

KOMBINATION VON SELEKTOREN

Selektoren können auf verschiedene Arten miteinander kombiniert werden:

- Gruppierung
- Sequenzen
- Kombinatoren

GRUPPIERUNG VON SELEKTOREN

- Sollen die gleichen Deklarationen für mehrere Elemente angewendet werden, so können diese zu einer Regel zusammengefasst werden
- Dazu können mehrere Selektoren mittels Kommata gruppiert werden → CSS-Code wird kompakter

Beispiel:

```
/* Mehrere Selektoren, die die  
   gleichen Eigenschaften definieren... */  
h1 { background-color: blue; }  
  
h2 { background-color: blue; }  
  
.highlighted { background-color: blue; }
```

```
/* ...können auch durch Kommata  
   gruppiert werden */  
h1, h2, .highlighted {  
    background-color: blue;  
}
```

SEQUENZEN VON SELEKTOREN

- Selektoren können als **Sequenz** miteinander verknüpft werden, um Elemente spezifischer auszuwählen
- Dazu werden die Selektoren direkt hintereinander geschrieben (ohne Trennzeichen)
- Eine Sequenz muss immer mit einem Typselektor oder einem Universalselektor beginnen
- Steht am Anfang der Sequenz weder ein Typselektor noch ein Universalselektor, so wird implizit der Universalselektor angenommen, z.B.:

`.wichtig` ist äquivalent zu `*.wichtig`

`[href$=".de"]` ist äquivalent zu `*[href$=".de"]`

SEQUENZEN VON SELEKTOREN: BEISPIELE

style.css

```
/* Links, deren "href"-Attribut mit "http" beginnt und über  
denen sich der Mauszeiger befindet */  
a[href^="http"]:hover { background-color: red; }  
  
/* Absätze der Klasse "wichtig" */  
p.wichtig { border: 3px dotted; }  
  
/* Absätze, die zu den Klassen "wichtig" und "warnung" gehören*/  
p.wichtig.warnung {  
    border-color: red;  
}
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css"  
        href="style.css">  
  </head>  
  
  <body>  
    <a href="http://foo.de">Link</a>  
    <p class="wichtig">Erster Absatz</p>  
    <p class="wichtig warnung">Zweiter Absatz</p>  
  </body>  
</html>
```

Link

Erster Absatz

Zweiter Absatz

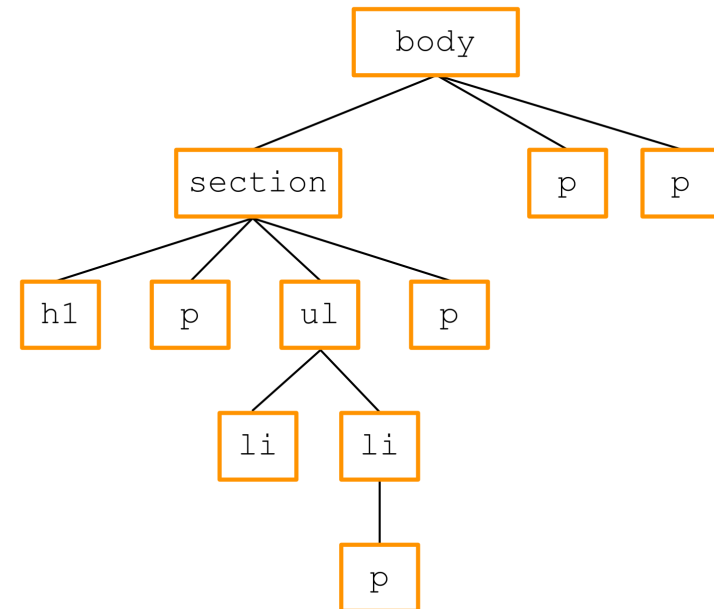
KOMBINATOREN

- Selektoren bzw. Sequenzen von Selektoren können mit **Kombinatoren** verkettet werden
- Kombinatoren betrachten Beziehungen zwischen selektierten Elementen

KOMBINATOREN: BEZIEHUNGEN

Beispiel: Der folgende HTML-Ausschnitt kann als Baumstruktur (rechts) dargestellt werden:

