>
  <dd>white cold drink</dd>
</dl>
```

Codebeispiel: Grundgerüst einer HTML-Seite

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
	Ungeordnete Liste	∅		
	Nummerierte Liste	start	Startwert	0, 1, 2, 3, 4, ...
		type	Aufzählungsart	1, A, a, I, i
	Einzelner Listeneintrag	∅		
<dl></dl>	Beschreibungsliste	∅		
<dt></dt>	Begriff Listeneintrag	∅		
<dd></dd>	Beschreibender Listeneintrag	∅		

Tabelle: Aufzählungslisten

Tabellen

Tabellen werden in HTML zeilenweise erstellt. Eine Tabellenzeile (`tr`) enthält Tabellenzellen (`td` oder `th`). Die Zellen an derselben Position in den Zeilen bilden eine Spalte.

```
<table>
  <tr>
    <th>A1</th><th>B1</th><th>C1</th>
  </tr>
  <tr>
    <td>A2</td><td>B2</td><td>C2</td>
  </tr>
</table>
```

Codebeispiel: Aufbau einer einfachen Tabelle

Tabellen können einen Titel (`caption`) haben und mit `thead`, `tbody` oder `tfoot` in Bereiche aufgeteilt werden. Mit `rowspan` und `colspan` können außerdem Zeilen- bzw. Spaltenübergreifende Zellen erstellt werden.

```
<table>
  <caption>Eine Tabelle</caption>
  <thead>
    <tr>
      <th>A1</th><th>B1</th><th>C1</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">A2</td><td>B2</td><td>C2</td>
    </tr>
    <tr>
      <td colspan="2">B3</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>A4</td><td>B4</td><td>C4</td>
    </tr>
  </tfoot>
</table>
```

Codebeispiel: Tabelle mit Bereichen

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
<table></table>	Definiert eine Tabelle	☒		
<tr></tr>	Tabellenzeile	☒		
<th></th>	Tabellenkopfzelle	colspan	Spaltenbreite	2, 3, 4, 5, ...
		rowspan	Zeilenbreite	2, 3, 4, 5, ...
<td></td>	Normale Tabellenzelle	colspan	Spaltenbreite	2, 3, 4, 5, ...
		rowspan	Zeilenbreite	2, 3, 4, 5, ...
<thead></thead>	Definiert Kopfbereich	☒		
<tbody></tbody>	Definiert Hauptbereich	☒		
<tfoot></tfoot>	Definiert Fußbereich	☒		
<caption></caption>	Tabellentitel	☒		

Tabelle: Tabellentags

Trennlinien

```
<hr>
```

Codebeispiel: Trennlinie

Tag	Beschreibung
<hr>	Horizontaler Trenner (horizontal rule)

Tabelle: Tag zum setzen einer Trennlinie

Audio und Video

```
<video controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
</video>

<audio src="music.mp3" autoplay loop></audio>
```

Codebeispiel: Einbinden von Audio- und Video-Inhalten

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
<audio> </audio>, <video> </video>	Bindet Audio/Video-Datei ein	autoplay	Automatisch starten	
		controls	Player Controls anzeigen	
		loop	Endlosschleife	
		muted	Ton stumm geschaltet	
		poster	Vorschaubild für Videos	URI
		src	Dateipfad oder URL	URI
		width, height	Breite und/oder Höhe des Videos in Pixeln	Bsp.: 320
<source>	Eine von mehreren Audio-/Video- Quellen	media	Quelle auf ein bestimmtes Medium einschränken	MEDIAQUERY, all, screen, ...
		src	Dateipfad oder URL	URI
		type	Typ der Video- /Audio-Datei	video/webm, audio/ogg, ...

Tabelle: Audio- und Video-Tag

Ein Media-Container ist ein Dateiformat, das einen oder mehrere Medienstreams (wie Audio oder Video) zusammen mit Metadaten kapselt, sodass sie gemeinsam gespeichert und wiedergegeben werden können. Es bestimmt nur, wie diese Daten organisiert und zusammengefügt werden, aber nicht, wie das Video oder der Ton komprimiert (kodiert) sind. Alle modernen Webbrowser unterstützen die Media-Container WebM und MPEG-4. Damit Video und Ton abgespielt werden, müssen aber auch die Verwendeten Codecs vom Webbrowser unterstützt werden.

Bezeichnung	Dateiendung	Media Type	Komprimierung
MPEG-4 Advanced Audio Coding (AAC) Audio Format	.m4a	audio/mp4	Verlustbehaftet
Free Lossless Audio Codec (FLAC)	.flac	audio/flac	Verlustfrei
MP3 Audio Format	.mp3	audio/mpeg	Verlustbehaftet
Ogg Vorbis Audio Format	.ogg	audio/ogg	Verlustbehaftet
Ogg Opus Audio Format	.opus	audio/ogg	Verlustbehaftet
Waveform Audio File Format (WAVE)	.wav	audio/wav	Keine

Tabelle: Audioformate

Bezeichnung	Dateiendung	Media Type	Video-Codec	Audio-Codec
MPEG-4 Video Format	.mp4	video/mp4	Advanced Video Coding (AVC), AOMedia Video 1 (AV1)	Advanced Audio Coding (AAC)
WebM Video Format	.webm	video/webm	VP8, VP9, AOMedia Video 1 (AV1)	Vorbis, Opus

Tabelle: Videoformate

Untertitel für Audio und Video

