

# CSS Einführung und Spickzettel

CSS steht für **Cascading Stylesheets**.

**style** bedeutet „Stil“, im Kontext der Gestaltung von digitalen Produkten meint man damit meist das Aussehen von Elementen.

**sheet** bedeutet „Blatt“ oder „Bogen“ – Ein Stylesheet ist also eine „Blatt voller Stilanweisungen“.

**cascading** kommt von „Kaskade“, das ist meist ein Wasserfall, der über mehrere Stufen herabfließt. Im übertragenen Sinn ist bei CSS ein Prinzip gemeint, nach dem *stufenweise* festgelegt wird, welche CSS-Regeln für welche Elemente gelten. Der CSS-Mitautor Håkon Wium Lie schreibt [auf seiner Website](#)

*cascading:*

*The process of combining several style sheets and resolving conflicts between them.*

Auf deutsch also etwa: Die Methode, wie verschiedene Stylesheets kombiniert werden und wie Konflikte zwischen ihnen aufgelöst werden. Diese Regeln sind durchaus komplex und werden im Artikel [The Cascade](#) näher erklärt.

## Erklärvideo: Cascading Stylesheets



## Prinzip von CSS

Modernem HTML und CSS liegt das Prinzip zugrunde, dass man Inhalt und Gestaltung in zwei verschiedene Bereiche trennt.

- HTML ist strukturierter Inhalt,
- CSS liefert die Gestaltungsinformationen für den Inhalt.

Eine CSS-Datei enthält also die Darstellungsinformationen, mit denen eine verknüpfte HTML-Struktur visuell gestaltet werden soll. Anstatt die Informationen über z.B. Schriftgröße, Farben, Layout etc. mit dem HTML-Inhalt zu mischen, werden all diese Informationen in eine oder wenige CSS-Dateien ausgelagert. So sind sie übersichtlicher und leichter zu pflegen.

Jede HTML-Datei enthält nur Inhalte, der mit **HTML-Elementen** strukturiert ist. Im `<head>`-Bereich der Datei wird eine Verknüpfung zur CSS-Datei hergestellt, von der die HTML-Datei ihre Darstellungsinformationen „holt“ (siehe unten).



## Verschiedene CSS-Dateien für verschiedene Geräte

Durch das Prinzip der Trennung von Inhalt und Gestaltung erlaubt es auch, den gleichen HTML-Inhalt je nach Kontext oder Gerät unterschiedlich zu gestalten. Man kann eine CSS-Datei für die Darstellung der Website auf dem Desktop anbieten, eine weitere für die kleinen Displays von Mobilgeräten und wieder eine andere für die Ausgabe an einen Drucker (wo z.B. Navigationselemente überflüssig sind und daher ausgeblendet werden können).



## Aufbau einer CSS-Regel

Eine CSS-Regel besteht aus dem **Selektor (*selector*)**, einer Liste von **Eigenschaften (*property*)** und deren konkreten **Werten (*value*)**. Wie bei allen „Computer-Sprachen“ muss man die exakte **Schreibweise** bis ins Detail beachten, damit keine Darstellungsfehler passieren. Achte also auf die Position der geschwungenen Klammern `{}`, des Doppelpunkts `:` und des Semikolons `;`



## Arten von Selektoren

Es gibt eine ganze Reihe von CSS-Selektoren. Hier sollen nur die wichtigsten erklärt werden.

Name	Beispiel	Erklärung
Element-Selektor	header	Wählt das jeweilige HTML-Element aus, hier z.B. <code>&lt;header&gt;...&lt;/header &gt;</code>
mehrere Element-Selektoren	h1, h2, h3	Wählt alle genannten HTML-Elemente aus, hier also <code>&lt;h1&gt;...&lt;/h1&gt;</code> , <code>&lt;h2&gt;...&lt;/h2&gt;</code> und <code>&lt;h3&gt;...&lt;/h3&gt;</code>
verschachtelte Element-Selektoren	header h2	Wählt nur die HTML-Elemente aus, die der Verschachtelung entsprechen. Hier also nur die h2-Elemente, die sich innerhalb des Elements header befinden <code>&lt;header&gt;&lt;h2&gt;...&lt;/h2&gt;&lt;/header&gt;</code>
ID-Selektor	#hauptartikel	Wählt das HTML-Element aus, welches das id-Attribut <code>hauptartikel</code> hat <code>&lt;article id="hauptartikel"&gt;...&lt;/article&gt;</code> .  Ein id-Attribut kann nur ein Mal pro HTML-Seite verwendet werden.

