

# CSS Tutorial (Grundlagen)

von [html-einfach.de](http://html-einfach.de)



**Cascading Style Sheets (CSS)**, ist eine Stylesheet-Sprache, die zur Beschreibung des Aussehens und der Formatierung eines Dokuments geschrieben in [HTML](#) oder XML verwendet wird. Sie ist ein entscheidendes Werkzeug in der Webentwicklung und bietet Entwicklern und Designern die Möglichkeit, das Aussehen einer Website zu kontrollieren und zu ändern.

## 1. Einleitung in unser CSS Tutorial

Im Vergleich zu HTML, das hauptsächlich zur Strukturierung des Inhalts verwendet wird, ermöglicht CSS die Gestaltung von Webseiten, einschließlich Layout, Farben und Schriftarten. Sie ermöglicht auch eine größere Konsistenz über mehrere Seiten einer Website hinweg und kann genutzt werden, um spezifische Styles für verschiedene Geräte und Bildschirmgrößen zu definieren.

Eine HTML-Seite und ihr CSS sind in der Regel getrennt. HTML-Elemente können über CSS-Selektoren gestaltet werden, wobei die CSS-Regeln entweder in externen Dateien, in internen `<style>`-Tags oder direkt auf einzelnen HTML-Elementen mit dem `style`-Attribut definiert werden. In diesem CSS Tutorial wirst du die wichtigsten Methoden kennenlernen und praktisches Wissen sammeln.

---

## 2. Selektoren

### Erklärung des Codebeispiels:

Das was vor der geschweiften Klammer steht wird zu unserem Selektor. Das bedeutet, dass alle zutreffenden [HTML](#) Elemente entsprechend dem Klammerinhalt definiert werden. Zutreffende HTML Elemente sind hier alle Absätze (p-Tags). Der Inhalt der Klammer ist die CSS Deklaration, in diesem Fall werden allen p-Tags eine Farbe zugewiesen. Color ist unsere Eigenschaft, die einen Wert zugewiesen bekommt (Rot).

Daraus ergibt sich folgendes Schema.

### Schema CSS-Selektor:

```
1 SELEKTOR { Eigenschaft: Wert }
```

### Beispiel: Alle Absätze (p-Tag) rote Schrift)

```
1 p { color: red }
```

CSS-Kommentare beginnen mit `/*` und enden mit `*/`. Sie können verwendet werden, um Anmerkungen zu Ihrem Code zu machen oder um bestimmte Teile des Codes vorübergehend auszuschalten, ohne sie zu löschen.

```
1 /* Dies ist ein CSS-Kommentar */
```

Es gibt verschiedene Arten von CSS-Selektoren, die Sie verwenden können, um die Elemente auszuwählen, die Sie gestalten möchten:

- **Element-Selektoren** wählen HTML-Elemente basierend auf dem Elementnamen aus. Zum Beispiel wendet `p {color: red;}` die Farbe Rot auf alle Absatz-Elemente (`<p>`) an.
- **Klassen-Selektoren** wählen Elemente basierend auf der Klasse des Elements aus. Klassenselektoren werden mit einem Punkt (`.`) eingeleitet. Zum Beispiel wendet `.highlight {background-color: yellow;}` eine gelbe Hintergrundfarbe auf alle Elemente mit der Klasse „highlight“ an.
- **ID-Selektoren** wählen ein bestimmtes Element basierend auf seiner ID aus. ID-Selektoren werden mit einer Raute (`#`) eingeleitet. Zum Beispiel wendet `#special {font-size: 20px;}` eine Schriftgröße von 20 Pixel auf das Element mit der ID „special“ an.

Sie können auch Selektoren gruppieren und verschachteln, um mehr Kontrolle über das Styling zu haben und Ihren CSS-Code effizienter zu gestalten. In den nächsten Kapiteln werden wir mehr über diese und andere fortgeschrittene Selektoren und CSS-Techniken lernen.