```
<video controls src="video.webm">
  <track kind="subtitles"
    srclang="en"
    src="sub-en.vtt"
    label="eng" default>
  <track kind="subtitles"
    srclang="de"
    src="sub-de.vtt"
    label="deu">
</video>
```

Codebeispiel: Track-Tags

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
<track>	Fügt Text-Track zu video hinzu	default	Aktiviert von Anfang an	
		kind	Art des Text-Tracks (Standard: subtitles)	subtitles, captions, descriptions, chapters
		label	Titel des Text-Tracks	Freitext
		src	Dateipfad oder URL	URI
		srclang	Sprache des Text-Tracks (Erforderlich bei Untertiteln)	de, en, ...

Tabelle: Track-Tag

Bezeichnung	Dateiendung	Media Type
Web Video Text Tracks Format (WebVTT)	.vtt	text/vtt

Tabelle: WebVTT-Format

Links

- ↗ World Wide Web - Wikipedia (https://de.wikipedia.org/wiki/World_Wide_Web)
- ↗ Hypertext Markup Language - Wikipedia (https://de.wikipedia.org/wiki/Hypertext_Markup_Language)
- ↗ W3C HTML 5.2 Specification (<https://www.w3.org/TR/2017/REC-html52-20171214/>)
- ↗ WHATWG HTML Living Standard (<https://html.spec.whatwg.org/>)
- ↗ Joint Photographic Experts Group (JPEG) (<https://jpeg.org/>)
- ↗ Portable Network Graphics (PNG) (<http://www.libpng.org/pub/png/>)
- ↗ WebP - An image format for the Web (<https://developers.google.com/speed/webp>)
- ↗ W3C Scalable Vector Graphics (SVG) 1.1 Specification (<https://www.w3.org/TR/SVG11/>)
- ↗ W3C WebVTT: The Web Video Text Tracks Format (<https://www.w3.org/TR/webvtt1/>)
- ↗ The WebM Project (<https://www.webmproject.org/>)
- ↗ AV1 Video Codec (<https://aomedia.org/av1/>)
- ↗ AV1 Image File Format (AVIF) (<https://aomediacodec.github.io/av1-avif/>)
- ↗ Farbtiefe (Computergrafik) - Wikipedia ([https://de.wikipedia.org/wiki/Farbtiefe_\(Computergrafik\)](https://de.wikipedia.org/wiki/Farbtiefe_(Computergrafik)))
- ↗ Google Chrome (https://www.google.com/intl/de_ALL/chrome/)
- ↗ Mozilla Firefox (<https://www.mozilla.org/de/firefox/>)
- ↗ Microsoft Edge (<https://www.microsoft.com/de-de/windows/microsoft-edge>)
- ↗ Apple Safari (<https://www.apple.com/de/safari/>)
- ↗ Visual Studio Code (<https://code.visualstudio.com/>)
- ↗ HTML: HyperText Markup Language - MDN (<https://developer.mozilla.org/docs/Web/HTML>)

CSS (Cascading Style Sheets)

Cascading Style Sheets (CSS) ist eine Stylesheet-Sprache für elektronische Dokumente und zusammen mit HTML eine der Kernsprachen des World Wide Webs. Sie wird vom World Wide Web Consortium (W3C) beständig weiterentwickelt. Mit CSS werden Gestaltungsanweisungen erstellt, die vor allem zusammen mit der Auszeichnungssprachen HTML eingesetzt werden.

Quelle: Wikipedia (https://de.wikipedia.org/wiki/Cascading_Style_Sheets)

Aufbau von Stylesheets

In CSS sind **Properties** innerhalb von Regeln organisiert. Eine **Regel** besteht aus einem **Selektor**, gefolgt von einem Bereich welcher durch geschweifte Klammern begrenzt wird ("{" und "}"). Innerhalb der geschweiften Klammern können beliebig viele (mindestens eine) **Property-Wert-Zuweisungen** vorgenommen werden. Jede Zuweisung muss durch einen Strichpunkt ";" beendet werden. Dasselbe Property darf innerhalb einer Regel nur einmal verwendet werden.

In CSS können mehrzeilige C-Style Kommentare verwendet werden. Alles was zwischen den Zeichenkombinationen "/*" und "*/" steht wird vom Browser ignoriert.

CSS-Dateien haben die Endung .css.

```
/* Kommentar */

selektor {

    property1: WERT;

    property2: WERT;

    /* ... */

}
```

Codebeispiel: Stylesheets

Es gibt drei Möglichkeiten Stylesheets in eine HTML-Seite einzubinden. Über das Style-Attribut können Properties für jeden Tag einzeln gesetzt werden (**inline**). Der **Style-Tag** erlaubt es Properties über Selektoren für die gesamte Seite zu definieren. Außerdem kann durch den **Link-Tag** eine externe CSS-Datei eingebunden werden.