Name	Beispiel	Erklärung
class-Selektor	.infobox	Wählt die HTML-Elemente aus, welche das class-Attribut infobox haben.  <aside class="infobox">...</aside>  Im Gegensatz zum id-Attribut kann das class-Attribut mehrfach auf einer HTML-Seite vorkommen, man kann es benutzen, um mehrere Elemente zu einer Gruppe („Klasse“) zusammen zu fassen und gemeinsam zu gestalten.
Pseudo-Klasse-Selektor	:link :visited :hover ...	Besondere Art von Selektor, der ein Element in einem bestimmten Zustand auswählt - meist in Kombination mit dem a-Element verwendet.  a:link gilt für einen nicht besuchten Link a:visited gilt für einen besuchten Link a:hover gilt für einen Link, solange der Mauszeiger darüber steht

Weitere Selektoren erklärt der Artikel [CSS Selectors](#).

## Häufig benutzte CSS Eigenschaften

Wenn man beginnt, mit CSS zu arbeiten, helfen die folgenden Eigenschaften schnell weiter. Wenn man konkrete Ideen hat, die darüber hinaus gehen, sollte man sich mit einer Referenz-Website wie z.B. [W3Schools CSS Properties](#) Stück für Stück weitere Eigenschaften aneignen.

Einige dieser Eigenschaften gibt es als „Zusammenfassungen“, um weniger schreiben zu müssen.

Anstatt z.B. border-width, border-style und border-color jeweils einzeln anzugeben, was viel Tipperei wäre, kann man sie in der Eigenschaft border zusammen fassen.

Die folgenden beiden **Beispiele** sind in ihrer Wirkung identisch:



```

article {
  border-width: 1px;
  border-style: solid;
  border-color: #ccc;
}

article {
  border: 1px solid #ccc;
}

```

Ähnliches gilt z.B. für background (umfasst background-color, background-image etc.),

margin (umfasst margin-top, margin-right etc.), font (umfasst font-family, font-size etc.) und andere.

Eigenschaft	Erklärung
background-color	Hintergrundfarbe eines Elements
background	Zusammenfassung verschiedener background-Eigenschaften
border	Zusammenfassung verschiedener border-Eigenschaften
margin	Zusammenfassung verschiedener margin-Eigenschaften
padding	Zusammenfassung verschiedener padding-Eigenschaften
line-height	Zeilenhöhe eines Textes
text-align	Text-Aurichtung (linksbündig, rechtsbündig, zentriert etc.)
text-decoration	Unterstreichung u.ä.
font-family	Schriftfamilie
font-size	Schriftgröße
font	Zusammenfassung verschiedener Schrift-Eigenschaften

## Grundlegender Aufbau einer CSS-Datei

Eine CSS-Datei hat einen sehr einfachen Aufbau: Sie enthält meist einfach eine Sammlung von CSS-Regeln. Die Reihenfolge der Regeln ist nur in fortgeschrittenen Spezialfällen von Bedeutung (Stichwort: „CSS-Cascade“).

Hier ein Beispiel:

```
body {
  background-color: #cce9d0;
}

header {
  margin: 2em 0;
  padding: 1em;
  background-color: #fff;
}

nav {
  margin: 2px 0px;
  padding: 0.5em;
  border: 1px solid #ccc;
  background-color: #fff;
}

article {
  margin: 2em 0;
  padding: 1em;
  background-color: #fff;
}

#main {
  background-color: #f7f58c;
```

```

}

article section {
    margin: 2em 0;
}

/* dient für Randbemerkungen */
article aside {
    margin: 1em 0;
    border: 1px solid #ccc;
    background-color: #ffc;
}

/* in der Infobox erscheinen weiterführende Links zu einem Artikel */
aside.infobox {
    width: 40%;
    float: right;
    margin-left: 2em;
    padding: 1em;
}

```

## Kommentare

In CSS-Dateien kann man Kommentare einbauen, die vom Browser ignoriert werden. Das ist nützlich, um sich später noch erinnern zu können, warum man bestimmte Dinge so und nicht anders gestaltet hat. Auch wenn man mit anderen zusammenarbeitet, ist es sinnvoll, Kommentare zu hinterlassen. Man schreibt den Kommentar zwischen folgende Zeichenkombination `/* dies ist ein Kommentar */`

## Verbindung von HTML- und CSS-Datei

Die CSS-Datei wird durch ein `link`-Element im `<head>`-Bereich einer HTML-Datei mit dieser verbunden. Einzelnen sieht dieses Element so aus:

```
<link rel="stylesheet" href="meinstile.css">
```

Im Kontext der ganzen HTML-Datei so:

```

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Seitentitel</title>

    <link rel="stylesheet" href="meinstile.css"> <!-- Dieses Element stellt
die Verbindung

```

der HTML-Datei mit der CSS-Datei her. Der Dateiname der CSS-Datei kann beliebig gewählt werden. Man muss nur darauf achten, dass der Pfad zur CSS-Datei korrekt ist. -->

```
</head>
```

```
<body>
```

```
<!-- Hier stehen weitere HTML-Elemente -->
```

```
</body>
```

```
</html>
```