



CSS Cheat Sheet

Einführung in CSS

- **CSS (Cascading Style Sheets):** Verwenden, um das Aussehen und Layout von Webseiten zu gestalten.
- **HTML:** Struktur einer Webseite.
- **CSS:** Look und Feel einer Webseite.

Aufbau einer CSS Regel

CSS:

```
p {  
  color: blue;           /* Textfarbe */  
}
```

- **Selektor:** p (wählt alle <p>-Elemente)
- **Eigenschaften:** color
- **Wert:** blue

Dieser CSS-Code macht den Text aller <p>-Elemente blau.

CSS Einbinden

Inline-Stile

```
<h1 style="color: purple;">Hallo Welt</h1>  
<button style="background-color: lightgreen;">Ich bin ein Button</button>
```

Interne Stile

```
<head>  
  <style>  
    h2 {  
      color: palevioletred;  
    }  
  </style>  
</head>  
<body>  
  <h2>Ich bin ein H2</h2>  
</body>
```

Externe Stile

```
<head>  
  <link rel="stylesheet" type="text/css" href="styles.css">  
</head>
```

styles.css

```
h2 {  
  color: indigo;  
}
```

Farben

Benannte Farben

```
color: red;  
background-color: PapayaWhip;
```

RGB-Farben

```
color: rgb(255, 0, 0); /* Rot */  
background-color: rgb(0, 255, 0); /* Grün */
```

Hexadezimale Farben

```
color: #FF0000; /* Rot */  
background-color: #00FF00; /* Grün */
```

Text Eigenschaften

Textausrichtung

```
h1 {  
    text-align: center;  
}  
p {  
    text-align: right;  
}
```

Schriftstärke

```
h1 {  
    font-weight: normal;  
}  
p {  
    font-weight: bold;  
}
```

Textdekoration

```
h1 {  
    text-decoration: underline;  
}  
p {  
    text-decoration: line-through;  
}  
a {  
    text-decoration: none;  
}
```

Zeilenhöhe

```
p {  
    line-height: 2;  
}
```

Buchstabenabstand

```
h1 {  
    letter-spacing: 5px;  
}  
p {  
    letter-spacing: 2px;  
}
```

Klassen und IDs

Verwendung von Klassen

- Klassen werden verwendet, um mehrere Elemente mit denselben Stilen zu versehen.

HTML

```
<p class="highlight">Dies ist ein hervorgehobener Absatz.</p>
<p class="highlight">Noch ein hervorgehobener Absatz.</p>
```

CSS

```
.highlight {
  color: blue;
  background-color: yellow;
}
```

Verwendung von IDs

- IDs sind eindeutig und sollten nur einmal pro Seite verwendet werden.

HTML

```
<p id="unique">Dies ist ein einzigartiger Absatz.</p>
```

CSS

```
#unique {
  color: white;
  background-color: black;
}
```

Selektoren

Grundlegende Selektoren

```
/* Element-Selektor */
h1 {
  color: green;
}

/* Klassenselektor */
.highlight {
  color: blue;
}

/* ID-Selektor */
#unique {
  color: red;
}
```

Kombinierte Selektoren und Spezifität in CSS

Kombinierte Selektoren

Kombinierte Selektoren sind eine Methode, um gezielter und detaillierter CSS-Regeln auf Elemente anzuwenden. Hier sind einige häufige kombinierte Selektoren:

Nachfahrenselektor

Der Nachfahrenselektor wählt Elemente aus, die **Nachkommen** eines bestimmten Elements sind.

```
/* Selektiert alle <p> innerhalb eines <div> */
div p {
  color: green;
}
```

Kindselektor

Der Kindselektor wählt nur **direkte** Kinder eines bestimmten Elements aus.

```
/* Selektiert alle direkten <p>-Kinder eines <div> */
div > p {
  color: blue;
}
```

Gruppenselektor

Der Gruppenselektor ermöglicht das **Zusammenfassen** mehrerer Selektoren, um dieselben Stile anzuwenden.

```
/* Selektiert alle <h1>, <h2> und <p> Elemente */
h1, h2, p {
    color: red;
}
```