---

## 3. CSS Grundlagen: Einbinden in HTML

**CSS Code in HTML Datei einbinden:**

```
1 <style type="text/css">
2     <!--
3         p { color: red }
4     -->
5 </style>
```

...und in den `<head>` eingefügt. So könnte eine vollständige HTML-Datei mit CSS aussehen:.

### CSS-Code in HTML Datei:

```
1
2
3 <html>
4   <head>
5     <style type="text/css">
6       <!--
7         p { color: red }
8       -->
9     </style>
10  </head>
11  <body>
12    <h1>So funktioniert</h1>
13    <p>Erster Absatz</p>
14    <p>Zweiter Absatz</p>
15    <p>Dritter Absatz</p>
16  </body>
17 </html>
18
19
20
```

Hier sehen Sie das [Ergebnis](#)

Achte beim Einbinden auf unsere Best Practises [HTML+CSS](#).

## CSS style-Attribut

```
1 <p style="color:blue">Zweiter Absatz</p>
```

Eine Besonderheit ist das direkte Formatieren mit Hilfe des `style-Attributs` (siehe Zeile 12) .

### CSS-Code:

```
1
2
3 <html>
4   <head>
5     <style type="text/css">
6       <!--
7         p { color: red }
8       -->
9     </style>
10  </head>
11  <body>
12    <h1>So funktioniert</h1>
13    <p>Erster Absatz</p>
14    <p style="color:blue">Zweiter Absatz</p>
15    <p>Dritter Absatz</p>
16  </body>
17 </html>
18
19
```

Hier sehen Sie das [Ergebnis](#)

### Erklärung:

Wenn ein Element mit dem `style-Attribut` formatiert wird, wird diese, sehr direkte Methode priorisiert und überschreibt somit gleichbedeutende Eigenschaften die im Head, oder in einem externen Stylesheet definiert wurden. Ich empfehle, eine gesonderte CSS-Datei einzubinden. So ist es möglich, CSS-Änderungen an einer Stelle vorzunehmen, die sich dann auf alle HTML-Dokumente auswirken, in denen die Datei eingebunden ist. Wie das funktioniert erfährst du [hier](#).

---

## 4. Farben und Hintergründe

CSS bietet eine Vielzahl von Möglichkeiten, um Farben und Hintergründe in Ihrer Webseite zu steuern. Die Farben können durch Schlüsselwörter (z.B. „red“), Hexadezimalcodes (z.B. „#ff0000“ für Rot), RGB-Werte (z.B. „rgb(255, 0, 0)“ für Rot) oder HSL-Werte (z.B. „hsl(0, 100%, 50%)“ für Rot) spezifiziert werden.

**CSS Hintergrundfarbe:** Mit der `background-color` Eigenschaft können Sie die Hintergrundfarbe eines Elements bestimmen:

```
1 codebody {
2   background-color: #f0f0f0; /* Hellgrau */
3 }
```

**CSS Hintergrundbild:** Sie können auch Hintergrundbilder mit der `background-image` Eigenschaft hinzufügen:

```
1 body {
2   background-image: url('hintergrund.jpg');
```

```
3 }
```

**CSS Farbverlauf:** Neben der Festlegung von Hintergrundbildern können Sie auch Verläufe verwenden, die von einer Farbe zu einer anderen (oder mehreren anderen) übergehen. Sie können lineare oder radiale Verläufe erstellen:

```
1 div {  
2   background-image: linear-gradient(to right, red , yellow);  
3 }
```

---

## 5. Text-Stile und Schriftarten

**CSS Schriftfarbe:** Mit CSS können Sie die Darstellung von Text auf vielfältige Weise gestalten. Die `color` Eigenschaft bestimmt die Textfarbe:

```
1 p {  
2   color: blue;  
3 }
```

**CSS Schriftart:** Die `font-family` Eigenschaft legt die Schriftart für den Text fest. Sie können eine spezifische Schriftart oder eine allgemeine Schriftfamilie (wie „serif“, „sans-serif“, „monospace“ usw.) angeben:

```
1 p {  
2   font-family: Arial, sans-serif;  
3 }
```