```

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="utf-8">
    <title>Seite</title>
    <link rel="stylesheet" href="stylesheet.css">
    <style>
        h1 {
            font-size: 24px;
        }
    </style>
</head>
<body>
    <h1>Kapitel</h1>
    <p style="font-size: 16px;">Text</p>
</body>
</html>

```

Codebeispiel: Einbindung von Stylesheets in HTML

Tag	Beschreibung	Attribut		
		Bezeichnung	Beschreibung	Werte
<link>	Stellt eine Beziehung (Link) zu einer externen Datei her	href	Absolute oder Relative URL	URL
		integrity	Checksumme für externe Ressourcen	HASH
		media	Stylesheets auf ein bestimmtes Medium einschränken	all, screen, print, MEDIAQUERIES
		rel	Art der Beziehung	stylesheet
		type	Typ	text/css
<style> </style>	Definiert Stylesheets für die gesamte Seite	media	Stylesheets auf ein bestimmtes Medium einschränken	all, screen, print, MEDIAQUERIES
		type	Typ	text/css

Tabelle: Tags zur Einbindung von Stylesheets

Selektoren

Mit dem **Selektor** wird bestimmt bei welchen Elementen bestimmte Properties gesetzt werden sollen. Mehrere Selektoren können Gruppiert (,) bzw. miteinander verknüpft werden.

Selektor	Beschreibung
*	Universal-Selektor, alle Tags
<i>ELEMENT</i>	Typ-Selektor, alle Tags des gleichen Typs
.CLASS	Klassen-Selektor, alle Tags einer Klasse
#ID	ID-Selektor, Tag mit einer spezifischen eindeutigen Id
<i>SELEKTOR1, SELEKTOR2</i>	Mehrere unterschiedliche Selektoren
<i>ELEMENT1 ELEMENT2</i>	Alle ELEMENT2-Tags die sich (direkt oder indirekt) in ELEMENT1-Tags befinden
<i>ELEMENT1 > ELEMENT2</i>	Alle ELEMENT2-Tags die sich direkt in ELEMENT1-Tags befinden
<i>ELEMENT1 ~ ELEMENT2</i>	Alle ELEMENT2-Tags die (direkt oder indirekt) einem ELEMENT1-Tag nachfolgen
<i>ELEMENT1 + ELEMENT2</i>	Alle ELEMENT2-Tags die direkt einem ELEMENT1-Tag nachfolgen
& <i>SELEKTOR1</i>	Parent-Selektor in nested Rules

Tabelle: Liste von einfachen Selektoren und Möglichkeiten der Gruppierung bzw. Verknüpfung

Selektor	Beschreibung
[attr]	Attribut-Selektor
[attr=value]	Attribut-Selektor mit Wert
[attr~=value]	Attribut-Selektor mit einer durch Leerzeichen getrennten Liste von Werten
[attr^=value]	Attributwert beginnt mit value
[attr\$=value]	Attributwert endet mit value
[attr*=value]	Attributwert enthält value

Tabelle: Liste von Attribut-Selektoren

Pseudo-Klassen

Selektor	Beschreibung
:active	Element wurde angeklickt und Aktion wird ausgeführt
:empty	Leere Elemente (auch kein Text/Whitespace)
:first-child	Erstes Kindelement
:first-of-type	Erstes Kindelement eines Typs
:hover	Maus über Element
:lang()	Sprache des Elements
:last-child	Letztes Kindelement
:last-of-type	Letztes Kindelement eines Typs
:link, :visited	Nicht-Besuchte und Besuchte Links
:not()	Negation eines Selektors
:nth-child(), :nth-last-child()	N-tes Kindelement (even = gerade, odd = ungerade)
:nth-of-type(), :nth-last-of-type()	N-tes Kindelement eines Typs (even = gerade, odd = ungerade)
:only-child	Einziges Kindelement
:only-of-type	Einziges Kindelement eines Typs
:root	Root-Element
:target	Element-ID in URL-Fragment (#id)

Tabelle: Liste von Pseudo-Klassen

Pseudo-Elemente

Selektor	Beschreibung
::after, ::before	Bereich vor oder nach einem Element
::cue	Video-Untertitel
::first-letter, ::first-line	Erste(s) Zeichen/Zeile eines Textes
::selection	Ausgewählter Text

Tabelle: Liste von Pseudo-Elementen

Spezifität

Wenn ein CSS-Selektor mehrfach im Dokument steht gibt es einen Konflikt. In diesem Fall interpretiert der Browser immer die zuletzt notierte Regel. Eine im Stylesheet weiter unten notierte Regel überschreibt also alle zuvor notierten identischen Selektoren.

Eine CSS-Regel wird nur dann von einer weiter unten notierten Regel überschrieben, wenn beide Selektoren identisch sind und das gleiche "Gewicht" haben. Jede Art von Selektor hat ein über ein Punkte-System festgelegtes Gewicht das nicht verändert werden kann. Das Gewicht des gesamten CSS-Selektors wird über die addierten Punkte der einzelnen Bestandteile ermittelt. Der Browser wendet im Konfliktfall immer die Regel mit dem höchsten Gewicht, also der höchsten Punkte-Anzahl an, auch wenn diese Regel weiter oben im Quellcode steht. Dieses Verhalten nennt man **Spezifität**.

Selektor-Art	Beispiel	Punkte
Universal-Selektor	*	0
einfacher Typ-Selektor	p	1
Klasse	.infobox	10
Attribut-Selektor	[for]	10
Pseudo-Klasse	:hover	10
Pseudo-Element	::first-letter	10
ID	#abc123	100
inline Style-Attribut	style="color: red;"	1000

Tabelle: Punkteverteilung nach CSS-Spezifikation

Farben

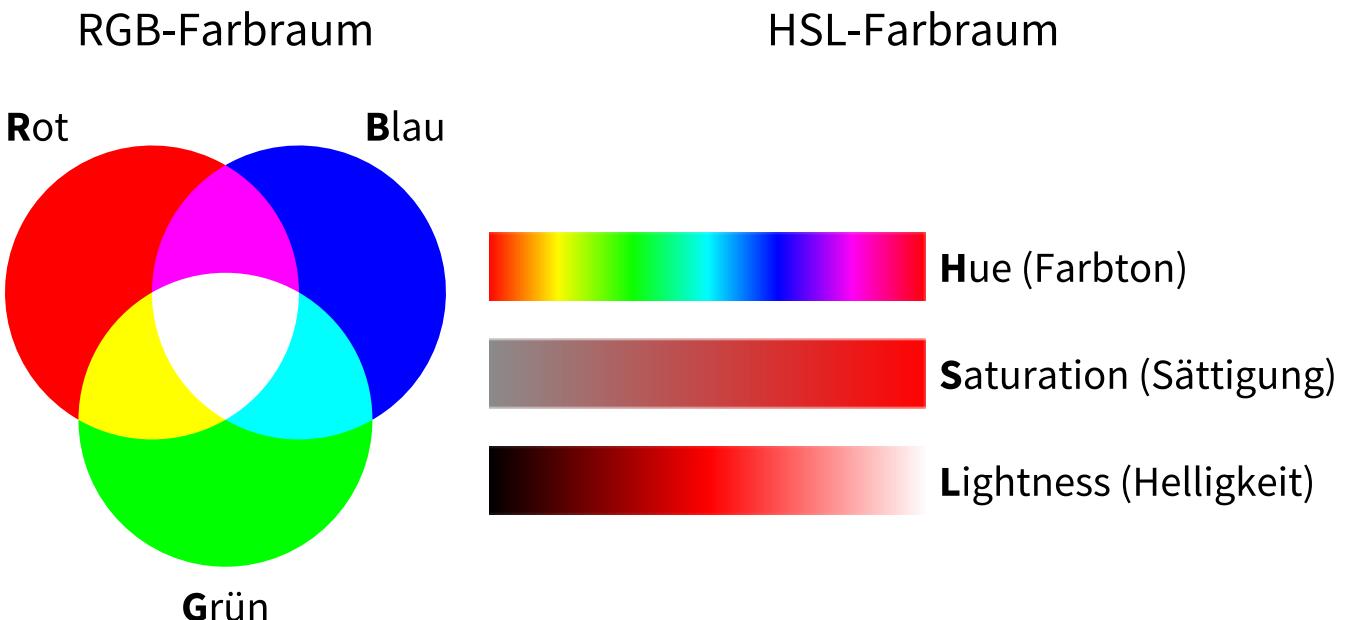


Abbildung: RGB- und HSL-Farbmodell

Im Web werden Farben entweder durch einen von 140 Farbenbezeichnungen oder durch einen 24-Bit Farbwert dargestellt. Der Farbwert beginnt mit einem "#" -Zeichen gefolgt von jeweils einer zweistelligen Hexadezimalzahl für die drei Bytes welche den Wert für Rot, Grün und Blau darstellen. Daraus ergeben sich über 16 Millionen Farben. Des Weiteren gibt es eine CSS Funktion `rgb()` und eine CSS Funktion `rgba()` für 32-Bit Farbwerte (Rot, Grün, Blau + Alpha Transparenz). Alternativ können Farben durch die Funktionen `hsl()` und `hsla()` mit Farbton (0-360), Sättigung und Helligkeit angegeben werden.

