

# 33 CSS in HTML einbinden

Damit CSS auf HTML wirken kann, muss eine Verknüpfung hergestellt werden. Dafür stehen verschiedene Möglichkeiten zur Verfügung, die in diesem Kapitel beschrieben sind.

HTML<sub>5</sub> vereinfacht das Einbinden von CSS. Was Sie bei der Einbindung für alte HTML-Standards berücksichtigen müssen, können Sie in den Kapiteln 26.3 und 26.4 lesen.

In den ersten beiden Kapiteln sind Inline-Styles und Styles im head des HTML beschrieben. Die Philosophie von CSS basiert auf der Trennung von Struktur und Darstellung. Von daher sollten CSS und HTML möglichst getrennt werden, was das Auslagern in separate Dateien bedeutet. Dies ist ab dem Kapitel 33.3 beschrieben und sollte die Standardeinbindung sein.

Die Grundlage dafür, dass man CSS überhaupt einbinden kann, liefert das Kaskaden-Konzept von CSS (siehe Kapitel 37.1). Alle HTML-Elemente werden durch das Browser-Stylesheet mit einem Minimal-CSS versehen und können dann durch eigenes CSS im Autoren-Stylesheet erweitert oder auch überschrieben werden.

## 33.1 Inline-Styles über das HTML-Attribut style

CSS-Regeln können direkt im HTML-Code definiert werden. Dazu dient das HTML-Universalattribut `style`, welches in alle HTML-Elemente innerhalb des `body`-Tags eingefügt werden kann:



*Inline-Styles haben die höchste Spezifität und können nur über !important von externen Styles überschrieben werden. Mehr dazu finden Sie in Kapitel 37.2.2.*

```
<h1 style="border: 2px dotted #333333;">
  HTML5/CSS3 Workshop</h1>
```

Für Inline-Styles gelten die gleichen Regeln wie für ausgelagerte Styles: Einzelne Werte werden durch Leerzeichen, mehrere Deklarationen

durch Semikolons voneinander getrennt. Diese sind in Kapitel 31 beschrieben. Das Beispiel eines Inline-Styles mit zwei Deklarationen könnte so aussehen:

```
<h2 style="border: 1px solid #999999; color: #cc0000;">
  Eine Einführung in Webstandards</h2>
```

Inline-Styles stellen allerdings eine archaische Einbindung von CSS dar und erinnern an direkte Layoutanweisungen über HTML-Attribute wie `align` oder `hspace`. Dieses Unterkapitel ist von daher nur der Vollständigkeit halber hier zu finden.

Inline-Styles zeugen nicht von gutem Stil und sollten nur im Ausnahmefall eingesetzt werden – zum Beispiel dann, wenn der Chef eine schnelle Lösung will.

### 33.2 Styles im head per HTML-Element style

Der erste kleine Schritt in Richtung Trennung von Layout und Inhalt findet über das HTML-Element `style` im head der HTML-Datei statt.

```
<style>
  h1 {
    color: red;
  }
</style>
```

Für die korrekte Einbindung ist für alte HTML-Standards das Attribut `type="text/css"` notwendig:

```
<style type="text/css">
  <!--
  h1 {
    color: red;
  }
  -->
</style>
```

Wie Sie im zweiten Beispiel sehen, wird die CSS-Regel innerhalb des `style`-Elements zusätzlich von HTML-Kommentarzeichen umschlossen. Dies hat historische Gründe, da früher Browser, die CSS nicht kannten, die CSS-Anweisungen auf der Webseite angezeigt haben. Darauf sei der Vollständigkeit halber hingewiesen. Sie können die Kommentarzeichen allerdings getrost weglassen, da die Zeit alter Browser – und damit meine ich richtig alte – längst vorbei sein sollte.

Ähnlich wie Inline-Styles sprechen Styles im Dokumentenkopf nicht für sonderlich guten Stil. Allerdings ist ihr Einsatz denkbar, wenn für nur eine einzige Seite im kompletten Webauftritt spezielle Styles benötigt werden und man ein ausgelagertes Style nicht unnötig aufblähen möchte.

Allerdings wäre hier der »Königsweg«, diese speziellen Styles in einer zweiten CSS-Datei auszulagern. Damit würde das Hauptstyle nicht unnötig groß, hätte allerdings den Vorteil, dass die CSS-Datei nur einmal geladen und dann im Browser-Cache vorgehalten wird – und damit Ladezeit spart, wenn der Surfer dieselbe Seite mehrfach besucht.

### 33.3 Externe Styles über das link-Tag einbinden

Die Einbindung ist ausschließlich im head der HTML-Datei zulässig und wird über das HTML-Tag `link` realisiert. Die Einbindung einer ausgelagerten CSS-Datei sieht beispielsweise so aus:

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



Mehr zum Doctype: Kapitel 25

Die korrekte Einbindung hängt vom Doctype und damit vom HTML-Standard ab, der für das HTML-Dokument definiert ist. Was Sie bei anderen HTML-Standards berücksichtigen müssen, lesen Sie in Kapitel 26.3.1.