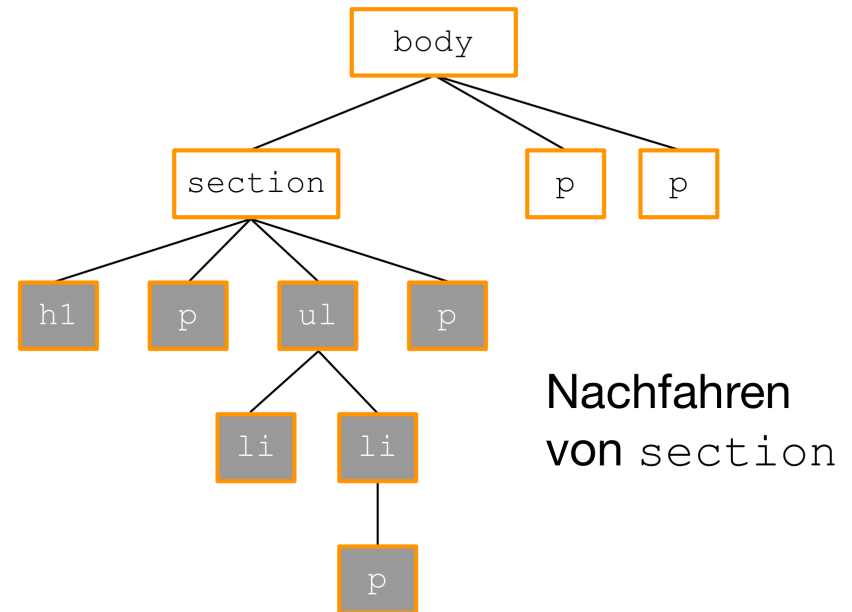
```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```



KOMBINATOREN: BEZIEHUNGEN

Nachfahren (*descendants*): Alle Unterelemente eines Elements (beliebige Tiefe)

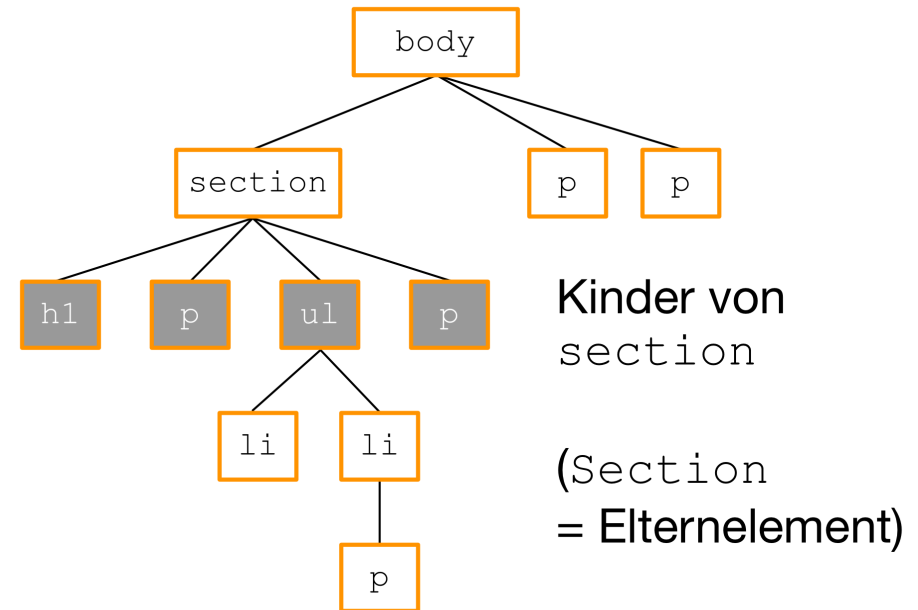
```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```



KOMBINATOREN: BEZIEHUNGEN

Kinder (*children*): Die direkten Nachfahren eines Elements (Tiefe 1)

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```



KOMBINATOREN: SYNTAX

Gegeben seien zwei Selektoren bzw. Sequenzen von Selektoren A und B:

Kombinator	Syntax	Bedeutung
Nachfahrenselektor	$A \ B$	B wird selektiert, wenn es ein Nachfahre von A ist
Kindselektor	$A > B$	B wird selektiert, wenn es ein direktes Kind von A ist
Nachbarselektor	$A + B$	B wird selektiert, wenn es im selben Eltern-Element direkt nach A vorkommt
Geschwisterselektor	$A \sim B$	B wird selektiert, wenn es im selben Eltern-Element nach A vorkommt

