

Cascading Style Sheets

aus Wikipedia, der freien Enzyklopädie

Cascading Style Sheets (englische Aussprache [kæs,keɪdɪŋˈstaɪlʃiːts]; für *stufenförmige* oder (*hintereinander*) *geschachtelte Gestaltungsvorlagen*), kurz **CSS** genannt, ist eine deklarative Sprache für Stilvorlagen (engl. stylesheets) von strukturierten Dokumenten. Sie wird vor allem zusammen mit HTML und XML (zum Beispiel bei SVG) eingesetzt.


Grundidee beim Entwurf von CSS war es, mittels HTML oder XML nur die inhaltliche Untergliederung eines Dokumentes und die Bedeutung seiner Teile zu beschreiben, während mittels CSS weitgehend unabhängig davon die konkrete Darstellung (Farben, Layout, Schriftigenschaften usw.) der Teile festgelegt wird. Elemente eines Dokumentes können z. B. aufgrund ihres Elementnamens (z. B. alle Linkelemente), ihrer ID oder auch aufgrund ihrer Position in diesem Dokument (z. B. alle Bildelemente innerhalb von Linkelementen) identifiziert werden. Mithilfe von CSS-Regeln können deshalb für jedes Element die konkreten Werte für bestimmte Darstellungsattribute festgelegt werden. Diese Festlegungen können an zentraler Stelle erfolgen, ggf. sogar in einem separaten Dokument, was ihre Wiederverwendung für andere Dokumente erleichtert. Daneben enthält CSS ein Vererbungsmodell für Auszeichnungsattribute – deshalb *cascading* – das die Anzahl nötiger Definitionen vermindert. CSS ermöglicht allenfalls auch die diskretionäre Auszeichnung von Teilen eines Dokuments.

Mit CSS ist es möglich, für verschiedene Ausgabemedien (Bildschirm, Papier, Projektion, Sprache) unterschiedliche Darstellungen vorzugeben. Das ist nützlich, um zum Beispiel Hyperlinks beim Drucken extra aufzuführen oder für Geräte mit geringerer Auflösung (zum Beispiel PDAs oder Mobiltelefone) die Anzeige mit Rücksicht auf die geringere Seitenbreite und -höhe anzupassen.

Neben verschiedenen Möglichkeiten, Farben und Schriften einzustellen, erlaubt CSS auch, Elemente detaillierter zu gestalten, frei zu positionieren oder mit CSS 3 zu animieren.

CSS gilt heute als die Standard-Stylesheetsprache für Webseiten. Die früher üblichen HTML-Elemente wie *font* oder *center* sind als deprecated (missbilligt) gekennzeichnet, das heißt, sie sollen in Zukunft aus dem HTML-Standard entfernt werden.^[1] Grund ist eine Ausgliederung der designtechnischen Angaben aus dem reinen HTML-Quelltext, der im Rahmen des Markups nur die Struktur einer Seite wiedergibt, jedoch keine Hinweise auf Optik und Gestaltung gibt.

Cascading Style Sheets (CSS)



Dateiendung:

MIME-Type:

Entwickelt von:

Art:

Standard(s):

.css

text/css

World Wide Web Consortium

Stylesheet-Sprache

Level 1 (Recommendation)
(http://www.w3.org/TR/CSS1)

Level 2 (Recommendation)
(http://www.w3.org/TR/CSS2/)

Level 2 Revision 1 (Recommendation)
(http://www.w3.org/TR/CSS21/)

Inhaltsverzeichnis

- 1 Geschichte und Versionen
 - 1.1 Anfänge
 - 1.2 CSS2 und 2.1
 - 1.3 CSS3
 - 1.4 Nachfolger
- 2 Syntax
 - 2.1 Der Aufbau von CSS-Regeln
 - 2.2 Selektoren
 - 2.3 Beispiel
 - 2.4 CSS-Hacks
- 3 Kombination mit HTML oder XHTML
- 4 Medienspezifische Stylesheets
- 5 Eigenschaftsspezifische Stylesheets (Media Queries)
- 6 Siehe auch
- 7 Literatur
- 8 Weblinks
- 9 Einzelnachweise

Geschichte und Versionen

Anfänge

Der erste Vorschlag für Web-Stylesheets kam 1993 auf, mehrere weitere folgten bis 1995. Cascading Style Sheets (CSS), wie sie aktuell bekannt sind, wurden 1994 von Håkon Wium Lie vorgeschlagen. Bert Bos arbeitete zu dieser Zeit an der Implementierung eines Browsers namens *Argo*, der seine eigene Stylesheet-Sprache benutzte. Die beiden entschieden sich, zusammenzuarbeiten und CSS zu entwickeln.