Attributselektor

Der Attributselektor wählt Elemente basierend auf ihren Attributen aus.

```
/* Selektiert alle <input> Elemente mit dem Attribut type="text" */
input[type="text"] {
    background-color: yellow;
}
```

Spezifität in CSS

Die Spezifität bestimmt, welche CSS-Regeln angewendet werden, wenn mehrere Regeln auf dasselbe Element zutreffen. CSS bewertet die Spezifität anhand der folgenden Kriterien:

1. **Inline-Stile** (direkt im HTML-Element): höchste Priorität
2. **ID-Selektoren** (#id)
3. **Klassenselektoren** (.class), Attributselektoren ([attr=value]), Pseudoklassen (:hover, :first-child, etc.)
4. **Element-Selektoren** (div, p, h1, etc.) und Pseudoelemente (::before, ::after)

Beispiel zur Veranschaulichung der Spezifität

HTML

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>CSS Spezifität</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1 id="main-heading" class="header">Willkommen auf meiner Seite</h1>
</body>
</html>
```

CSS (styles.css)

```
/* Element-Selektor */
h1 {
    color: green;
}

/* Klassenselektor */
.header {
    color: blue;
}

/* ID-Selektor */
#main-heading {
    color: red;
}
```

In diesem Beispiel wird der Text des <h1>-Elements rot angezeigt, weil der ID-Selektor (#main-heading) die höchste Spezifität hat.

Beispiel mit kombinierter Spezifität

HTML

```
<!DOCTYPE html>
<html lang="de">
```

```

<head>
  <meta charset="UTF-8">
  <title>CSS Kombinierte Selektoren</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="container">
    <p id="unique" class="text">Dies ist ein einzigartiger Absatz.</p>
    <p class="text">Dies ist ein normaler Absatz.</p>
  </div>
</body>
</html>

```

CSS (styles.css)

```

/* Element-Selektor */
p {
  color: green;
}

/* Klassenselektor */
.text {
  color: blue;
}

/* ID-Selektor */
#unique {
  color: red;
}

```

```

/* Inline-Stil */
<p id="unique" class="text" style="color: black;">Dies ist ein einzigartiger Absatz.</p>

```

Hier ist die Farbe des Textes im <p> mit der ID unique schwarz, weil der Inline-Stil die höchste Spezifität hat. Wenn der Inline-Stil entfernt wird, wäre die Farbe rot, da der ID-Selektor die nächsthöchste Spezifität hat.

Zusammengefasst:

- **Inline-Stile** überschreiben alle anderen.
- **ID-Selektoren** haben eine höhere Spezifität als Klassenselektoren.
- **Klassenselektoren** haben eine höhere Spezifität als Element-Selektoren.

Durch das Verständnis der Spezifität und kombinierter Selektoren kannst du präzise und konfliktfrei Stile auf deine HTML-Elemente anwenden.

Beispiel Zusammenfassung

HTML

```

<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>CSS Beispiel</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1 class="header">Willkommen auf meiner Seite</h1>
  <p id="unique">Dies ist ein einzigartiger Absatz.</p>
  <p class="highlight">Dies ist ein hervorgehobener Absatz.</p>
  <p class="highlight">Noch ein hervorgehobener Absatz.</p>
</body>
</html>

```

CSS (styles.css)