BEISPIEL: NACHFAHRENSELEKTOR

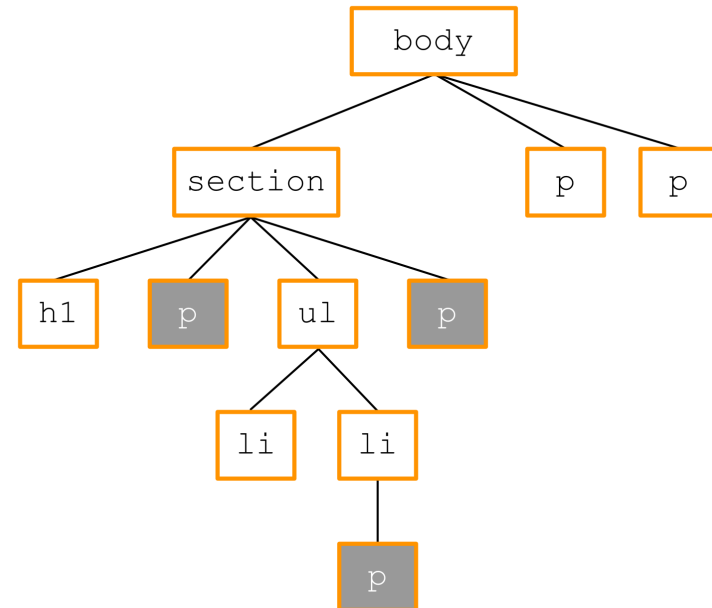
style.css

```
section p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Selektierte Elemente:



BEISPIEL: NACHFAHRENSELEKTOR (2)

style.css

```
section p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Überschrift

Absatz 1

- Eintrag 1
- Absatz in Eintrag 2

Absatz 2

Absatz 1 nach Section

Absatz 2 nach Section

BEISPIEL: KINDSELEKTOR

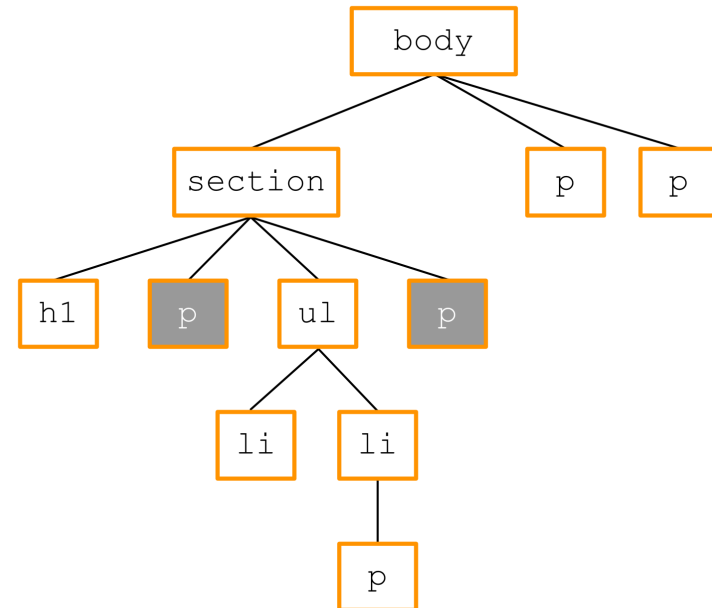
style.css

```
section > p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Selektierte Elemente:



BEISPIEL: KINDSELEKTOR (2)

style.css

```
section > p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Überschrift

Absatz 1

- Eintrag 1
- Absatz in Eintrag 2

Absatz 2

Absatz 1 nach Section

Absatz 2 nach Section

BEISPIEL: NACHBARSELEKTOR

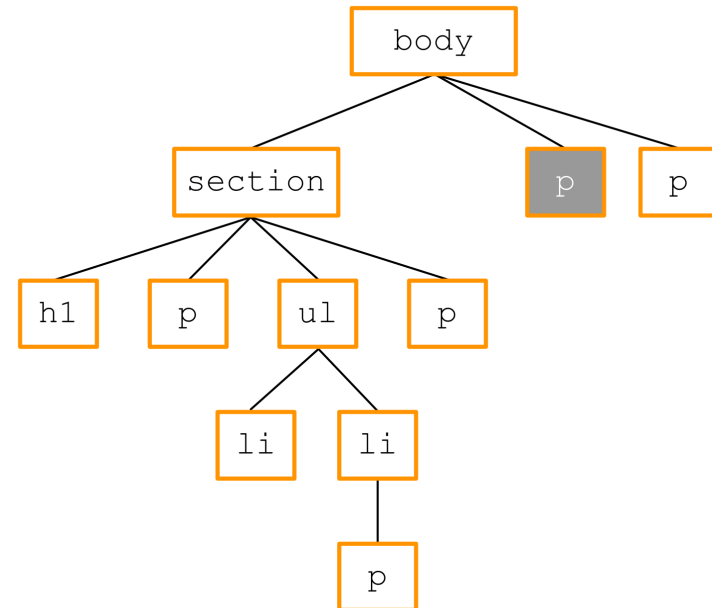
style.css

```
section + p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Selektierte Elemente:



BEISPIEL: NACHBARSELEKTOR (2)

style.css