Es gab zu dieser Zeit auch andere Sprachen mit dem gleichen Ziel, die Erfinder von CSS brachten aber als erste die Idee auf, Regeln zu definieren, die über mehrere Stylesheets hinweg vererbt werden konnten.

Nach der Präsentation von CSS durch Lie auf der Konferenz „Mosaic and the Web“ in Chicago 1994, und später mit Bos 1995 wurde das World Wide Web Consortium (W3C) auf CSS aufmerksam. Lie und Bos arbeiteten mit anderen Mitgliedern in diesem Rahmen an CSS weiter. Im Dezember 1996 wurde dann die *CSS Level 1 Recommendation* publiziert. Diesen Normierungsvorschlag folgten die aktuellen Browser mittlerweile fast vollständig.

CSS2 und 2.1

CSS Level 2 (CSS2) wurde im Mai 1998 veröffentlicht. Bis heute (Stand Januar 2010) wird diese Empfehlung von keinem verbreiteten Browser vollständig implementiert. Manche Browser setzen den größten Teil von CSS2 korrekt um, andere hingegen bieten nur eine mangelhafte Unterstützung von wenigen Teiltechniken. Bei der Verwendung im Web ergeben sich daher oft Schwierigkeiten.

Seit 2002 arbeitete das W3C an einer Zwischenversion *CSS Level 2 Revision 1* (CSS2.1),^[2] die die Erfahrungen mit CSS2 berücksichtigt, Unstimmigkeiten korrigiert und diejenigen Teiltechniken streicht, die bisher in verschiedenen Browsern nicht korrekt implementiert wurden. Am 7. Juni 2011 wurde CSS2.1 als Recommendation veröffentlicht.^[3] CSS2.1 bringt selbst keine grundlegend neuen Fähigkeiten mit sich.

CSS3

Seit 2000 ist *CSS Level 3* in der Entwicklung.^[4] Hier werden die Entwicklungen weiter vorangetrieben, die bereits mit CSS2 begonnen wurden. CSS3 wird im Gegensatz zu den Vorgängern modular aufgebaut sein, womit einzelne Teiltechniken (beispielsweise Steuerung der Sprachausgabe oder Selektoren) in eigenen Versionsschritten entwickelt werden können. So nähert sich CSS bei seinen Fähigkeiten mehr dem etablierten DSSSL (für SGML) an und wird wohl auch in Zukunft noch eine Alternative zu XML-basierten Stylesheet-Sprachen wie XSL-FO sein. Derzeit veröffentlichte und breit unterstützte Standards sind unter anderem *CSS Color Level 3*, *CSS Namespaces*, *Selectors Level 3* und *Media Queries*. Neben diesen Modulen stehen weitere Elemente zur Diskussion, etwa ein Layout-Modul oder verschiedene Grafikfilter.^[5] Moderne Browser unterstützen jedoch schon viele CSS3-Module, die noch nicht durch das W3C standardisiert sind.^[6]

Nachfolger

Im Frühjahr 2012 gab es mehrere Berichte, das W3C arbeite an einem Nachfolger von CSS3. Dieser sollte naturgemäß unter der Bezeichnung CSS4 veröffentlicht werden und unter anderem die Selektoren nochmals leistungsfähiger machen und gleichzeitig vereinfachen.^[7] Im September 2012 haben Vertreter der entsprechenden Arbeitsgruppe des W3C jedoch klargestellt, dass es CSS4 nicht geben wird: "There will never be a CSS4."^[8] Vielmehr soll die künftige Entwicklung des Standards in Form einer schrittweisen Aktualisierung der Module von CSS3 geschehen.

Syntax

Der Aufbau von CSS-Regeln

```
Selektor [, Selektor2, ...]
{
    Eigenschaft-1: Wert-1;
    ...
    Eigenschaft-n: Wert-n[;]
}
/* Kommentar */
/* In eckigen Klammern stehen optionale Angaben */
```

Eine CSS-Regel gibt an, dass für durch sie festgelegte Teile eines Dokuments eine bestimmte, ebenfalls in ihr angegebene Eigenschaftskombination gelten soll. Sie wird geschrieben als eine kommagetrennte Liste von Selektoren (*"Für diese Typen von Teilen ..."*), gefolgt von einer in geschweiften Klammern stehenden, semikolongetrennten Liste von Eigenschafts-Deklarationen (*"... nimm alle diese Eigenschaften!"*); nach der letzten Eigenschafts-Deklaration ist ein abschließendes Semikolon erlaubt. Eine Eigenschafts-Deklaration besteht aus deren Namen und der Angabe, welchen Wert sie annehmen soll; diese Name-Wert-Paare werden ihrerseits durch einen Doppelpunkt getrennt.

