

Die vielen Möglichkeiten, CSS einzubinden, weisen schon auf eine Besonderheit der Sprache hin: Das »C« in CSS steht für »Cascading« (kaskadierend). Damit ist gemeint, dass mehrere Stylesheets zusammenwirken können, nacheinander oder parallel, selbst wenn sie konkurrierende Regeln enthalten.

Reihenfolge:

1. Finde alle Deklarationen, die in diesem Ausgabemedium für das Element angewandt werden müssen.
2. Sortiere die Deklarationen aufsteigend nach Wichtigkeit.
Zuerst die Deklarationen des Browserstylesheets, dann die der Userstylesheets, dann die des Autors, dann die des Autors mit einem !important und zuletzt die des Userstylesheets mit einem !important.
3. Sortiere die Deklarationen in der vorgegebenen Reihenfolge.
Haben zwei Regeln dieselbe Gewichtung, denselben Ursprung, erhält die jeweils zuletzt angegebene den Vorrang.

Einbindung von CSS in ein XHTML-Dokument:

Es gibt drei gebräuchliche Methoden, CSS in ein XHTML-Dokument zu integrieren.

Einbindung per *STYLE* - Attribut:

Das style-Attribut ist ein Universalattribut und darf an fast allen XHTML-Elementen notiert werden.

```
<p style="color: silver; font-weight: bold;">Absatz</p>
```

Diese sogenannten **Inline-Styles** werden häufig verwendet, wenn Angaben nur ein einziges Mal für ein bestimmtes Element vorgenommen werden sollen oder die Vergabe einer Klasse oder ID nicht in Frage kommt oder nicht möglich ist.

Nachteil:

- Die Trennung von Struktur und Präsentation ist damit hinüber.
- Inline-Styles können autorensseitig nur sehr schwer, nämlich ausschließlich mit **!important**, überschrieben werden

Browserunterstützung

Inline-Styles werden von allen modernen Browsern unterstützt.

Einbindung per *STYLE* – Element:

Das style-Element wird innerhalb des Dokumentkopfs head notiert.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>CSS-Einbindung per style-Element</title>

    <style type="text/css">
      p { color: silver; font-weight: bold; }
    </style>
  </head>

  <body>
    <p>Erster Absatz</p>
    <p>Zweiter Absatz</p>
  </body>
</html>
```

style-Elemente werden dazu verwendet, CSS zu notieren, das dokumentweit gelten soll, in anderen Dokumenten jedoch nicht verwendet wird.

Browserunterstützung

Inline-Styles werden von allen modernen Browsern unterstützt.

Einbindung per *LINK* – Element:

Dies ist die beste Methode. Das Element link verweist auf ein externes Stylesheet.

Vorteil:

- Webautoren können Stylesheets über viele Dokumente (und Websites) hinweg verwenden.
- Webautoren können das Stylesheet auswechseln und dadurch das Layout einer Website ändern, ohne Änderungen an den XHTML-Dokumenten vornehmen zu müssen.

```
<link rel="stylesheet" type="text/css" href="/stylesheets/default.css" media="screen" />
```

Das **Attribut rel** lässt alle Browser erkennen, dass es ein Stylesheet gibt, das auf das Dokument anzuwenden ist.

Das **Attribut media** gibt an, für welche Ausgabemedien das referenzierte Stylesheet bestimmt ist. Es kann eine einzelne Medienbeschreibung sein oder eine durch Kommata getrennte Liste.

Das **Attribut href** teilt dem Browser über einen URI mit, wo er das Stylesheet finden kann.

Browserunterstützung

Inline-Styles werden von allen modernen Browsern unterstützt.

Ausgabemedien:

all	Für alle Geräte.
aural	Für Sprachbrowser, die den Inhalt der Webseite vorlesen.
braille	Für Ausgabegeräte mit sogenannter Braillezeile.
embossed	Für „Brailledrucker“ Der Text der Website wird dabei gestanzt und ist dann für blinde Menschen tastbar.
handheld	Für kleine Handcomputer wie Palms oder Mobiltelefone
print	Für die Ausgabe durch Drucker.
projection	Für Projektoren und ähnliche Geräte.
screen	Für die Ausgabe auf dem Bildschirm, also auf einem Computermonitor.
tty	Für die Ausgabe auf nicht grafischen Ausgabemedien mit Festbreitenschrift, wie zum Beispiel Terminals oder auch Textbrowser wie Lynx.
tv	Für die Ausgabe auf Fernsehern oder ähnlichen Geräten.

Regeln:

Ein Stylesheet besteht im Wesentlichen aus einer Menge von Regeln.

Regeln bestehen aus mindestens einem **Selektor** und einem Paar geschweiften Klammern.

Deklarationen bestehen aus einer **Eigenschaft**, gefolgt von einem Doppelpunkt, gefolgt von einem *Wert*. Vor und hinter diesen Bestandteilen kann sich Leerraum befinden. Deklarationen werden durch Semikola voneinander getrennt.

Teilen sich mehrere Selektoren denselben Deklarationsblock, können sie durch Kommata getrennt nacheinander aufgeführt werden.