```
.colors {
  background-color: black;
  color: #FF0000;
}

.rgb {
  background-color: rgb(255, 0, 0);
  color: rgba(255, 0, 0, 0.5);
}

.hsl {
  background-color: hsl(180, 100%, 50%);
  color: hsla(180, 100%, 50%, 0.5);
}
```

Codebeispiel: Zusammensetzung von Farbwerten

	aliceblue		antiquewhite		aqua			aquamarine
	azure		beige		bisque			black
	blanchedalmond		blue		blueviolet			brown
	burlywood		cadetblue		chartreuse			chocolate
	coral		cornflowerblue		cornsilk			crimson
	cyan		darkblue		darkcyan			darkgoldenrod
	darkgray		darkgreen		darkgrey			darkkhaki
	darkmagenta		darkolivegreen		darkorange			darkorchid
	darkred		darksalmon		darkseagreen			darkslateblue
	darkslategray		darkslategrey		darkturquoise			darkviolet
	deeppink		deepskyblue		dimgray			dimgrey
	dodgerblue		firebrick		floralwhite			forestgreen
	fuchsia		gainsboro		ghostwhite			goldenrod
	gold		gray		green			greenyellow
	grey		honeydew		hotpink			indianred
	indigo		ivory		khaki			lavenderblush
	lavender		lawngreen		lemonchiffon			lightblue
	lightcoral		lightcyan		lightgoldenrodyellow			lightgray
	lightgreen		lightgrey		lightpink			lightsalmon
	lightseagreen		lightskyblue		lightslategrey			lightslategrey
	lightsteelblue		lightyellow		lime			limegreen
	linen		magenta		maroon			mediumaquamarine
	mediumblue		mediumorchid		mediumpurple			mediumseagreen
	mediumslateblue		mediumspringgreen		mediumturquoise			mediumvioletred
	midnightblue		mintcream		mistyrose			moccasin
	navajowhite		navy		oldlace			olive
	olivedrab		orange		orangered			orchid
	palegoldenrod		palegreen		paleturquoise			palevioletred
	papayawhip		peachpuff		peru			pink
	plum		powderblue		purple			rebeccapurple
	red		rosybrown		royalblue			saddlebrown
	salmon		sandybrown		seagreen			seashell
	sienna		silver		skyblue			slateblue
	slategray		slategrey		snow			springgreen
	steelblue		tan		teal			thistle
	tomato		turquoise		violet			wheat
	white		whitesmoke		yellow			yellowgreen
	transparent		currentColor					

Tabelle: Farbnamen

Text

Property	Beschreibung	Wert
color	Text-/Fordergrund-Farbe	<i>FARBWERT, FARBNAME, rgb(), rgba(), hsl(), hsla()</i>
font	Shorthand-Property für Schriften (Feste Reihenfolge: <i>[STYLE] [WEIGHT] SIZE FAMILY</i>)	Bsp.: <code>font: italic bold 10pt sans-serif;</code>
font-family	Schriftart	Bsp.: <code>font-family: Arial, sans-serif;</code>
font-size	Schriftgröße	<i>MASSEINHEIT</i>
font-style	Schriftstil	normal, italic, oblique
font-weight	Schriftdicke (normal = 400, bold = 700)	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900
line-height	Zeilenhöhe	<i>MASSEINHEIT</i>
letter-spacing	Zeichenabstand	<i>MASSEINHEIT</i>
text-align	Textausrichtung	left, center, right, justify
text-decoration	Text unterstreichen, durchstreichen etc.	none, underline, line-through
text-overflow	Text welcher über das Element hinausgeht abschneiden (clip) oder mit "..." abkürzen (ellipsis)	clip, ellipsis
text-transform	Groß- und Kleinschreibung beeinflussen	none, capitalize, uppercase, lowercase
vertical-align	Vertikale Ausrichtung	baseline, sub, super, text-top, text-bottom, middle, top, bottom
white-space	Darstellung von Whitespaces	normal, nowrap, pre, pre-wrap

Tabelle: Schriftproperties

Webschriften

Es gibt drei Standard-Schriftbezeichnungen: **serif**, **sans-serif**, **monospace**. Diese stehen jeweils für die Browserinstellung des Systems für eine Schrift mit Serifen (Bsp.: Times New Roman), ohne Serifen (Bsp.: Arial) und mit fester Zeichenbreite (Bsp.: Courier New). Ihre Verwendung ist demnach unabhängig von den tatsächlich auf dem System installierten Schriften.

Die angegebene Schrift muss auf dem System des Browsers vorhanden sein. Es kann eine Liste von Schriftarten angegeben werden und der Browser verwendet, von links nach rechts, die erste auf dem System vorhandene Schriftart.

Mit der At-Regel `@font-face` lassen sich Schriftarten an den Browser ausliefern. Jede Variante (Bold, Italic, Regular, Bold+Italic) derselben Schriftart muss mit einer eigenen `@font-face`-Regel unter demselben Namen registriert werden. Es muss mindestens eine Datei mit dem entsprechenden Format angegeben werden. Mit `local()` kann auf eine bereits installierte Schriftart verwiesen werden.