Um diese Teile einer Regel herum ist Weißraum frei verwendbar; häufig schreibt man den Doppelpunkt ohne Zwischenraum hinter den Eigenschaftsnamen, jede Eigenschaftsdekларation auf eigener Zeile und schließt auch die letzte Eigenschaft mit einem Semikolon – so kommt es zu weniger Syntaxfehlern bei Änderungen.

Ein Stylesheet darf beliebig viele solcher Regeln enthalten. Die folgende Tabelle enthält eine vollständige Übersicht aller Selektoren, mit denen Elemente (meist HTML-Elemente) ausgewählt werden können.

Selektoren

Ein Selektor nennt die Bedingungen, die auf ein Element zutreffen müssen, damit der nachfolgende Satz an CSS-Dekларationen mit seinen Darstellungsvorgaben auf das Element angewendet wird. Solche Bedingungen beschreiben unzweideutig, welche Eigenschaften (Typ, Klasse, ID, Attribut oder Attributwert) Elemente haben müssen oder in welchem Kontext sie im Dokument stehen müssen (Existenz eines bestimmten übergeordneten Elementes oder eines Vorgängerelementes bestimmten Typs), damit die Darstellungsvorgaben für sie gelten sollen. In einem Selektor können mehrere Auswahlkriterien verknüpft sein.

Muster	Bedeutung	Eingeführt in	Definition	Erläuterung
<i>einfache Selektoren</i>				
*	Selektiert jedes Element	CSS2	Universal selector (http://www.w3.org/TR/CSS21/selector.html#universal-selector)	Der Universalselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Universal-selector)
E	Selektiert jedes Element vom Typ E	CSS1	Type selectors (http://www.w3.org/TR/CSS21/selector.html#type-selectors)	Der Typselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Type-selector)
.c	Selektiert jedes Element der Klasse c (analog [class~='c'])	CSS1	Class selectors (http://www.w3.org/TR/CSS21/selector.html#class-html)	Der Klassenselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Class-selector)
#myid	Selektiert das Element mit der ID „myid“.	CSS1	ID selectors (http://www.w3.org/TR/CSS21/selector.html#id-selectors)	Der ID-Selektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/ID-selector)
[foo]	Selektiert Elemente, deren „foo“-Attribut gesetzt ist (unabhängig vom Wert)	CSS2		Attributpräsenz (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Attributselector/Pr%C3%A4senz)
[foo=bar]	Selektiert Elemente, deren „foo“-Attribut mit dem Wert „bar“ gesetzt ist	CSS2		

[foo~=bar]	Selektiert Elemente, deren „foo“-Attribut das Wort „bar“ enthält (also beispielsweise „bar“ oder auch „bar quz“ aber nicht „barquz“)	CSS2	Attribute selectors (http://www.w3.org/TR/CSS21/selector.html#attribute-selectors)	Attributwerte (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Attributselektoren)
[foo =bar]	Selektiert Elemente, deren „foo“-Attribut exakt die Zeichenfolge „bar“ ist oder mit der Zeichenfolge „bar-“ beginnt	CSS2		
[foo^=bar]	Selektiert Elemente, deren „foo“-Attribut mit der Zeichenfolge „bar“ beginnt	CSS3	Substring matching attribute selectors (http://www.w3.org/TR/css3-selectors/#attribute-substrings)	Teilübereinstimmungen (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Attributselektor/Teil%C3%
[foo\$=bar]	Selektiert Elemente, deren „foo“-Attribut mit der Zeichenfolge „bar“ endet	CSS3		
[foo*=bar]	Selektiert Elemente, deren „foo“-Attribut die Zeichenfolge „bar“ enthält	CSS3		

Kombinatoren

E F	Selektiert alle Elemente F, die Nachfahren eines Elements E sind	CSS1	Descendant selectors (http://www.w3.org/TR/CSS21/selector.html#descendant-selectors)	Der Nachfahrenselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Kombinator/Nachfahrenselektor)
E > F	Selektiert alle Elemente F, die ein Kind eines Elements E sind	CSS2	Child selectors (http://www.w3.org/TR/CSS21/selector.html#child-selectors)	Der Kindselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Kombinator/Kindselektor)
E ~ F	Selektiert alle Elemente F, die einen Vorgänger E auf gleicher Ebene haben	CSS3	General sibling combinator (http://www.w3.org/TR/css3-selectors/#general-sibling-combinators)	Der Geschwisterselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Kombinator/Geschwisterselektor)
E + F	Selektiert alle Elemente F, die einen direkten Vorgänger E auf gleicher Ebene haben	CSS2	Adjacent sibling combinator (http://www.w3.org/TR/css3-selectors/#adjacent-sibling-combinators)	Der Nachbarselektor (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Kombinator/Nachbarselektor)

