

Cascading Stylesheets (CSS)

Cascading Stylesheets (CSS) sind eine Sprache für Stilvorlagen zur Gestaltung von strukturierten Dokumenten.

So kann man mit Hilfe von HTML die inhaltliche Untergliederung vornehmen und mit CSS die Gestaltung des Layouts. Das hat den Vorteil,

- dass mit wenigen Änderungen innerhalb der CSS weitläufig sichtbare Änderungen an der Webseite vorgenommen werden können,
- CSS auf mehrere Dokumente angewendet werden können, um ein gleichmäßiges Gesamtbild der Webseite zu schaffen,
- vom Inhalt unabhängige Layouts für diverse Geräte (PC, Smartphone u. a.) in einer Datei geschaffen werden können.

CSS im Dokument

CSS können direkt in dem entsprechenden Dokument integriert werden, wenn sie nur für dieses Dokument benötigt werden.

Dafür wird der Tag **style** benötigt mit dem Attribut **type** und dem Inhalt **text/css**.

Der Start-Tag sieht folgendermaßen aus:

```
<style type="text/css">
```

...[INHALT]...

Nach dem Inhalt kommt der Schluss Tag:

```
</style>
```

CSS als eigenständiges Dokument

Wenn sie die Stilvorlagen für mehrere Seiten nutzen möchten, empfiehlt sich eine Auslagerung der CSS in eine separate Datei. die Dateiendung hierfür ist *.css, diese Datei wird mittels eines **link**-Tags mit der HTML-Seite verbunden, dieser Tag steht im **head**-Bereich.

Beispiel für die Integration einer test.css - Datei:

```
<head>  
  <title>Name des Dokuments</title>  
  <link href=test.css" rel="stylesheet" type="text/css" />  
</head>
```

CSS Elemente

Es gibt verschiedene Möglichkeiten zur CSS-Definition, so unter anderem:

- die Definition von HTML-Tags
- die Definition von Klassen oder
- die Definition von ID's.

Ein CSS-Element wird grundsätzlich nach folgendem Schema definiert:

```
name {attribut:wert;.. ..attribut:wert;}
```

Wichtig ist die strikte Benutzung der geschweiften Klammer **{** und am Ende **}**, weiterhin nach dem Attribut die Nutzung des Doppelpunktes und nach jedem Wert - auch dem letzten - ein Semikolon.

Die Schreibweise kann nebeneinander sein, der Übersichtlichkeit halber sollte man sich aber angewöhnen, die einzelnen Attribute untereinander zu schreiben.

Beispiel:

```
body {  
    background:#fff;  
    color: #222;  
    font-family:Arial, Helvetica, FreeSans, sans-serif;  
    font-size:13px;  
}
```

Das obere Beispiel zeigt die CSS-Definition für den HTML-Tag **body**. So kann für jeden Tag genau festgelegt werden, wie der Inhalt aussehen soll.

Individueller gestalten kann man seine Webseite mit Klassen und ID's.

Eine CSS-Definition für Klassen (**class**) kann für mehrere verschiedene Elemente vorgenommen werden. Dazu im Vergleich wird die Definition für ID's (**id**) nur für ein ganz konkretes Element genutzt.

Ein Beispiel für die Nutzung von Klassen sieht man direkt auf der Webseite

<http://www.ralf-rabe.de>

→ **Schulung** → **HTML Basiswissen** → **Cascading Stylesheets**

Ich habe für die farbliche Darstellung von Quelltext eine **class**-Definition genutzt.

Beispiel:

```
<head>
```

Im Quelltext steht:

```
<p><b class="progblau">&lt;head&gt;</b></p>
```

Die Definition für die Klasse progblau in der CSS-Datei lautet

```
.progblau{
    color: #039;
    font-family:"Courier New", Courier, monospace;
    font-size:14px;
}
```

Ein Beispiel für die Nutzung von ID's ist die Verwendung einer CSS-Definition für die genaue Platzierung spezieller Objekte, z. B. einer Navigationsleiste.

Beispiel:

Der Container (div) für die Navigation der Seite bekommt eine eindeutige ID:

```
<div id="navigation">... [Inhalt] ...</div>
```

Die Definition für die ID navigation in der CSS-Datei lautet

```
#navigation {
    position:absolute;
    top:119px;
    left:12px;
    width:940px;
    overflow:hidden;
    z-index:100;
}
```



WICHTIG:

Bei der Definition von Klassen und ID's kann eine eigene Bezeichnung gewählt werden.

- Bei den Klassen kommt ein Punkt (.) vor die Bezeichnung.
- Bei den ID's kommt eine Raute (#) vor die Bezeichnung.