```

/* Element-Selektor */
h1 {
  text-align: center;
}

/* ID-Selektor */

```

```
#unique {
  color: white;
  background-color: black;
}

/* Klassenselektor */
.highlight {
  color: blue;
  background-color: yellow;
}
```

CSS Box-Modell Cheat Sheet:

Das CSS-Box-Modell beschreibt, wie Elemente auf einer Webseite in Boxen modelliert werden, die verschiedene Eigenschaften wie Inhalt, Padding, Border und Margin haben. Hier ist ein Überblick über die Schlüsselkonzepte:

1. Content (Inhalt):

- Der eigentliche Inhalt des Elements.
- Definiert durch die Breite (width) und Höhe (height) des Elements.

2. Padding (Innenabstand):

- Der Raum zwischen dem Inhalt des Elements und seiner Grenze (Border).
- Definiert durch padding-top, padding-right, padding-bottom und padding-left.
- Beispiel: padding: 10px; oder padding: 10px 20px; (oben/unten, links/rechts) oder padding: 10px 20px 30px 40px; (oben, rechts, unten, links).

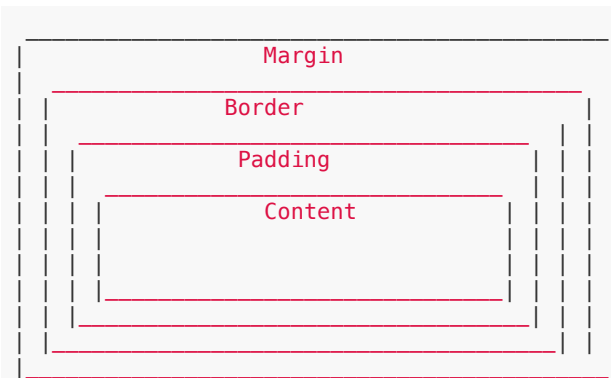
3. Border (Rahmen):

- Eine Linie um den Inhalt und das Padding herum.
- Definiert durch border-width, border-style und border-color.
- Beispiel: border: 1px solid black;.

4. Margin (Außenabstand):

- Der Raum zwischen den Grenzen (Border) des Elements und seinen Nachbarelementen.
- Definiert durch margin-top, margin-right, margin-bottom und margin-left.
- Beispiel: margin: 10px; oder margin: 10px auto; (oben/unten, links/rechts zentriert) oder margin: 10px 20px 30px 40px; (oben, rechts, unten, links).

Box-Modell Diagramm:



Beispiele

1. Content (Inhalt):

- Der eigentliche Inhalt des Elements.
- Definiert durch die Breite (width) und Höhe (height) des Elements.
- Beispiel: Ein Absatz-Element `<p>` mit dem Text "Lorem ipsum dolor sit amet" und darunter ein Button `<button>` mit der Beschriftung "Klick mich!".

2. Padding (Innenabstand):

- Der Raum zwischen dem Inhalt des Elements und seiner Grenze (Border).
- Definiert durch padding-top, padding-right, padding-bottom und padding-left.

- o Beispiel: `<p style="padding: 20px;">Lorem ipsum dolor sit amet</p>`. Hier ist der Innenabstand des Absatzes 20px von allen Seiten.

3. Border (Rahmen):

- o Eine Linie um den Inhalt und das Padding herum.
- o Definiert durch `border-width`, `border-style` und `border-color`.
- o Beispiel: `<button style="border: 1px solid black;">Klick mich!</button>`. Hier hat der Button eine schwarze Linie mit einer Dicke von 1px um ihn herum.

4. Margin (Außenabstand):

- o Der Raum zwischen den Grenzen (Border) des Elements und seinen Nachbarelementen.
- o Definiert durch `margin-top`, `margin-right`, `margin-bottom` und `margin-left`.
- o Beispiel: `<button style="margin-top: 10px;">Klick mich!</button>`. Hier hat der Button einen oberen Außenabstand von 10px zu seinem Nachbarelement.

HTML:

```
<!-- Content (Inhalt) -->
<p class="content">Lorem ipsum dolor sit amet</p>

<!-- Padding (Innenabstand) -->
<p class="padding">Lorem ipsum dolor sit amet</p>

<!-- Border (Rahmen) -->
<button class="border">Klick mich!</button>

<!-- Margin (Außenabstand) -->
<button class="margin">Klick mich!</button>
```

CSS:

```
.content,
.padding {
    background-color: lightblue;
}

/* Padding (Innenabstand) */
.padding {
    padding: 20px;
}

/* Border (Rahmen) */
.border {
    border: 5px solid black;
}

/* Margin (Außenabstand) */
.margin {
    margin-top: 100px;
}
```