Pseudoklassen

:link	Selektiert unbesuchte Verweise	CSS1	The link pseudo-classes (http://www.w3.org/TR/CSS2/selector.html#link-pseudo-classes)	Die Pseudoklassen :link und :visited (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/link,_visited)
:visited	Selektiert bereits besuchte Verweise	CSS1		
:active	Selektiert das Element, das gerade angeklickt ist	CSS1	The dynamic pseudo-classes (http://www.w3.org/TR/CSS2/selector.html#dynamic-pseudo-classes)	Die Pseudoklassen :hover, :active und :focus (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/hover,_active,_focus)
:hover	Selektiert das Element, über dem sich der Mauszeiger befindet	CSS2		
:focus	Selektiert das Element, das den Fokus (z. B. durch die Tabulatortaste) erhalten hat	CSS2		
:lang()	Selektiert Elemente, deren Sprache über das Attribut „lang“ festgelegt wurde	CSS2	The language pseudo-class (http://www.w3.org/TR/CSS2/selector.html#lang)	Die Pseudoklasse :lang() (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/lang)
:target	Selektiert das Element, das Ziel des gerade eben angeklickten	CSS3	The target pseudo-class (http://www.w3.org/TR/css3-selectors/#target-pseudo)	Die Pseudoklasse :target (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/target)

	Verweises ist			
:enabled	Selektiert Elemente in Benutzeroberflächen, die auswählbar sind	CSS3	The :enabled and :disabled pseudo-classes (http://www.w3.org/TR/css3-selectors/#enableddisabled)	Die Pseudoklassen :enabled, :disabled und :checked (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/disabled,_e
:disabled	Selektiert Elemente in Benutzeroberflächen, die gesperrt sind	CSS3		
:checked	Selektiert Elemente von Benutzeroberflächen, die angewählt sind	CSS3		
:not(foo)	Selektiert Elemente, für die „foo“ nicht zutrifft	CSS3	The negation pseudo-class (http://www.w3.org/TR/css3-selectors/#negation)	Die Pseudoklasse :not() (http://wiki.selfhtml.org/wiki/CSS/Selektoren/P
<i>strukturelle Pseudoklassen</i>				
:first-child	Selektiert Elemente, die das erste Kind ihres Elternelementes sind	CSS2	:first-child pseudo-class (http://www.w3.org/TR/CSS21/selector.html#first-child)	Die Pseudoklasse :first-child (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_child)
:last-child	Selektiert Elemente, die das letzte Kind ihres Elternelementes sind	CSS3	:last-child pseudo-class (http://www.w3.org/TR/css3-selectors/#last-child-pseudo)	Die Pseudoklasse :last-child (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_child)
:nth-child(...)	Selektiert Elemente, die beispielsweise das <i>n</i> -te Kind ihres Elternelementes sind	CSS3	:nth-child() pseudo-class (http://www.w3.org/TR/css3-selectors/#nth-child-pseudo)	Die Pseudoklasse :nth-child() (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_child)
:nth-last-child(...)	Selektiert Elemente, die beispielsweise das <i>n</i> -te Kind ihres Elternelementes sind, Zählung beginnt von hinten	CSS3	:nth-last-child() pseudo-class (http://www.w3.org/TR/css3-selectors/#nth-last-child-pseudo)	Die Pseudoklasse :nth-last-child() (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_last-child)
:only-child	Selektiert Elemente, die das einzige Kind ihres Elternelementes sind	CSS3	:only-child pseudo-class (http://www.w3.org/TR/css3-selectors/#only-child-pseudo)	Die Pseudoklasse :only-child (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_child)
:first-of-type	Selektiert Elemente, die das erste Element dieses Typs ihres Elternelementes sind	CSS3	:first-of-type pseudo-class (http://www.w3.org/TR/css3-selectors/#first-of-type-pseudo)	Die Pseudoklasse :first-of-type (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_of-type)
:last-of-type	Selektiert Elemente, die das letzte Element dieses Typs ihres Elternelementes sind	CSS3	:first-of-type pseudo-class (http://www.w3.org/TR/css3-selectors/#last-of-type-pseudo)	Die Pseudoklasse :last-of-type (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_of-type)
:nth-of-type(...)	Selektiert Elemente, die beispielsweise das <i>n</i> -te Element dieses Typs ihres Elternelementes sind	CSS3	:nth-of-type() pseudo-class (http://www.w3.org/TR/css3-selectors/#nth-of-type-pseudo)	Die Pseudoklasse :nth-of-type() (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_type)
:nth-last-of-type(...)	Selektiert Elemente, die beispielsweise das <i>n</i> -te Element dieses Typs ihres Elternelementes sind, Zählung beginnt von hinten	CSS3	:nth-last-of-type() pseudo-class (http://www.w3.org/TR/css3-selectors/#nth-last-of-type-pseudo)	Die Pseudoklasse :nth-last-of-type() (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_last-of-type)
:only-of-type	Selektiert Elemente, die das einzige Element dieses Typs ihres Elternelementes sind	CSS3	:only-of-type pseudo-class (http://www.w3.org/TR/css3-selectors/#only-of-type-pseudo)	Die Pseudoklasse :only-of-type (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_of-type)
:root	Selektiert das Wurzelement	CSS3	:root pseudo-class (http://www.w3.org/TR/css3-selectors/#root-pseudo)	Die Pseudoklasse :root (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_
:empty	Selektiert Elemente, die keine Kinder haben	CSS3	:empty pseudo-class (http://www.w3.org/TR/css3-selectors/#empty-pseudo)	Die Pseudoklasse :empty (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoklasse/strukturelle_
<i>Pseudoelemente</i>				