```
@font-face {
  font-family: 'Source Sans Pro';
  font-weight: 400;
  font-style: italic;
  src: local('Source Sans Pro'),
       url('SourceSansPro-It.woff') format('woff');
}
```

Codebeispiel: Einbindung von Webschriften

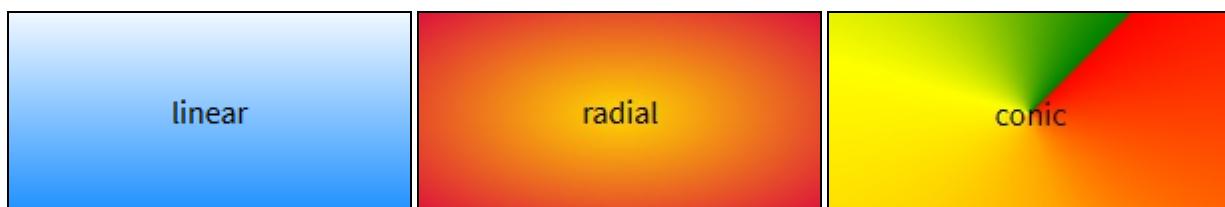
Format	Dateiendung	Beschreibung	Browser
truetype (TrueType Font)	.ttf	Standard von Microsoft und Apple, nicht speziell für Webschriften	
opentype (OpenType Font)	.otf	Erweiterung von TrueType von Microsoft und Adobe, nicht speziell für Webschriften	
woff, woff2 (Web Open Font Format)	.woff, .woff2	Komprimierendes Containerformat für Webschriften (W3C Standard)	

Tabelle: Liste von Dateiformaten für Webschriften

Hintergründe und Farbverläufe

Property	Beschreibung	Wert
background	Shorthand-Property für Hintergründe	Bsp.: background: #000000 url('bg.jpg') repeat;
background-attachment	Hintergrundbild mitscrollen (scroll) oder nicht (fixed)	scroll, fixed
background-color	Hintergrundfarbe	FARBWERT, FARBNNAME, rgb(), rgba(), hsl(), hsla()
background-image	Hintergrundbild	url(), linear-gradient(), radial-gradient(), conic-gradient()
background-position	Position des Hintergrundbildes	MASSEINHEIT, top, bottom, left, right, center
background-repeat	Hintergrundbild kacheln	no-repeat, repeat, repeat-x, repeat-y
background-size	Hintergrundbildgröße	MASSEINHEIT, auto, contain, cover

Tabelle: Backgroundproperties



```
/* linear-gradient(<WINKEL>, <FARBWERT> [BEGINN], ...) */
background-image: linear-gradient(to bottom, white, dodgerblue);

/* radial-gradient(<FARBWERT> [BEGINN], ...) */
background-image: radial-gradient(gold, red);

/* conic-gradient([WINKEL], <FARBWERT> [WINKEL], ...) */
background-image: conic-gradient(red, orange, yellow, green);
```

Codebeispiel: Farbverläufe

Einheiten

Einheit	Bezeichnung	Vergleich
px	Pixel	ein Bildpunkt auf dem Display
in	Zoll	$1\text{in} = 96\text{px}$
cm, mm	Zentimeter, Millimeter	$2.54\text{cm} = 25.4\text{mm} = 1\text{in} = 96\text{px}$
pt	Punkte	$3\text{pt} = 4\text{px}$
em, rem	Schriftgröße	$2\text{em} = \text{Element-Schriftgröße} \times 2$ $3.5\text{rem} = \text{Root-Element-Schriftgröße} \times 3,5$
%	Prozent	☒
deg	Winkel in Grad	Voller Kreis = 360deg
rad	Winkel in Bogenmaß	$1\text{rad} = 180/\pi$
s, ms	Sekunden, Millisekunden	$1\text{s} = 1000\text{ms}$
vh, vw	Viewport-Höhe und Viewport-Breite	$1/100$

Tabelle: Liste von Maßeinheiten in CSS

Box-Model

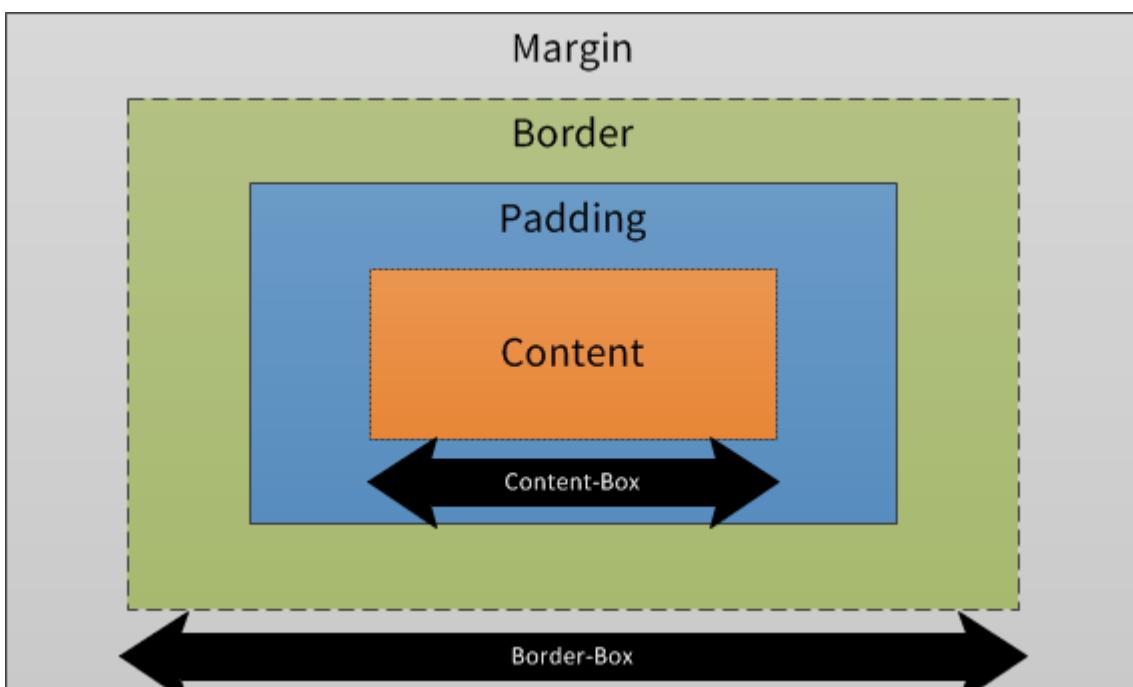


Abbildung: Box-Model

Property	Beschreibung	Wert
clear	Umfließen beenden	none, left, right, both
box-sizing	Legt die Bemessung der Breite und Höhe fest	content-box, border-box
display	Box-Typ festlegen	none, inline, block, inline-block, flex, inline-flex
float	Element umfließen lassen	none, left, right
overflow, overflow-x, overflow-y	Bestimmt das Verhalten wenn der Inhalt nicht in das Element passt	visible, hidden, scroll, auto
position	Art der Positionierung	absolute, fixed, relative, static, sticky
width, height	Breite und Höhe	MASSEINHEIT
max-width, max-height, min-width, min-height	Maximale oder minimale Breite und Höhe	MASSEINHEIT
bottom, left, right, top	Absolute oder relative Position abhängig von position Property	MASSEINHEIT

Tabelle: Box-Model-Properties

Property	Beschreibung	Wert
margin	Shorthand-Property für Außenabstand	MASSEINHEIT
margin-bottom, margin-left, margin-right, margin-top	Außenabstand	MASSEINHEIT
padding	Shorthand-Property für Innenabstand	MASSEINHEIT
padding-bottom, padding-left, padding-right, padding-top	Innenabstand	MASSEINHEIT

Tabelle: Abstand-Properties

Rahmen