```
p { margin: 0; }
```

```
h1, h2, h3, h4, h5, h6 {  
  color: #996600;  
  margin: 2px 0px 1px 0px;  
}
```

Regel	p { margin: 0; }
Selektor	p { margin: 0; }
Deklarationsblock	p { margin: 0; }
Deklaration	p { margin: 0; }
Eigenschaft	p { margin: 0; }
Wert	p { margin: 0; }

SELEKTOR:

Typselektor:

Der Typselektor spricht Elemente anhand des Elementnamens an.

```
p { margin: 0; }
```

Browserunterstützung

Typselektoren werden von allen modernen Browsern erkannt.

Klassen-Selektor:

Das Universalattribut **class** kann für fast alle Elemente vergeben werden und wird mit CSS über den Klassenselektor angesprochen, indem man den Klassennamen mit einem führenden Punkt ohne Leerzeichen direkt hinter den einfachen Selektor setzt.

```
p.hinweis { font-weight:bold; }
```

```
<p class="hinweis">Dies ist ein Absatz</p>
```

oder

```
.hinweis { font-weight:bold; }
```

```
<span class="hinweis">Dies ist kein Absatz</span>
```

ID-Selektor:

Das Universalattribut **id** hat in XHTML-Dokumenten eine besondere Bedeutung, weil sein Wert pro Dokument nur ein Mal auftreten darf und Elemente mit diesem Attribut als Zielanker eines Links dienen können. Aus der zweiten Eigenschaft ergibt sich auch die Notation des ID-Selektors in CSS: Es wird eine Raute (#) vorangestellt:

```
p#hinweis { font-weight:bold; }
```

```
<p id="hinweis">Dies ist ein Absatz</p>
```

oder

```
#hinweis { font-weight:bold; }
```

```
<span id="hinweis">Dies ist kein Absatz</span>
```

Pseudoklassen:

Die Notation beginnt immer mit genau einem Doppelpunkt (:).

a:link	{ color:green; text-decoration:none; }	(Link ist noch nicht besucht worden)
a:visited	{ color:red; text-decoration:none; }	(Link wurde bereits besucht)
a:hover	{ color:silver; text-decoration:none; }	(Link kann eine Rollover Effect haben)
a:focus	{ color:black; text-decoration:none; }	(Link kann einen focus haben)
a:active	{ color:yellow; text-decoration:none; }	(Link kann aktiviert sein)

WERT:

In CSS können Sie Farben auf zwei Arten angeben: als RGB-Farbwert oder über ein Schlüsselwort.

```
color: #FFF;  
color: #FFFFFF;  
color: rgb(255,255,255);  
color: rgb(100%, 100%, 100%);  
color: white;
```

black	Schwarz	#000000	rgb(0, 0, 0)
gray	Grau	#808080	rgb(128, 128, 128)
silver	Silber	#C0C0C0	rgb(192, 192, 192)
white	Weiß	#FFFFFF	rgb(255, 255, 255)
purple	Violett	#800080	rgb(128, 0, 128)
fuchsia	Fuchsia	#FF00FF	rgb(255, 0, 255)
maroon	Kastanienbraun	#800000	rgb(128, 0, 0)
red	Rot	#FF0000	rgb(255, 0, 0)
olive	Olivgrün	#808000	rgb(128, 128, 0)
yellow	Gelb	#FFFF00	rgb(255, 255, 0)
green	Grün	#008000	rgb(0, 128, 0)
lime	Limonengelb	#00FF00	rgb(0, 255, 0)
navy	Navygrün	#000080	rgb(0, 0, 128)
blue	Blau	#0000FF	rgb(0, 0, 255)
teal	Blaugrün	#008080	rgb(0, 128, 128)
aqua	Aquamarinblau	#00FFFF	rgb(0, 255, 255)
orange	Orange	#FFA500	rgb(255, 165, 0)

Geviertbreite (em):

```
h1 { line-height: 1.5em; }
```

Pixel (px):

```
h1 { line-height: 15px; }
```

color:

Die Eigenschaft color bestimmt die Vordergrundfarbe eines Elements, die als Schriftfarbe verwendet wird.

```
p.fehler {  
    color: red;  
    background-color: white;  
}
```

Ausrichtung und Elementfluss:

Positionierung:

Position:

Legt fest, auf welche Art und Weise die Position einer Box berechnet wird. (relative, absolute)

Absolute Positionierung: Angaben left, top, right, bottom.

Absolut positionierte Elemente sind außerhalb des Dokumentenfluß, sie liegen über den anderen Elementen und beeinflussen das Layout nicht. Die Absolute Position wird relativ zu den Rändern des Elternelements berechnet, wenn dieses positioniert ist.