::first-line	Selektiert die erste Zeile des formatierten Textes	CSS1	The ::first-line pseudo-element (http://www.w3.org/TR/css3-selectors/#first-line)	Das Pseudoelement ::first-line (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoelement/first-line)
::first-letter	Selektiert das erste Zeichen des formatierten Textes	CSS1	The ::first-letter pseudo-element (http://www.w3.org/TR/css3-selectors/#first-letter)	Das Pseudoelement ::first-letter (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoelement/first-letter)
foo::before	Erzeugt und selektiert ein Element vor einem Element „foo“	CSS2	The ::before and ::after pseudo-elements (http://www.w3.org/TR/css3-selectors/#gen-content)	Die Pseudoelemente ::before und ::after (http://wiki.selfhtml.org/wiki/CSS/Selektoren/Pseudoelement/before,_a)
foo::after	Erzeugt und selektiert ein Element nach einem Element „foo“	CSS2		

Beispiel

CSS:

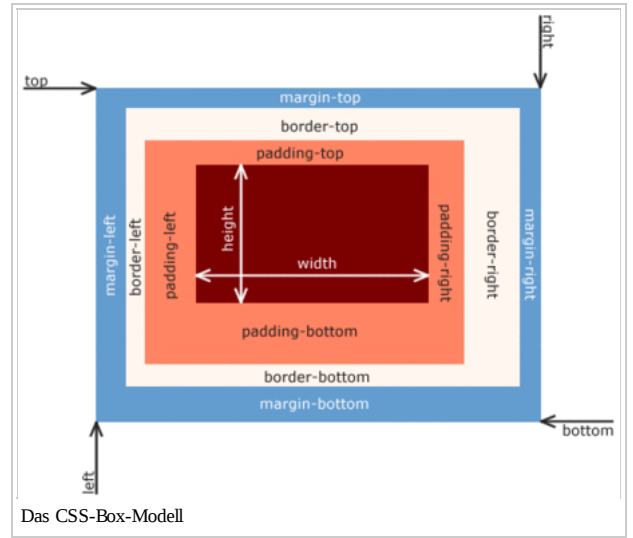
```
p.info {
  font-family: arial, sans-serif;
  line-height: 150%;
  margin-left: 2em;
  padding: 1em;
  border: 3px solid red;
  background-color: #f89;
  display: inline-block;
}
p.info span {
  font-weight: bold;
}
p.info span::after {
  content: " ";
}
```

HTML:

```
<p class="info">
  <span>Hinweis</span>
  Sie haben sich erfolgreich angemeldet.
</p>
```

Die p-Tags machen den dazwischen stehenden Text zu einem Absatz. Da ihm die Klasse „info“ zugewiesen wird, wird er von einem CSS-kompatiblen Browser wie folgt dargestellt:

Hinweis: Sie haben sich erfolgreich angemeldet.