```
section + p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Überschrift

Absatz 1

- Eintrag 1
- Absatz in Eintrag 2

Absatz 2

Absatz 1 nach Section

Absatz 2 nach Section

BEISPIEL: GESCHWISTERSELEKTOR

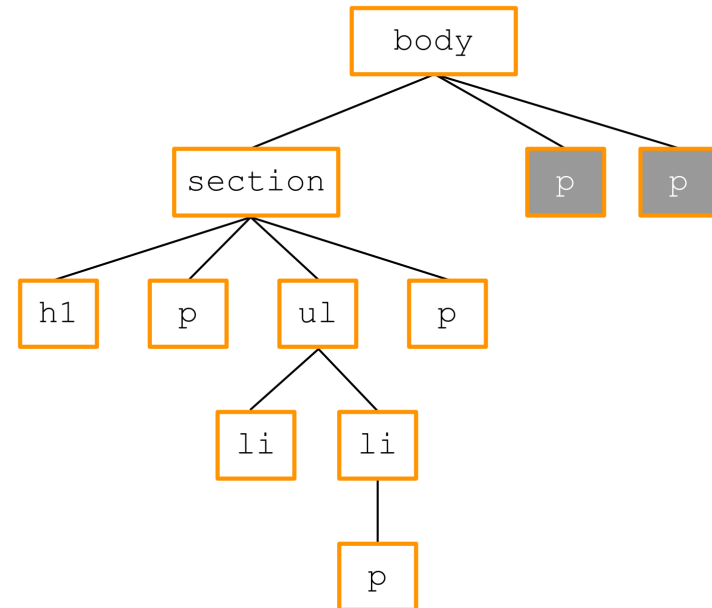
style.css

```
section ~ p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Selektierte Elemente:



BEISPIEL: GESCHWISTERSELEKTOR (2)

style.css

```
section ~ p { background-color: red; }
```

seite.html

```
[...]
<body>
  <section>
    <h1>Überschrift</h1>
    <p>Absatz 1</p>
    <ul>
      <li>Eintrag 1</li>
      <li>
        <p>Absatz in Eintrag 2</p>
      </li>
    </ul>
    <p>Absatz 2</p>
  </section>
  <p>Absatz 1 nach Section</p>
  <p>Absatz 2 nach Section</p>
</body>
[...]
```

Überschrift

Absatz 1

- Eintrag 1
- Absatz in Eintrag 2

Absatz 2

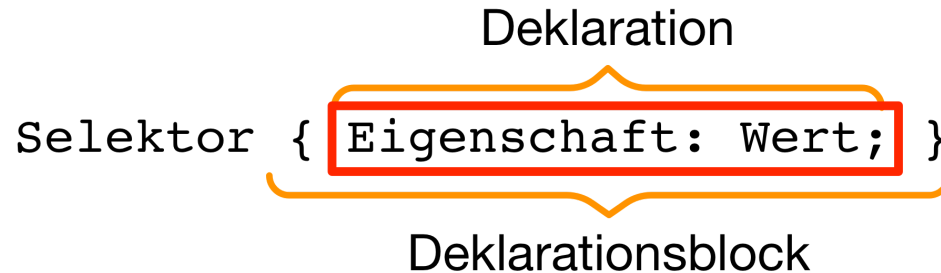
Absatz 1 nach Section

Absatz 2 nach Section

MEHR ZU SELEKTOREN IN CSS:

[Selectors Level 3](#) 

CSS-EIGENSCHAFTEN



- Erinnerung: Eine Deklaration in einer CSS-Regel besteht aus zwei Teilen:
 1. Die **Eigenschaft**, welche modifiziert werden soll
 2. Der **Wert** (bzw. mehrere Werte) für die Eigenschaft
- CSS bietet mehrere hundert Eigenschaften, daher im Folgenden einige Beispiele für verschiedene Anwendungsbereiche

FARBEN

In den bisherigen Beispielen haben wir schon zwei wichtige Eigenschaften zur Beeinflussung der Farbe von Elementen gesehen:

- `color`: Textfarbe
- `background-color`: Hintergrundfarbe

style.css