```
table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}
```

Codebeispiel: Rahmen für Tabellen

Property	Beschreibung	Wert
border	Shorthand-Property für Rahmen	Bsp.: border: 1px solid black;
border-bottom, border-left, border-right, border-top	Rahmen	Bsp.: border-left: 1px dotted black;
border-color, border-bottom-color, border-left-color, border-right-color, border-top-color	Rahmenfarbe	FARBWERT, FARBNAME, rgb(), rgba(), hsl(), hsla()
border-radius, border-bottom-left-radius, border-bottom-right-radius, border-top-left-radius, border-top-right-radius	Abgerundete Rahmen	MASSEINHEIT
border-style, border-bottom-style, border-left-style, border-right-style, border-top-style	Rahmenstil	none, dotted, dashed, solid, double, groove, ridge, inset, outset
border-width, border-bottom-width, border-left-width, border-right-width, border-top-width	Rahmenbreite	MASSEINHEIT, thin, medium, thick

Tabelle: Rahmen-Properties

Tabellen

Property	Beschreibung	Wert
border-collapse	Tabellenzellen mit eigenem Rahmen (separate) oder nur ein Rahmen zwischen den Zellen (collapse)	collapse, separate
border-spacing	Innerer Rahmenabstand zwischen Tabellenzellen	MASSE/INHEIT
caption-side	Position des Tabellentitels	bottom, top
table-layout	Legt den Algorithmus zur Ermittlung der Zellenbreite von Tabellen fest	auto, fixed

Tabelle: Tabellen-Properties

Listen

```
ol li {  
    list-style-type: lower-roman;  
}
```

Codebeispiel: Property zum Festlegen eines Listentyps

Property	Beschreibung	Wert
list-style	Shorthand-Property für Listenstil	Bsp.: list-style: square;
list-style-image	Bild als Listenaufzählungszeichen	url(), none
list-style-position	Position des Listenaufzählungszeichens	inside, outside
list-style-type	Listenaufzählungszeichen	TEXT, disc, circle, square, decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-roman, upper-roman, lower-latin, upper-latin, none

Tabelle: Listen-Properties

Cursor, Content und Counter

Property	Beschreibung	Wert
cursor	Darstellung des Mauszeigers (Bildateien dürfen max. 32x32 Pixel groß sein und es muss eine Alternative als Fallback angegeben werden, Bsp.: <code>cursor: url('hand.png'), pointer;</code>)	url(), auto, crosshair, default, move, text, wait, help, n-resize, e-resize, s-resize, w-resize, n-resize, nw-resize, se-resize, sw-resize, pointer, progress, not-allowed, no-drop, vertical-text, all-scroll, col-resize, row-resize, alias, cell, copy, ew-resize, ns-resize, nesw-resize, nwse-resize, none, inherit

Tabelle: Cursor-Property

```
body {
    counter-reset: ebene1;
}
h1::before {
    content: counter(ebene1, upper-roman) " ";
    counter-increment: ebene1;
}
```

Codebeispiel: Content und Counter

Property	Beschreibung	Wert
counter-increment	Counter hochzählen	COUNTERNAME [WERT]
counter-reset	Setzt einen Counter (zurück)	COUNTERNAME [WERT]
content	Fügt Inhalt vor (::before) oder nach (::after) einem Element	TEXT, url(), counter(name, style), attr(name), open-quote, close-quote, no-open-quote, no-close-quote
quotes	Legt die Zeichen für open-quote bzw. close-quote fest	Bsp.: quotes: '\201e' '\201c' ', ' ' ';

Tabelle: Content-Properties

Flexbox

Wenn das `display`-Property eines Elementes auf `flex` oder `inline-flex` gesetzt wird, wird dieses Element zu einem **Flex-Container**. Die Elemente direkt in diesem Container werden zu **Flex-Items**.

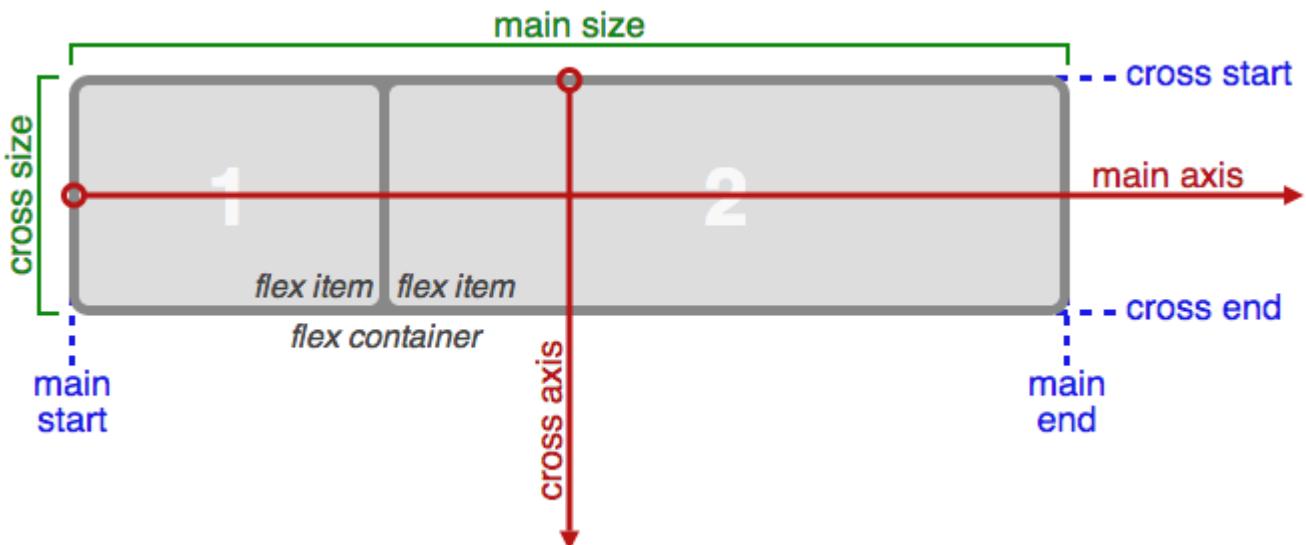


Abbildung: Flexbox