**CSS Textgröße:** Mit der `font-size` Eigenschaft können Sie die Größe des Textes bestimmen:

```
1 p {  
2   font-size: 16px;  
3 }
```

**CSS Schrift:** Es gibt auch viele andere Eigenschaften zur Steuerung von Textstilen, einschließlich `text-align` für die Ausrichtung des Textes, `line-height` für den Zeilenabstand, `text-decoration` für das Hinzufügen von Linien zu Text (wie Unterstreichungen) und `text-transform` für die Umwandlung von Groß- und Kleinbuchstaben.

```
1 p {  
2   text-align: center;  
3   line-height: 1.5;  
4   text-decoration: underline;  
5   text-transform: uppercase;  
6 }
```

In den folgenden Kapiteln in dem CSS Tutorial werden wir mehr über fortgeschrittene Stylings und Techniken in CSS lernen, einschließlich des CSS-Box-Modells, Positionierung und Layouts, Flexbox und Grid, Responsive Design und Animationen.

---

## 6. Box-Modell

Das CSS-Box-Modell ist ein grundlegendes Konzept, das hilft, das Design und das Layout von Elementen auf der Webseite zu kontrollieren. Jedes Element auf einer Webseite kann als Box betrachtet werden, und das Box-Modell beschreibt, wie diese Elemente in Bezug auf Layout und Design behandelt werden.

Jede Box besteht aus vier Teilen: Inhalt, Padding, Border und Margin. Der Inhalt ist das, was Sie innerhalb des Elements sehen, wie Text oder Bilder. Das Padding ist der

Raum zwischen dem Inhalt und dem Border. Der Border ist eine Linie, die um das Padding und den Inhalt herum verläuft. Der Margin ist der Raum um die Border.

Die Eigenschaften `padding`, `border` und `margin` können verwendet werden, um diese Aspekte der Box zu steuern. Sie können Werte in Pixel, Prozent, em, rem oder anderen Einheiten angeben:

```
1 div {  
2   padding: 10px;  
3   border: 2px solid black;  
4   margin: 15px;  
5 }
```

Die `width` und `height` Eigenschaften steuern die Breite und Höhe der Box. Beachten Sie jedoch, dass diese Eigenschaften standardmäßig nur die Breite und Höhe des Inhalts steuern, nicht die des Paddings, Borders oder Margins. Sie können die Eigenschaft `box-sizing` auf `border-box` setzen, um dieses Verhalten zu ändern:

```
1 div {  
2   box-sizing: border-box;  
3   width: 50%;  
4   height: 200px;  
5 }
```

---

## 7. Positionierung und Layout

CSS bietet verschiedene Techniken zur Positionierung und Gestaltung des Layouts Ihrer Webseite. Die `display` Eigenschaft ist eine der wichtigsten für das Layout und das Verhalten von Boxen. Sie kann Werte wie `block`, `inline`, `inline-block`, `none` und andere annehmen:

```
1 div {
2   display: block;
3 }
```

Die `position` Eigenschaft ermöglicht es Ihnen, Elemente auf spezifische Weisen zu positionieren:

- `static` ist die Standardeinstellung und positioniert das Element nach dem normalen Dokumentfluss.
- `relative` positioniert das Element relativ zu seiner normalen Position.
- `absolute` positioniert das Element absolut in Bezug auf den nächstgelegenen positionierten Vorfahren (anstatt in Bezug auf das Dokumentfenster).
- `fixed` positioniert das Element in Bezug auf das Browserfenster.

```
1 div {
2   position: relative;
3   top: 20px;
4   left: 50px;
5 }
```

Mit den `float` und `clear` Eigenschaften können Sie Elemente so gestalten, dass sie auf der linken oder rechten Seite des Containers stehen und der Text um sie herum fließt. Dies kann für Dinge wie Bilder in Text oder mehrspaltige Layouts nützlich sein.

```
1 img {
2   float: right;
3   margin-left: 20px;
4 }
```

Das waren auch schon die wichtigsten CSS Grundlagen. Weiter geht es mit dem [Responsive Design mit HTML & CSS](#) sowie den [Best Practises](#).