```
h1 { background-color: lightskyblue; }  
p { color: darkorchid; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <h1>Überschrift</h1>  
    <p>Absatz</p>  
  </body>  
</html>
```

Überschrift

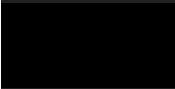








Absatz

FARBEN (2)

Farbwerte können in CSS auf verschiedene Arten angegeben werden:

- Definierte Farbnamen
- RGB-Werte (**R**ot, **G**rün, **B**lau) in verschiedenen Schreibweisen:
 - Hexdezimal in der Form #RRGGBB → RR/GG/BB sind jeweils Hex-Werte von 0-FF (dezimal: 0-255)
 - Über die CSS-Funktion `rgb(Rot, Grün, Blau)` → Werte für Rot/Grün/Blau entweder dezimal (0-255) oder in Prozent
- HSL-Werte (**H**ue = Farbton, **S**aturation = Sättigung, **L**ightness = Helligkeit):
 - Über die CSS-Funktion `hsl(Farbton, Sättigung, Helligkeit)`
 - Farbton ist ein Dezimalwert von 0-360, Sättigung und Helligkeit sind Prozentwerte

FARBWERTE: BEISPIELE

Farbe	Farbname	Hex	RGB	HSB
	black	#000000	rgb(0,0,0)	hsl(0,0%,0%)
	silver	#C0C0C0	rgb(192,192,192)	hsl(0,0%,75%)
	gray	#808080	rgb(128,128,128)	hsl(0,0%,50%)
	white	#FFFFFF	rgb(255,255,255)	hsl(0,0%,100%)
	maroon	#800000	rgb(128,0,0)	hsl(0,100%,25%)
	red	#FF0000	rgb(255,0,0)	hsl(0,100%,50%)
	purple	#800080	rgb(128,0,128)	hsl(300,100%,25%)
	green	#008000	rgb(0,128,0)	hsl(120,100%,25%)
	yellow	#FFFF00	rgb(255,255,0)	hsl(60,100%,50%)
	blue	#0000FF	rgb(0,0,255)	hsl(240,100%,50%)

FARBEN: TRANSPARENZ

- Für RGB und HSL existieren jeweils Varianten, um zusätzlich die Transparenz (*Alphawert*) anzugeben:
 - `rgba(Rot, Grün, Blau, Alphawert)`
 - `hsla(Farbton, Sättigung, Helligkeit, Alphawert)`
- Der Alphawert ist eine Fließkommazahl von 0.0 (vollständige Transparenz) bis 1.0 (keine Transparenz)

Beispiel:

`rgba(255,0,0,0.0)`

`rgba(255,0,0,0.5)`

`rgba(255,0,0,1.0)`

FARBVERLÄUFE

Lineare Farbverläufe über CSS-Funktion `linear-gradient`

- Syntax:

```
linear-gradient(Richtung, Farbstopp1, ..., Farbstoppn)
```

- `Richtung` kann die Angabe eines Winkels sein (Maßeinheit `deg`, z.B. `30deg`) oder ein Schlüsselwort wie z.B. `to left`, `to bottom`, `to top right`, `to bottom left`
- `Farbstoppx` ist ein Farbwert mit einer optionalen Prozentangabe für die Positionierung des Farbüberganges


FARBVERLÄUFE (2)

Radiale Farbverläufe über CSS-Funktion `radial-gradient`

- Syntax (vereinfacht):

```
radial-gradient(Form, Farbstopp1, ..., Farbstoppn)
```

- `Form` kann ein Kreis (`circle`) oder eine Ellipse (`ellipse`) sein
- `Farbstoppx` ist ein Farbwert mit einer optionalen Prozentangabe für die Positionierung des Farbüberganges

! Weitere Möglichkeiten von `radial-gradient` → [CSS Image Values and Replaced Content Module Level 3](#) 

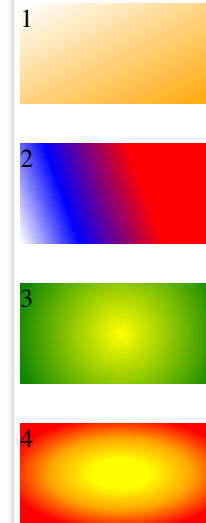
FARBVERLÄUFE: BEISPIEL

style.css

```
p.styled {  
  width: 100px;  
  height: 50px;  
}  
  
p.linear1 {  
  background: linear-gradient(to bottom right, white, orange);  
}  
  
p.linear2 {  
  background: linear-gradient(70deg, white 0%, blue 25%, red 60%);  
}  
  