Hier werden die Deklarationen allen p-Elementen zugewiesen, die das class-Attribut mit dem Wert info besitzen. Würde man das p im Selektor weglassen, würden alle Elemente der Klasse info betroffen sein, beim Weglassen des . info alle p-Elemente. span-Elemente innerhalb solcher Absätze werden in Fettschrift dargestellt; der Doppelpunkt wird als Pseudoelement erzeugt.

Ein wichtiges Prinzip von CSS ist die Vererbung der Eigenschaftswerte an untergeordnete Elemente und die Kombination verschiedener Stylesheets (Kaskadeneffekt). Diese können dabei aus verschiedenen Quellen stammen: vom Autor des Stylesheets, dem Browser (User Agent) oder dem Benutzer.

CSS-Hacks

Eine Anwendung von CSS-Syntax bei der Gestaltung von Weblayouts sind sogenannte CSS-Hacks. Sie werden benutzt, um Unterschiede bei der Darstellung von Weblayouts in verschiedenen Browsern auszugleichen oder CSS-Anweisungen für bestimmte Webbrowser gesondert zuzuweisen oder auszuschließen. Der Begriff Hack bezeichnet dabei nichtstandardisierte CSS-Befehle, mit denen die Interpretationsschwäche eines Webbrowsers ausgenutzt wird, der diese Anweisungen entweder interpretiert oder ignoriert. Damit können Schwachstellen von Browsern ausgeglichen werden, um möglichst in jedem Webbrowser das gleiche Ergebnis angezeigt zu bekommen.

Ein CSS-Hack kombiniert z. B. fehlerhaft angegebene Selektoren in Kombination mit zusätzlichen Zeichen oder enthält Anweisungen, die bestimmte Webbrowser nicht kennen. Ein bekanntes Beispiel für einen CSS-Hack ist der sogenannte Star-HTML-Hack. Das *-Zeichen dient als Universal-Selektor und ist vor dem Selektor html sinnlos.

CSS-Code-Beispiel:

```
p { background-color: blue; }
* html p { background-color: red; }
```

In diesem Fall würden zunächst alle Browser die p-Elemente mit einem blauen Hintergrund darstellen. Lediglich der Internet Explorer vor Version 7 interpretiert auch die zweite Zeile und färbt die Absätze rot, obwohl <html> kein Eltern-Element besitzt, auf das * zutreffen könnte.

Kombination mit HTML oder XHTML

Am häufigsten wird CSS mit HTML oder XHTML kombiniert. Dies kann an mehreren Orten geschehen, hier einige Beispiele:

- Als externes Stylesheet für eine (X)HTML-Datei (Link-Element)

```
<link rel="stylesheet" type="text/css" href="beispiel.css" />
```

- Als externes Stylesheet für eine XHTML-Datei (xml-stylesheet-Verarbeitungsanweisung)

```
<?xml-stylesheet type="text/css" href="beispiel.css" ?>
```

- Als internes Stylesheet in einer (X)HTML-Datei (style-Element)

```
<head>
  <title>Dokument mit Formatierungen</title>
  <style type="text/css">
    body { color: purple; background-color: #d8da3d }
  </style>
</head>
```

- Innerhalb von (X)HTML-Tags (style-Attribut)

```
<span style="font-size: small;">Text</span>
```

Die Einbindung als externes Stylesheet ist dabei die am häufigsten verwendete Methode. Sie bietet den Vorteil, dass für mehrere Dokumente, die denselben Regelsatz benutzen, das Stylesheet nur einmal heruntergeladen werden muss. Auch vermeidet man so sich wiederholenden Code. CSS selbst ermöglicht durch die @import-Regel das Einbinden von weiteren externen Stylesheets.

```
<head>
  <title>Beispiel</title>
  <style type="text/css">
    @import url(url_des_stylesheets);
  </style>
</head>
```

Es gibt drei Varianten Stylesheets mittels eines link-Elements einzubinden, die sich darin unterscheiden, wie zwingend die Stylesheets berücksichtigt werden:

Dauerhafte Stylesheets (persistent)

```
<link rel="stylesheet" type="text/css" href="beispiel.css" />
```

Wenn man ein Stylesheet so einbindet, wird es auf jeden Fall verwendet.

Bevorzugte Stylesheets (preferred)

```
<link rel="stylesheet" type="text/css" href="beispiel.css" title="IrgendeinTitel" />
```

Sollte man diese Einbindung verwenden, wird das Stylesheet verwendet, bis der Benutzer ein anderes auswählt.