```
#container {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  flex-wrap: wrap;  
  align-content: center;  
  row-gap: 2em;  
}  
& img {  
  height: 160px;  
}
```

Codebeispiel: Flexbox CSS

```
<!DOCTYPE html>  
<div id="container">  
    
    
    
    
</div>
```

Codebeispiel: Flexbox HTML

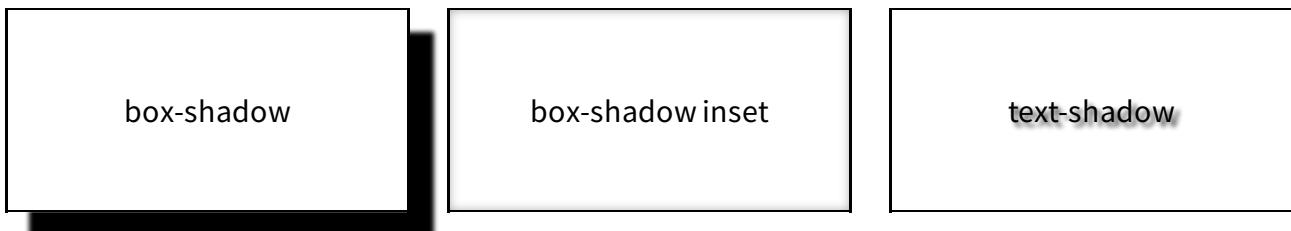
Property	Beschreibung	Wert
align-content	Ausrichtung von umgebrochenen Flex-Items (cross axis)	stretch, center, flex-start, flex-end, start, end, space-between, space-around, space-evenly
align-items	Ausrichtung der Flex-Items (cross axis)	stretch, center, flex-start, flex-end, start, end
flex	Shorthand-Property für flex-grow, flex-shrink und flex-basis	0 1 auto
flex-basis	Basisgröße eines Flex-Items	MASSEINHEIT, auto
flex-direction	Ausrichtung des Flex-Containers	row, row-reverse, column, column-reverse
flex-grow	Vergrößerung von Flex-Items	0, 1, 2, 3, ...
flex-shrink	Verkleinerung von Flex-Items	0, 1, 2, 3, ...
flex-wrap	Flex-Items umbrechen wenn nicht genug Platz vorhanden	nowrap, wrap
gap	Shorthand-Property für Zeilen- und Spaltenabstände	MASSEINHEIT
column-gap	Spaltenabstand	MASSEINHEIT
row-gap	Zeilenabstand	MASSEINHEIT
justify-content	Ausrichtung der Flex-Items (main axis)	flex-start, flex-end, center, space-between, space-around, space-evenly, start, end, left, right
order	Reihenfolge der Flex-Items	..., -1, 0, 1, 2, 3, ...

Tabelle: Flexbox-Properties

Schatteneffekte

Property	Beschreibung	Wert
box-shadow	Schatten	Bsp.: box-shadow: inset 0 0 8px darkgray;
text-shadow	Textschatten	Bsp.: text-shadow: 2px 2px 2px #333333;

Tabelle: Schatten-Properties



```
/* box-shadow: [inset] <OFFSETX> <OFFSETY> [BLUR] [FARBWERT] */
  box-shadow: inset 5em 1em gold;
  box-shadow: 12px 12px 4px black;

/* text-shadow: <OFFSETX> <OFFSETY> [BLUR] [FARBWERT] */
  text-shadow: 2px 2px #333333;
  text-shadow: 3px 3px rgb(255, 0, 0);
```

Codebeispiel: Schatten

Filter-Effekte und Transformationen

Property	Beschreibung	Wert
filter	Wendet eine oder mehrere Filterfunktionen an	blur(), brightness(), contrast(), drop-shadow(), grayscale(), hue-rotate(), invert(), saturate(), sepia()
transform	Mit Whitespace getrennte Liste von Transformationsfunktionen	matrix(), translate(), scale(), rotate(), skew(), none
transform-origin	Ausgangspunkt der Transformation	XOFFSET, YOFFSET, left, right, top, bottom, center

Tabelle: Filter-/Transform-Properties

Transitionen

```
.box {  
  display: block;  
  width: 100px;  
  height: 100px;  
  background-color: #0000ff;  
  transition: width 2s, height 2s, background-color 2s;  
}  
  
.box:hover {  
  background-color: #ffcccc;  
  width: 200px;  
  height: 200px;  
}
```

Codebeispiel: Transition-Property

Property	Beschreibung	Wert
transition	Shorthand-Property für Transitions	Bsp.: transition: width 2s, height 2s, background-color 2s;
transition-delay	Zeitverzögerung der Transition	ZEITEINHEIT
transition-duration	Zeitlänge der Transition	ZEITEINHEIT
transition-property	Transition auf bestimmte Properties einschränken	PROPERTY, all
transition-timing-function	Beschleunigungskurve der Transition	ease, ease-in, ease-out, ease-in-out, linear, step-start, step-end

Tabelle: Transition-Properties

Animationen

```
@keyframes jump {  
  from { top: 180px; }  
  50% { top: 0; }  
  to { top: 180px; }  
}  
  
#thing {  
  animation-name: jump;  
  animation-duration: 3s;  
}
```

Codebeispiel: CSS-Animation

Property	Beschreibung	Wert
animation	Shorthand-Property für Animationen	Bsp.: animation: jumping 3s linear infinite alternate;
animation-delay	Zeitverzögerung bis zum Start der Animation	ZEITEINHEIT
animation-direction	Richtung der Animation	normal, reverse, alternate, alternate-reverse
animation-duration	Zeitlänge der Animationen	ZEITEINHEIT
animation-iteration-count	Anzahl der Wiederholungen	infinite, 1, 2, 3, ...
animation-name	Namen einer oder mehrerer Animationssequenzen	NAME
animation-play-state	Animation anhalten/fortsetzen	running, paused
animation-timing-function	Beschleunigungskurve der Animation	ease, ease-in, ease-out, ease-in-out, linear, step-start, step-end

Tabelle: Animation-Properties

Nesting

```
/* ohne Nesting */
a.fancy-link { color: lime; }
a.fancy-link:hover { color: yellow; }

/* mit Nesting */
a.fancy-link {
    color: lime;
    &:hover { color: yellow; }
}
```

Codebeispiel: Verschachtelung von Rules (Nesting)

Weitere At-Rules

```
@charset "UTF-8";