p.radial1 { background: radial-gradient(circle, yellow, green); }  
  
p.radial2 {  
  background: radial-gradient(ellipse, yellow 20%,  
                             orange 50%, red 80%);  
}
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <p class="styled linear1">1</p>  
    <p class="styled linear2">2</p>  
    <p class="styled radial1">3</p>  
    <p class="styled radial2">4</p>  
  </body>  
</html>
```



MEHR ZU FARBEN IN CSS:

[CSS Color Module Level 3](#) 

Ausprobieren und Auswählen von Farben:

[HTML Color Codes](#) 

NUMERISCHE WERTE UND MASSEINHEITEN

- Viele Eigenschaften ermöglichen/erfordern die Angabe von numerischen Werten
- Numerische Werte können Ganzzahlen oder Fließkommazahlen sein (für das Komma wird ein Punkt verwendet, z.B. 1 . 5)
- Numerische Werte können absolut oder relativ sein
- Häufig erfolgt die Angabe in Kombination mit einer Maßeinheit

MASSEINHEITEN IN CSS: BEISPIELE

Maßeinheit	CSS-Name	Beziehung	Bedeutung
Pixel	px	relativ bzgl. verschiedener Ausgabegeräte, absolut bzgl. der Anzeige auf einem konkreten Ausgabegerät	Abhängig von der Pixeldichte des Ausgabegeräts, z.B. <code>font-size: 12px;</code>
Zentimeter, Millimeter	cm, mm	absolut	z.B. <code>margin-top: 2.5cm</code>
Geviertbreite	em	relativ zur Schriftgröße des Elements	Wird em für die Schriftgröße (<code>font-size</code>) selbst verwendet, dann relativ zur Schriftgröße des Eltern-Elements, z.B. <code>font-size: 0.8em;</code>
Prozent	%	relativ	Je nach CSS-Eigenschaft relativ zum Wert einer anderen Eigenschaft (z.B. desselben Elements oder des Elternelements), z.B. bezieht sich <code>line-height: 50%;</code> auf die <code>font-size</code>
Viewport-Breite/-Höhe	vw/vh	relativ	Entspricht 1% der Breite/Höhe des sichtbaren Anzeigebereiches (=Viewport)

EIGENSCHAFTEN FÜR TEXT: SCHRIFT

Schriftart: `font-family`

- Wert kann konkrete Schriftart sein, z.B. `font-family: Arial;`
- ! Klappt nur, wenn die Schriftart auf dem Betriebssystem des Benutzers installiert ist!
- Wert kann auch eine Schriftklasse sein:

Schriftklasse	Bedeutung	Beispiele
<code>serif</code>	Schriften mit Serifen	Times, Times New Roman
<code>sans-serif</code>	Schriften ohne Serifen	Arial, Helvetica
<code>monospace</code>	Festbreitenschrift	Courier, Courier New
<code>cursive</code>	Handschriften	Comic Sans MS, <i>Brush Script MT</i>
<code>fantasy</code>	"Dekorative" Schriften	Impact , Chalkduster

EIGENSCHAFTEN FÜR TEXT: SCHRIFT (2)

Schriftart: `font-family` (Fortsetzung)

- Bei Angabe einer Schriftklasse wählt der Browser eine passende Schriftart
- Best Practice:
 1. Mehrere Schriftarten als Liste angeben → Browser wählt die erste installierte Schriftart
 2. Am Ende der Liste eine Schriftklasse als "Fallback" angeben

Beispiel: `font-family: Helvetica, Arial, sans-serif;`

EIGENSCHAFTEN FÜR TEXT: SCHRIFT (3)

Schriftgröße: **font-size**

- Numerischer Wert, oder
- eines der Schlüsselwörter: `small`, `x-small`, `xx-small`, `medium`, `large`, `x-large`, `xx-large` (`medium` ist Normalgröße, Unterschied jeweils Faktor 1.2)

Schriftstil und -gewicht: **font-style**, **font-weight**

- z.B. kursive Schrift: `font-style: italic;`
- z.B. fette Schrift: `font-weight: bold;`

SCHRIFT: BEISPIEL

style.css

```
p.serious {  
  font-family: Arial,sans-serif;  
  font-style: italic;  
  font-size: 3em;  
}  
  
p.stylish {  
  font-family: fantasy;  
  font-size: large;  
  font-weight: bold;  
}
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css"  
      href="style.css">  
  </head>  
  
  <body>  
    <p class="serious">Ein Absatz</p>  
    <p class="stylish">Noch ein Absatz</p>  
  </body>  
</html>
```

Ein Absatz

Noch ein Absatz

EIGENSCHAFTEN FÜR TEXT: ABSTÄNDE

Zeilenabstand: **line-height**

- Numerischer Wert
- Bezieht sich bei relativem Wert auf die `font-size`, z.B.
`line-height: 1.5em;`

Buchstaben- und Wortabstand: **letter-spacing, word-spacing**

- Numerischer Wert
- z.B. `letter-spacing: 0.5em; word-spacing: 10px;`

EIGENSCHAFTEN FÜR TEXT: AUSRICHTUNG

Horizontale Ausrichtung: **text-align**

Mögliche Werte: `left` (linksbündig), `center` (zentriert),
`right` (rechtsbündig), `justify` (Blocksatz)

Vertikale Ausrichtung: **vertical-align**

Mögliche Werte: `top` (oben), `middle` (mittig), `bottom` (unten)

ABSTÄNDE UND AUSRICHTUNG: BEISPIEL

style.css

