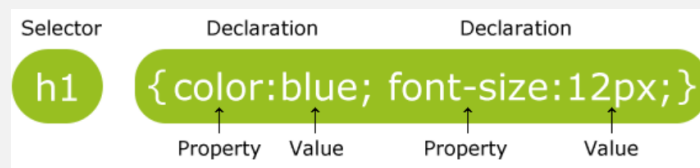


## Syntax und Regelsätze

## Syntax von CSS Regeln

Ein CSS Regelsatz besteht aus

- einem **Selektor**: bestimmt, welches HTML-Tag formatiert werden soll
- einer **Anweisung**: bestimmt die Eigenschaft und den gewünschten ein Wert



## Beispiel eines CSS-Codes

```
1 body {  
2     background-color: lightblue;  
3 }  
4  
5 h1 {  
6     color: white;  
7     text-align: center;  
8 }  
9  
10 p {  
11     font-family: verdana;  
12     font-size: 20px;  
13 }
```

Dieser Code verwendet **Element-Selektoren**. So werden alle Tags gleich formatiert, z.B. alle <h1> Überschriften haben eine weiße Schriftfarbe und sind zentriert.

## Übung 16 Artikel (intern)

Im Ordner 5.2\_Artikel findest du die Datei *cascading\_style\_sheets\_intern.html*.

Verändere die Farb- und Textgestaltung nach den unten angeführten Kriterien. Der Screenshot dient dir als Hilfe, wie die fertige Webseite aussieht.

Verwende für folgende Zuweisungen das **interne Stylesheet**:

(CSS Code im `<style>...</style>` Bereich der HTML-Datei schreiben)

➤ **<body>**

- Schriftarten Calibri, sans-serif (als fall-back)
- Hintergrundfarbe #5b5b5b

➤ **<h1> und <h2> Überschriften**

- Schriftfarbe #e2e0e0
- Textausrichtung zentriert (*center*)

➤ **<h3> Überschriften**

- Schriftfarbe #ffffff
- Hintergrundfarbe #c1bfbf
- Textausrichtung zentriert (*center*)

➤ **<p> Absätze**

- Schriftfarbe #ffffff
- Hintergrundfarbe #c1bfbf
- Textausrichtung Blocksatz (*justify*)

➤ **<a> Hyperlinks**

- Schriftfarbe #ffffff
- Textgestaltung nicht unterstrichen

➤ **Füge folgenden Code ein** (dadurch wird der standardmäßige Abstand auf 0 gesetzt)

```
h3, p {  
    margin: 0;  
}
```



Mit dem Beistrich kann man mehrere Selektoren aneinanderketten.  
Der CSS-Code gilt dann für alle angegebenen Selektoren.

Verwende für folgende Zuweisungen das **Inlineattribut**:

(CSS Code direkt beim HTML Tag mit `style="..."`)

➤ **<p> Absatz im Kapitel „Quelle“**

- Hintergrundfarbe #5b5b5b
- Schriftstil kursiv (*italic*)
- Textausrichtung zentriert (*center*)

## Screenshot

### CSS - Cascading Style Sheets

CSS (Cascading Style Sheets, zu deutsch „Mehrstufige Formatvorlagen“) ist eine Formatierungssprache für HTML-, SVG- und XML-Dokumente.

#### Warum CSS?

##### Trennung von Inhalt und Design

Der „große Sinn von CSS“ besteht in der Trennung von Inhalt und Design. Das hört man oft, stellt sich nichts drunter vor und bastelt dann doch eine besondere Überschrift mit einem inline-style, einen neuen div-container mit einer ganz speziellen id oder Klasse, die später nirgends wieder auftaucht. Das funktioniert auch mehr oder weniger - man wird ohne viel Mühe eine fixe Seite zusammenstellen, die ihren Zweck erfüllt - und vergisst das ganze. Es ist aber beinahe unmöglich, einen so über die Zeit gewachsenen Internetauftritt umzugestalten - man müsste dutzende oder mehr Einzeldateien umschreiben und im Gewusel der Klassen und Elemente wird man sich schnell verlieren. Letztlich dauert die Änderung beinahe länger als die Neuerstellung. Und genau darin liegt die Stärke von CSS: uneingeschränkte Flexibilität, wenn z.B. das Layout nicht mehr zeitgemäß ist oder wenn neue Strukturen, vor allem bei dynamischen Seiten, Änderungen erfordern. Dabei wäre man ohne CSS buchstäblich „verloren im Quelltext“.

##### Layout - am Anfang ohne!

Das erste Gebot von CSS lautet: „am Anfang ohne.“ Soll heißen - man gestaltet die Seite erst komplett ohne CSS, nur die Seitenstrukturierung in Kapitel und Absätze, bei diesen dann die Textauszeichnung von Überschriften und Ähnlichem. Es stellt auch die „allgemeine Gültigkeit“ des Layouts sicher - ist die Grundlage an sich stimmig, kann man die Seite beliebig umgestalten, jeder Fehler schränkt dies wieder ein. Eine solche Webseite sieht nicht gut aus, ist aber uneingeschränkt nutzbar. Die einzelnen Elemente werden vom Browser einfach untereinander auf dem Bildschirm angeordnet, und zwar in der Reihenfolge, in der sie in der HTML-Datei aufgeschrieben sind. Die Elemente selbst werden gemäß den Voreinstellungen des Browsers dargestellt; so ist z.B. festgelegt, dass der Hintergrund des Dokuments weiß oder grau darzustellen ist, dass Überschriften in fett und einer Größe von soundsoviel Punkten darzustellen sind.

##### CSS - Formatsprache für HTML-Elemente

An diesem Punkt setzen die Cascading Stylesheets ein. Es handelt sich dabei um eine unmittelbare Ergänzungssprache, die vorwiegend für HTML (aber auch für andere Dokumente wie SVG) entwickelt wurde. Sie klinkt sich nahtlos in HTML ein und erlaubt das beliebige Formatieren einzelner HTML-Elemente.

#### Quelle

Text entnommen von SELFHTML-Wiki, [https://wiki.selfhtml.org/wiki/CSS/Tutorials/Warum\\_Layouts\\_mit\\_CSS%3F](https://wiki.selfhtml.org/wiki/CSS/Tutorials/Warum_Layouts_mit_CSS%3F)

## Validierung

- Überprüfe deinen **HTML-Code** mit dem W3C HTML Validator!  
<https://validator.w3.org/>

Document checking completed. No errors or warnings to show.

- Überprüfe deinen **CSS-Code** mit dem W3C CSS Validator!  
<https://jigsaw.w3.org/css-validator/>

Gratulation! Es wurden keine Fehler gefunden.

## Übung 17 Artikel (extern)

Im Ordner *5.2\_Artikel* findest du die Datei *cascading\_style\_sheets\_extern.html*.

Führe die Zuweisungen aus Übung 16 noch einmal durch – verwende diesmal aber nicht das interne Stylesheet, sondern das externe Stylesheet (Anweisungen in separater CSS-Datei)!

- Erstelle dazu im gleichen Ordner wie die HTML-Datei eine leere Textdatei. Benenne sie *style.css*.
- Kopiere den Code aus dem internen Stylesheet und füge ihn in die separate CSS-Datei ein.
- Lösche das interne Stylesheet.
- Öffne nun die HTML-Datei → die Webseite muss wieder wie in Übung 16 aussehen!