Alternative Stylesheets (alternate)

```
<link rel="alternate stylesheet" type="text/css" href="beispiel.css" title="IrgendeinTitel" />
```

Wird das Stylesheet so mit dem HTML-Dokument verknüpft, muss der Benutzer ausdrücklich wählen dieses zu verwenden. Das wirkt sich in den meisten Browsern aus (z. B. Internet Explorer, Firefox, Opera und Konqueror). Somit wird diese Funktion von den meist benutzten Browsern implementiert. Außerdem sollte ein "alternate stylesheet" nur in Verbindung mit einem anderen fest eingebundenen verwendet werden, damit es auch nur eine echte Alternative ist.

Medienspezifische Stylesheets

Es ist auch möglich, für verschiedene Medien verschiedene Stylesheets einzubinden, um zum Beispiel die Gestaltung beim Drucken oder auf Handy-Displays zu regulieren. Diesen Zweck erfüllt das Attribut media. In dieses Attribut werden die Parameter notiert, die für dieses Stylesheet gelten sollen.

Parameter	Bedeutung
all	Stylesheet gilt für alle Ausgabegeräte
speech (CSS2: aural)	aurale Ausgabegeräte (z. B. Screenreader)
braille	Blindenschriftfähige Ausgabegeräte
embossed	Blindenschriftfähige Drucker
handheld	Handhelds (Palmtops, PDAs, WinCE-Geräte)
print	Drucker
projection	Video-Beamer, Overhead-Projektoren
screen	Bildschirme
tty	Ausgabegeräte mit feststehenden Zeichentypen (z. B. Fernschreiber, Terminals oder auch ältere Mobiltelefone)
tv	TV-Geräte

```
@media print {
  body {
    color: black;
    background-color: white;
  }
  h1 {
    font-size: 14pt;
  }
}
```

```

}
.navigation {
    display: none;
}
}

```

Eigenschaftsspezifische Stylesheets (Media Queries)

Bei Media Queries handelt es sich um ein Konzept, welches mit CSS3 eingeführt wurde und das Prinzip des Medientyps in CSS2 erweitert. Anstatt starr zu definieren, welches Medium das Zielmedium ist, können mit Media Queries die Eigenschaften des aktuellen Gerätes direkt abgefragt werden. Verfügbare Geräteeigenschaften sind zum Beispiel:

- Breite und Höhe des Browserfensters
- Breite und Höhe des Gerätes
- Orientierung (Quer- oder Hochformat)
- Bildschirmauflösung

Vor allem im Bereich der mobilen Webprogrammierung werden Media Queries bereits jetzt häufig eingesetzt, um die Webseite ideal an das aktuell verwendete Gerät anzupassen.

Im folgenden Beispiel werden Elemente mit CSS-Regeln versehen. Diese Regeln gelten für das gesamte Dokument. Anschließend wird eine Media Query eingesetzt, die greift, sobald die Breite des Browserfensters kleiner als 1025 Pixel ist. In diesem Fall ändern sich die Eigenschaften, die vorher allgemein definiert wurden bzw. gelten zusätzliche Eigenschaften.

```

#inhalt {
    width: 800px;
}

@media screen and (max-width: 1024px) {
    #inhalt {
        width: 600px;
    }

    aside {
        display: none;
    }
}

```

Siehe auch

- Reset-Stylesheet

Literatur

- Ingo Chao, Corina Rudel: *Fortgeschrittene CSS-Techniken*. 3. Auflage. Galileo Computing, Bonn 2012, ISBN 978-3-8362-1695-1 (Detailreiche und anspruchsvolle Erklärung von CSS-Internia).
- Michael Jendryschik: *Einführung in XHTML, CSS und Webdesign*. Addison-Wesley, München 2007, ISBN 978-3-8273-2477-1 (Standardkonforme, moderne und barrierefreie Websites erstellen).
- Kai Laborenz: *CSS Praxis*. 3. Auflage. Galileo Computing, Bonn 2005, ISBN 3-89842-577-0 (Grundlagen und Referenz, Praxisbeispiele, Tipps und Tricks, Barrierefreies Webdesign mit CSS).
- Bettina K. Lechner, Bernhard Stockmann: *CSS pur – Ultimative Weblösungen mit Stil*. Addison Wesley, München 2009, ISBN 978-3-8273-2781-9 (Moderne CSS-Techniken: kompakter Grundlagenteil, 3 Website-Prototypen mit Detailerklärungen).
- Florence Maurice: *Jetzt lerne ich CSS3*. Markt+Technik, München 2012, ISBN 978-3-8272-4745-2 (Einführung in CSS und ausgewählte Techniken aus CSS3).
- Florence Maurice: *CSS3: Leitfaden für Webdesigner*. Addison-Wesley, München 2011, ISBN 978-3-8273-3031-4 (Neue Features in CSS3).
- Jens Meiert: *Webdesign mit CSS*. 2. Auflage. O'Reilly, 2007, ISBN 978-3-89721-712-6 (Methodik bei Arbeit mit CSS, Praxisbeispiele).
- Peter Müller: *Little boxes. Webseiten gestalten mit CSS. Grundlagen*. Markt+Technik, München 2007, ISBN 978-3-8272-4224-2 (aktualisierte, erweiterte Ausgabe).
- Christopher Schmitt: *CSS Kochbuch*. 2. Auflage. O'Reilly, 2007, ISBN 978-3-89721-492-7 (Grundlagen und Beispiele zu CSS).
- Heiko Stiebert: *CSS-Design – Die Tutorials für Einsteiger*. 1. Auflage. Galileo Design, Bonn 2008, ISBN 978-3-8362-1155-0 (über 30 praxisnahe Tutorials, inkl. 4 Exkursen zum Thema CSS).
- Heiko Stiebert: *Modernes Webdesign mit CSS: Schritt für Schritt zur perfekten Website*. 1. Auflage. Galileo Design, Bonn 2011, ISBN 978-3-8362-1666-1 (über 20 praxisnahe Tutorials).
- Dirk Jesse: *CSS-Layouts*. 2. Auflage. Galileo Design, Bonn 2008, ISBN 978-3-8362-1135-2 (Grundlagen, Praxiseinsatz und Integration).