```
p.buchstaben { letter-spacing: 0.5em; }  
  
p.worte { word-spacing: 20px; }  
  
p.zeilen { line-height: 2em; }  
  
p.zentriert { text-align: center; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <p class="buchstaben">Größerer Abstand zwischen  
      den Buchstaben</p>  
    <p class="worte">Größerer Abstand zwischen den Wörtern</p>  
    <p class="zeilen">Größerer Abstand zwischen den einzelnen  
      Zeilen, wodurch sich die Lesbarkeit durchaus  
      merklich erhöhen kann.</p>  
    <p class="zentriert">Ein zentrierter Absatz</p>  
  </body>  
</html>
```

G r ö ß e r e r A b s t a n d z w i s c h e n d e n
B u c h s t a b e n

Größerer Abstand zwischen den Wörtern

Größerer Abstand zwischen den einzelnen Zeilen, wodurch sich die
Lesbarkeit durchaus merklich erhöhen kann.

Ein zentrierter Absatz

EIGENSCHAFTEN FÜR TEXT: EFFEKTE

Text-Dekoration: **text-decoration**

Mögliche Werte: `underline` (unterstrichen), `line-through` (durchgestrichen), `overline` (überstrichen), `none` (keine Dekoration)

Groß- und Kleinschreibung: **text-transform**

Mögliche Werte: `uppercase` (alles in Großbuchstaben), `lowercase` (alles in Kleinbuchstaben), `capitalize` (erster Buchstabe als Großbuchstabe)

EIGENSCHAFTEN FÜR TEXT: EFFEKTE (2)

Schatten: `text-shadow`

Konfigurierbar über bis zu vier Angaben:

1. Horizontaler Versatz (numerischer Wert)
2. Vertikaler Versatz (numerischer Wert)
3. Verlaufsradius (optional, numerischer Wert)
4. Farbe (optional, Farbwert)

Beispiel: `text-shadow: 5px 5px 2px gray;`

EFFEKTE: BEISPIEL

style.css

```
h1 {  
  color: maroon;  
  text-shadow: 3px 3px 1px gray;  
}  
  
.gestrichen { text-decoration: line-through; }  
  
.gross { text-transform: uppercase; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <h1>Überschrift</h1>  
    <p>  
      Auf <span class="gestrichen">diesen Text</span>  
      wurden <span class="gross">Effekte angewendet</span>.  
    </p>  
  </body>  
</html>
```

Überschrift

Auf ~~diesen Text~~ wurden EFFEKTE ANGEWENDET.

EIGENSCHAFTEN FÜR AUFZÄHLUNGEN

Aufzählungszeichen anpassen: `list-style-type`

- Für ungeordnete Aufzählungen:
`none` (kein Zeichen), `disc` (gefüllter Kreis), `square` (Quadrat), `circle` (leerer Kreis)
- Für geordnete Aufzählungen u.A.:
`none` (keine Nummerierung), `decimal` (1., 2., 3., ...),
`lower-alpha` (a., b., c., ...), `upper-alpha` (A., B., C., ...),
`upper-roman` (I., II., III., ...)

Bilder als Aufzählungszeichen: `list-style-image`

Wert: `url (Pfad)` mit Pfad zum gewünschten Bild

AUFZÄHLUNGEN: BEISPIEL

style.css

