

Cascading Stylesheets

Einige Texte und Grafiken in diesem Skript sind aus dem SELFHTML-Wiki

<https://wiki.selfhtml.org/wiki>

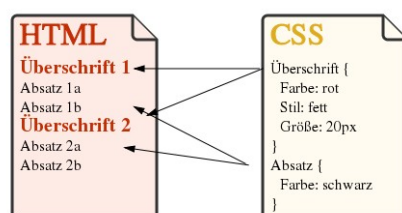
Trennung von Inhalt und Design

Der „große Sinn von CSS“ besteht in der Trennung von Inhalt und Design.

Es ist beinahe unmöglich, einen über die Zeit gewachsenen Internetauftritt umzugestalten - man müsste dutzende oder mehr Einzeldateien umschreiben und im Gewusel der Elemente wird man sich schnell verlieren. Letztlich dauert die Änderung beinahe länger als eine Neuerstellung.

Aus diesem Grund trennt man die Inhalte (Mit HTML strukturierte Texte und Bilder) von der Gestaltung (Stylesheets). So kann man, ohne den Inhalt zu verändern, die Gestaltung ohne Probleme anpassen. Auch wenn die Website aus einzelnen HTML-Dateien besteht.

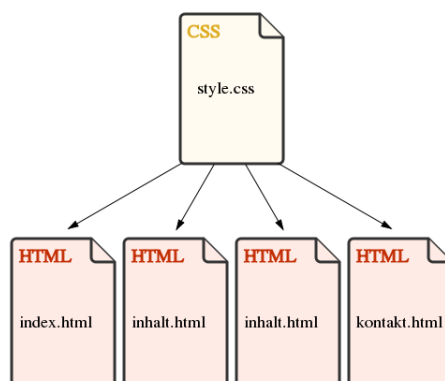
CSS erlaubt es, Stile, Farben und Formen zu definieren, beispielsweise für alle Überschriften, oder für alle Textabsätze mit einem bestimmten Klassennamen, oder für Text, der innerhalb einer Tabellenzelle vorkommt



Zentrale Formate für mehrere HTML-Dateien

Zentrale Formate können sich auf eine HTML-Datei beziehen, aber auch in eine externe Style-Datei ausgelagert werden, die man in beliebig viele Seiten einbinden kann. So werden einheitliche Formatvorgaben über eine Website möglich, und der HTML-Code wird von unnötigem Ballast befreit. Spätere Änderungen am Design können so leicht durchgeführt werden.

In der folgenden Abbildung dient eine einzige CSS-Datei beispielsweise vier HTML-Dateien als Formatvorlage:



CSS-Styles in HTML einbinden

Einbinden im HTML-Kopf

Für jedes HTML-Dokument können Formatierungsanweisungen im Kopf der HTML-Datei eingefügt werden. Die Formatierungsanweisungen müssen zwischen die Tags `<style type="text/css">` und `</style>` eingefügt werden.

Beispiel:

```
<head>
<title>Zentrale Formate</title>
<style type="text/css">
h1 { color:red; font-size:48px; }
</style>
</head>
```

Die im HTML-Kopf eingefügten Formatierungsanweisungen sind für die gesamte HTML-Datei gültig.

Einbinden von Stylesheet-Dateien

Die Formatierungsanweisungen können auch in einer eigenen Datei (Endung .css) gespeichert werden und über das link-Tag in HTML-Dateien eingefügt werden. So können Sie die Formatierung für mehrere HTML-Dokumente verwenden.

Beispiel:

```
<head>
<title>Das link-Element &uuml;r Stylesheets</title>
<link rel="stylesheet" type="text/css" href="formate.css">
</head>
```

In den meisten Fällen ist das Einbinden von Stylesheet-Dateien die beste Lösung bei der Entwicklung von modernen Webseiten.

Direkte Formatierungen

Jedem HTML-Element kann auch direkt ein Format zugewiesen werden. Das sollte nur in Ausnahmefällen verwendet werden.

Beispiel:

```
<h1 style="text-align: center; text-decoration: underline;">
```

Aufbau einer CSS Formatierungsanweisung

Eine CSS Anweisung besteht immer aus einem Selektor, der bestimmt welche HTML-Elemente formatiert werden sollen, und einer Liste von Eigenschaften, denen Werte zugewiesen werden. Eine Übersicht über alle möglichen Eigenschaften finden Sie in selfhtml (siehe oben).

```
Selector { Eigenschaft1 : Wert1;  
Eigenschaft2 : Wert2;  
}
```

Beispiel:

```
h1 { color:red;  
font-size:48px;  
}
```

CSS Selektoren

Damit Formateigenschaften auf ein Element angewendet werden können, muss definiert werden, welche Elemente angesprochen werden sollen. Dies geschieht über Selektoren. Als Selektoren bezeichnet man die Teile einer CSS-Regel, die vor dem Abschnitt aus geschweiften Klammern stehen. Voneinander unabhängige Selektoren, denen dieselben Eigenschaften zugewiesen werden, können mit Kommata getrennt werden.

Eine genaue Beschreibung aller Möglichkeiten für Selektoren finden Sie im Selfhtml-Wiki:

<https://wiki.selfhtml.org/wiki/CSS/Selektoren>

Beispiel:

```
einfacher Selektor { CSS-Eigenschaften }  
erster Selektor, zweiter Selektor { CSS-Eigenschaften }
```

CSS kennt verschiedene Selektor-typen. Die wichtigsten sind:

Universalselektor: *	Formatiert alle HTML-Elemente
Typselektoren: tagname	Formatiert bestimmte HTML-Tags Beispiel: <pre>h1: {font-size: 20px;}</pre>
Klassenselektoren: .class	Formatiert alle Elemente mit der zugewiesenen Klasse (Attribut class in HTML zugewiesen)
ID-Selektoren: #id	Formatiert nur das Element mit der zugewiesenen id (Attribut id in HTML zugewiesen)

Achtung: Es darf innerhalb eines Dokuments mehrmals die gleiche Klasse zugewiesen werden aber jede id nur einmal!