```
#box1 { position: absolute; top: 50px; left: 70px; }
```

Relative Positionierung: Angaben left, top, right, bottom.

Relativ positionierte Elemente sind normal im Dokumentenfluß. Mit left, top, right, bottom wird das Element verschoben. Die nachfolgenden Elemente verhalten sich, als ob das Element nicht verschoben wurde.

```
#box1 { position: relative; top: 50px; left: 70px; }
```

Float:

Boxen können an bestimmten Stellen aus dem normalen Fluss herausgelöst und ganz links oder ganz rechts am Rand des umschließenden Blocks platziert werden.

(left, right)

```
#box1 {float: left; }  
#box2 {float: right; margin: 20px; }
```

clear:

Der Elementfluss um floatierte Boxen kann unterbrochen werden.

(left, right, both)

Das CSS-Boxmodell:

Jedes Element erzeugt eine rechteckige Box, die das Element umgibt und sich aus weiteren Boxen zusammensetzt: einer **Contentbox** (zum Beispiel ein Text oder ein Bild), einer **Paddingbox** (Polsterung), einer **Borderbox** (Rahmen) und einer **Marginbox** (Randabstand). Jeder dieser Bereiche kann in oben, rechts, unten und links aufgeteilt werden

padding:

```
div#inhalt {padding-top: 20px; padding-right: 20px; padding-bottom: 20px; padding-left: 20px;}
```

```
div#inhalt{padding: 20px; }
```

```
div#inhalt{padding: 20px 20px 20px 20px; }
```

Margin:

```
div#inhalt {margin-top: 20px; margin-right: 20px; margin-bottom: 20px; margin-left: 20px;}
```

```
div#inhalt{margin: 20px; }
```

```
div#inhalt{margin: 20px 20px 20px 20px; }
```

border-style:

(none, dotted, dashed, solid, double, groove ...)

```
div#inhalt {border-style: solid; }
```

border-color:

```
div#inhalt {border-color: #ff0000; }
```

border:

```
div#inhalt {border: 1px solid #ff0000; }
```

background-color:

Die Eigenschaft background-color bestimmt die Hintergrundfarbe eines Elements.

```
p.fehler {  
    color: red;  
    background-color: white;  
}
```

background-image:

Relative Verweise gehen vom Stylesheet aus, nicht von dem Dokument, das das Stylesheet einbindet.

```
body {  
    background-image: url(images/bg.png);  
}
```

background-repeat

Die Eigenschaft background-repeat bestimmt, ob und in welche Richtung das Hintergrundbild eines Elements wiederholt werden soll.

repeat-x (Das Hintergrundbild wird nur in x-Richtung wiederholt.)

repeat-y (Das Hintergrundbild wird nur in y-Richtung wiederholt.)

no-repeat (Das Hintergrundbild wird nicht wiederholt.)

```
body { background-repeat: repeat-x; }
```

background-position

Die Eigenschaft `background-position` bestimmt die Position eines Hintergrundbildes. Vom oberen linken Eck ausgehend.

```
body {  
    color: black;  
    background-color: white;  
    background-image: url(bg.gif);  
    background-repeat: repeat-x;  
    background-position: 100px 200px;  
}
```

background:

Fasst die Einstellung der einzelnen Hintergrundeigenschaften zusammen.

```
body {  
    color: black;  
    background: white url(bg.gif) repeat-x 100px 200px;  
}
```

font-family:

Gibt die Schriftfamilie (Schriftart) an, die zur Darstellung verwendet werden soll.

```
body {  
    font-family: Verdana, Arial, Helvetica, sans-serif;  
}
```

font-style:

Gibt an, ob eine normale oder eine schräg laufende Schriftauszeichnungsart gewählt wird.
(normal, italic)

```
body {  
    font-style: italic;  
}
```

font-weight:

Gibt die Gewichtung (Schriftdicke) der verwendeten Schrift an.
(normal, bold, 100, 200, 300, 400, 500, 600, 700, 800, 900)

normal = 400
bold = 700

```
body {  
    font-weight: bold;  
}
```

font-size:

Gibt die Schriftgröße eines Elements an.

```
body {  
    font-size: 12px;  
}
```

font:

Fasst die Einstellungen der einzelnen Schrifteigenschaften sowie die Eigenschaft line-height zur Bestimmung der Zeilenhöhe zusammen.

```
body {  
    font-size: 12px;  
    line-height: 1.3em;  
    font-family: Verdana, Arial, Helvetica, sans-serif;  
}
```

```
body {  
    font: 12px/1.3em Verdana, Arial, Helvetica, sans-serif;  
}
```