@import url("base.css");
@import url("screen.css") screen;

@media print {
    a { text-decoration: none; }
    a::after { content: ' (' attr(href) ')'; }
}
```

Codebeispiel: At-Regeln

At-Regel	Beschreibung
<code>@charset WERT;</code>	Definiert den Zeichensatz einer CSS-Datei, der Wert muss in Anführungszeichen geschrieben werden (Bsp.: "UTF-8").
<code>@import url(URL) <MEDIAQUERIES> supports(<MEDIAQUERIES>);</code>	Bindet eine CSS-Datei in eine andere externe CSS-Datei ein
<code>@media MEDIAQUERIES { }</code>	Schränkt CSS-Regeln durch Media Queries ein
<code>@supports FEATURE { }</code>	Prüft ob bestimmte Feature vom Browser unterstützt werden

Tabelle: At-Regeln

Custom Properties

Mit **Custom Properties** lassen sich für Werte von CSS-Eigenschaften Variablen festlegen und später mit der Funktion `var()` verwenden. Sie müssen mit zwei Minus-Zeichen beginnen (--) .

```
:root {  
  --primary-color: blue;  
  --btn-padding: 10px;  
}  
  
button {  
  background-color: var(--primary-color);  
  padding: var(--btn-padding);  
  width: calc(120px + var(--btn-padding))  
}
```

Codebeispiel: Custom Properties

Vendor Prefixes

Nicht standardisierte Properties, Funktionen und Selektoren werden von den Browserherstellern mit einem Prefix beginnend mit einem Minus-Zeichen (-) versehen um diese zu kennzeichnen. Sollten die Properties, Funktionen oder Selektoren später in den Standard aufgenommen werden, wird das Prefix entfernt.

```
img:-webkit-full-screen, img:-moz-full-screen, img:fullscreen {  
  -webkit-border-radius: 0;  
  -moz-border-radius: 0;  
  border-radius: 0;  
}
```

Codebeispiel: Vendor Prefixes

Webbrowser	Prefix
⌚ Google Chrome, ⚡ Apple Safari, ⚡ Microsoft Edge	-webkit-*
⌚ Mozilla Firefox	-moz-*
⌚ Microsoft Edge, ⚡ Microsoft Internet Explorer	-ms-*

Tabelle: Vendor Prefixes

Berechnungen

Funktion	Beschreibung	Beispiel
calc()	Rechenausdruck mit +, -, *, /; Konstanten wie e oder pi können verwendet werden; Vor und nach + oder - muss zwingend ein Leerzeichen geschrieben werden!	height: calc(100vh - 20px - var(--foot-height));
clamp()	Wert auf eine Ober- und Untergrenze beschränken	font-size: clamp(1rem, 2.5vw, 2rem);
max()	Größten absoluten Wert ermitteln	width: max(20vw, 100px);
min()	Kleinsten (auch negativen) absoluten Wert ermitteln	width: min(50vw, 200px);

Tabelle: Mathematikfunktionen

Links

- ↗ Cascading Style Sheets - Wikipedia
(https://de.wikipedia.org/wiki/Cascading_Style_Sheets)
- ↗ W3C CSS Syntax Module Level 3 (<https://www.w3.org/TR/css-syntax-3/>)
- ↗ W3C CSS Selectors Level 3 (<https://www.w3.org/TR/selectors-3/>)
- ↗ W3C CSS Text Module Level 3 (<https://www.w3.org/TR/css-text-3/>)
- ↗ W3C CSS Color Module Level 3 (<https://www.w3.org/TR/css-color-3/>)
- ↗ W3C CSS Backgrounds and Borders Module Level 3 (<https://www.w3.org/TR/css-backgrounds-3/>)
- ↗ W3C CSS Box Model Module Level 3 (<https://www.w3.org/TR/css-box-3/>)
- ↗ W3C CSS Image Values and Replaced Content Module Level 3 (<https://www.w3.org/TR/css3-images/>)
- ↗ W3C CSS Values and Units Module Level 3 (<https://www.w3.org/TR/css3-values/>)
- ↗ W3C CSS Fonts Module Level 3 (<https://www.w3.org/TR/css-fonts-3/>)
- ↗ W3C CSS Flexible Box Layout Module Level 1 (<https://www.w3.org/TR/css-flexbox-1/>)
- ↗ W3C CSS Transitions (<https://www.w3.org/TR/css-transitions-1/>)
- ↗ W3C CSS Custom Properties for Cascading Variables Module Level 1
(<https://www.w3.org/TR/css-variables-1/>)
- ↗ W3C Media Queries (<https://www.w3.org/TR/css3-mediaqueries/>)
- ↗ W3C CSS Conditional Rules Module Level 3 (<https://www.w3.org/TR/css-conditional-3/>)
- ↗ W3C CSS Lists and Counters Module Level 3 (<https://www.w3.org/TR/css-lists-3/>)
- ↗ W3C CSS Generated Content Module Level 3 (<https://www.w3.org/TR/css-content-3/>)
- ↗ W3C CSS Filter Effects Module Level 1 (<https://www.w3.org/TR/filter-effects-1/>)
- ↗ W3C CSS Transforms Module Level 1 (<https://www.w3.org/TR/css-transforms-1/>)
- ↗ W3C CSS Animations Level 1 (<https://www.w3.org/TR/css-animations-1/>)
- ↗ W3C CSS Nesting Module (<https://www.w3.org/TR/css-nesting-1/>)
- ↗ W3C WOFF File Format (<https://www.w3.org/TR/WOFF/>)
- ↗ Microsoft OpenType specification (<https://www.microsoft.com/typography/otspec/>)
- ↗ Mozilla Developer Network CSS (<https://developer.mozilla.org/de/docs/Web/CSS>)
- ↗ Microsoft Edge Platform Status (<https://developer.microsoft.com/en-us/microsoft-edge/status/>)
- ↗ Can I use... Support tables for HTML5, CSS3, etc (<https://caniuse.com/>)
- ↗ Font Squirrel (<https://www.fontsquirrel.com/>)
- ↗ Google Fonts (<https://www.google.com/fonts>)
- ↗ CSS Diner Lernspiel (<https://flukeout.github.io/>)
- ↗ Flexbox Froggy Lernspiel (<https://flexboxfroggy.com/>)
- ↗ Flexbox Defense Lernspiel (<http://www.flexboxdefense.com/>)
- ↗ Vollkorn Schriftart (<http://vollkorn-typeface.com/>)