Weblinks

 **Wikibooks: Websiteentwicklung: CSS** – Lern- und Lehrmaterialien

- W3C: Cascading Style Sheets (<http://www.w3.org/Style/CSS/>) (englisch)
- CSS-2-Spezifikation (<http://www.edition-w3c.de/TR/REC-CSS2/>) – deutsche Übersetzung der Spezifikation
- Doktorarbeit von Håkon Wium Lie (<http://people.opera.com/howcome/2006/phd/>) – Geschichte, alternative Formate, Designgrundsätze von CSS (englisch)
- SELFHTML (<http://de.selfhtml.org/css/>) – Ausführliche Darstellung, Verwendung zusammen mit HTML. Download möglich.
- CSS-Tutorial (<http://de.html.net/tutorials/css/>) für einen kurzen Einstieg in CSS
- Browserkompatibilität (<http://caniuse.com/>) – Ausführliche Tabelle zur Unterstützung von CSS-Eigenschaften und Selektoren gängiger Webbrowser

Einzelnachweise

1. SELFHTML (http://de.selfhtml.org/html/text/schrift.htm#art_groesse_farbe), Kompendium zu HTML und CSS, deutsch
2. *Cascading Style Sheets, level 2 revision 1 CSS2.1 Specification W3C Working Draft 2 August 2002* (<http://www.w3.org/TR/2002/WD-CSS21-20020802/>). W3C. Abgerufen am 11. Oktober 2010.
3. *Cascading Style Sheets Standard Boasts Unprecedented Interoperability* (<http://www.w3.org/2011/05/css-pr>). W3C. Abgerufen am 7. Juni 2011.
4. *CSS3 introduction W3C Working Draft, 14 April 2000*. (<http://www.w3.org/TR/2000/WD-css3-roadmap-20000414>) W3C, abgerufen am 8. Juli 2010 (englisch).
5. w3.org: CSS Current Work (<http://www.w3.org/Style/CSS/current-work>)
6. caniuse.com: Vergleich von CSS3-Unterstützung aktueller Browser, abgerufen am 10. Mai 2012

(<http://caniuse.com/#compare=ie+9,firefox+12,chrome+18,safari+5.1,opera+11.6>)

7. Nicolai Schwarz: *CSS4: So sieht die Zukunft des Webstandards aus*. (<http://t3n.de/magazin/webstandards-aktuell-css4-heute-schon-morgen-denken-229619/>) In: *t3n Magazin*. 5. September 2012, abgerufen am 21. November 2012.
8. Tob Atkins: *A Word About CSS4*. (<http://www.xanthir.com/b4Ko0>) 5. September 2012, abgerufen am 21. November 2012 (englisch).

Normdaten (Sachbegriff): GND: 4467617-7

Von „http://de.wikipedia.org/w/index.php?title=Cascading_Style_Sheets&oldid=122399801“

Kategorie: Cascading Style Sheets

- Diese Seite wurde zuletzt am 10. September 2013 um 10:55 Uhr geändert.
- Abrufstatistik

Der Text ist unter der Lizenz „Creative Commons Attribution/Share Alike“ verfügbar; Informationen zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden.
Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.