### 33.4 Styles über die @import-Regel einbinden

Die Einbindung über das `link`-Tag ist die in HTML vorgesehene Möglichkeit. Eine andere Art der Einbindung kann über die `@`-Regel (At-Regel genannt) realisiert werden, die CSS unabhängig von HTML bietet.

Dazu wird das in Kapitel 26.4 beschriebene HTML-Element `style` eingesetzt, welches einen CSS-Bereich im HTML-Code definiert. CSS sieht zum Erweitern einer CSS-Datei die `@import`-Regel vor. Darüber kann innerhalb einer CSS-Datei eine weitere CSS-Datei verlinkt und damit der Style für die komplette Webseite erweitert werden.

```
<style>
  @import 'css/layout.css';
</style>
```

Diese Regel kann auch dazu verwendet werden, in einen CSS-Bereich innerhalb einer HTML-Datei weitere CSS-Dateien »einzuhängen«, sprich zu importieren. Dies ist eine zweite Möglichkeit, um im head des HTML eine externe CSS-Datei einzubinden.

Der große Vorteil der `@import`-Regel ist jedoch, dass Sie innerhalb von CSS-Dateien weitere Dateien verlinken können. Damit ist es Ihnen möglich, die Schnittstelle zum HTML auf eine einzige zu beschränken, sind aber auf CSS-Seite völlig frei, falls Erweiterungen notwendig werden. Vor allem schaffen Sie damit auch eine eindeutige Trennung von HTML-Struktur und CSS-Design.

Wichtig ist – unabhängig davon, ob Sie die `@import`-Regel in HTML- oder in CSS-Dateien verwenden –, dass diese stets am Anfang steht. Das heißt: Innerhalb des `style`-Tags muss die Regel vor allen anderen CSS-Regeln stehen, in CSS-Dateien müssen `@import`-Regeln immer am Anfang positioniert sein.

Generell können Sie mehrere `@import`-Regeln nacheinander verwenden, jede einzelne wird hinter dem Namen der referenzierten CSS-Datei mit einem Semikolon abgeschlossen.

*`@import` muss stets vor allen anderen CSS-Regeln stehen.*

### 33.5 @media-Regel / Media Queries

Wird CSS über das `link`-Tag eingebunden, kann das Attribut `media` vergeben werden. Über die Parameter (beispielsweise `screen` oder `print`) wird CSS für eine bestimmte Art der Ausgabe definiert. Dies ist die Möglichkeit, die HTML vorsieht. Über die `@media`-Regel kann diese »Weiche« im CSS-Code selber konfiguriert werden:

```
body {
  color: #222222;
}
@media screen, projection {
  body {
    font-family: 'Comic Sans MS', serif;
  }
}
@media print {
  body {
    font-family: Arial, sans-serif;
  }
}
```

Innerhalb einer CSS-Datei ist es so möglich, Bereiche für eine bestimmte Ausgabe zu definieren. Damit sind Sie unabhängig vom HTML und können die Einbindung ohne den `media`-Attributwert vornehmen.

Die Regel wird eingeleitet mit einem `@media`, daran anschließend wird das Schlüsselwort für den Medientyp (mit einem Leerzeichen getrennt) angegeben. Mehrere Medientypen können definiert werden und werden jeweils mit einem `,` (Komma) voneinander getrennt. Wie jede andere CSS-Regel wird die folgende Deklaration mit einem Paar geschweiften Klammern umschlossen.

Im Gegensatz zur `@import`-Regel muss die `@media`-Regel nicht am Anfang einer Datei stehen. Es können beliebig viele Regeln in einer Datei stehen und mehrere auch denselben Medientyp beinhalten.

CSS, das außerhalb der `@media`-Regeln definiert ist (wie im Beispiel des `body`-Elementsektors mit der `color`-Angabe), wird von allen Ausgabegeräten verwendet.

Eine klassische Verwendung in der Praxis ist der Einsatz von `@media print { ... }`. Im Style wird damit ein Bereich definiert, der ausschließlich dem Ausdruck dient. Navigationsblöcke, die im Ausdruck nur selten Sinn machen, oder Werbeblöcke, die nur unnötig Papier verschwenden, werden ausgeblendet. Oder es wird die Festlegung einer anderen Schrift für den Ausdruck definiert, da die Lesbarkeit im Druck von anderen Kriterien bestimmt wird als auf dem Bildschirm.

### 33.5.1 Media Queries (Eigenschaftsspezifische Stylesheets)



Media Queries im praktischem Einsatz: Kapitel 15

CSS3 hat die `@media`-Regel ergänzt und betitelt das Modul jetzt als »Media Queries«, was sicher zu den wichtigsten Erweiterungen des neuen Standards zählt. Außerdem gehört es zu jenen Modulen, die am weitesten entwickelt sind – Seit dem 19. Juni 2012 sind Media Queries eine offizielle Empfehlung des W3C und gelten damit als verbindlich.

Media Queries macht insbesondere aus, dass nicht nur bestimmte Medientypen unterschieden, sondern weitere Merkmale des Bildschirms erfasst und davon abhängig unterschiedliche CSS-Regeln definiert werden können.

```
#inhalt {
    float: left;
}
@media screen and (max-width: 800px){
    #inhalt {
        float: none;
    }
}
```