```
ul.quadrat { list-style-type: square; }  
  
ol.roman { list-style-type: upper-roman; }  
  
ul.hand { list-style-image: url(assets/hand.png) }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <ul class="quadrat">  
      <li>JavaScript</li>  
      <li>CSS</li>  
      <li>HTML</li>  
    </ul>  
    <ol class="roman">  
      <li>Vue.js</li>  
      <li>Angular</li>  
      <li>React</li>  
    </ol>  
    <ul class="hand">  
      <li>JavaScript</li>  
      <li>CSS</li>  
    </ul>  
  </body>  
</html>
```

- JavaScript
- CSS
- HTML

I. Vue.js
II. Angular
III. React



JavaScript

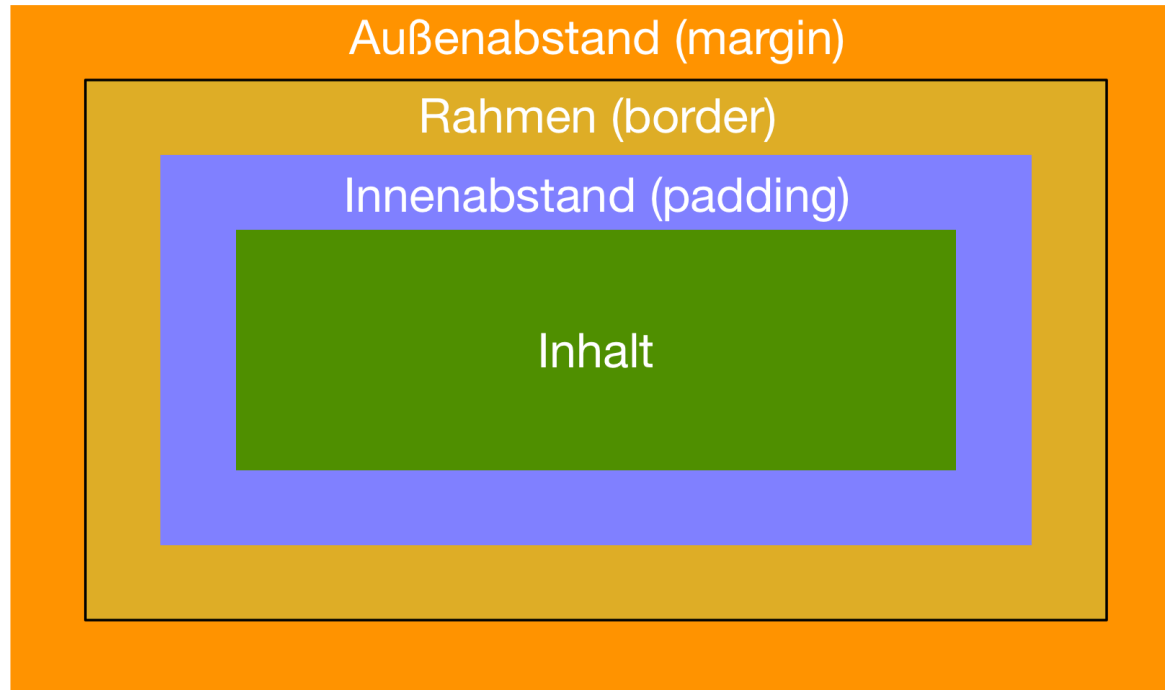


CSS

BOX-MODELL

- Aus Sicht von CSS ist jedes HTML-Element ein rechteckiges Kästchen (= eine **Box**)
- Das **Box-Modell** beschreibt eine Reihe von Eigenschaften, die zu einer solchen Box gehören
- Das Box-Modell ist eine wichtige Grundlage für Formatierung, Positionierung und Layouting mit CSS

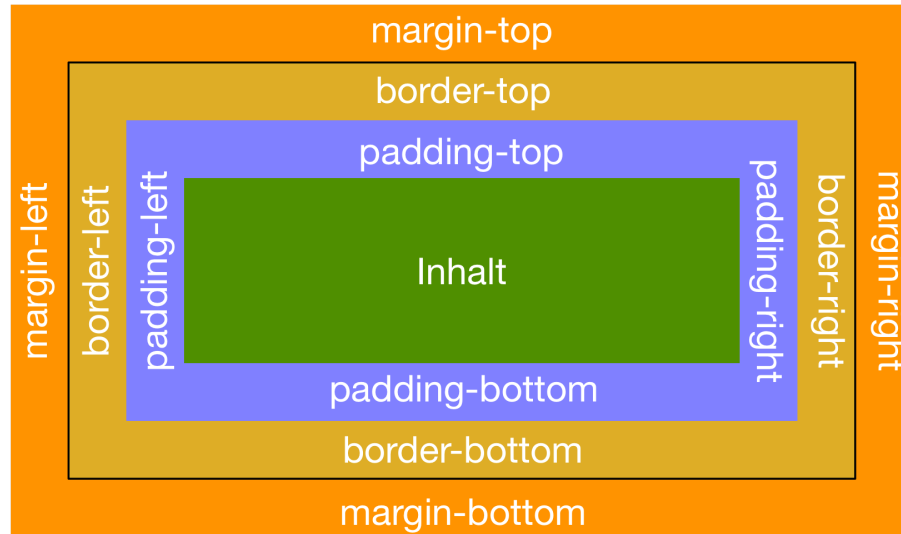
BOX-MODELL (2)



Eine Box besteht aus folgenden Teilen:

- Inhaltsbereich
- Innenabstand (optional)
- Rahmen (optional)
- Außenabstand (optional)

BOX-MODELL (3)



Zu jedem Teil der Box gibt es entsprechende CSS-Eigenschaften:

- Innenabstand: `padding`, `padding-top`, `padding-left`, `padding-right`, `padding-bottom`
- Rahmen: `border`, `border-top`, `border-left`, `border-right`, `border-bottom`
- Außenabstand: `margin`, `margin-top`, `margin-left`, `margin-right`, `margin-bottom`

BOX-MODELL: INNEN- UND AUSSENABSTAND

- Die Eigenschaften für Innen- und Außenabstand (`padding` und `margin`) akzeptieren jeweils einen numerischen Wert
- Mit den Varianten `-top`, `-left`, `-right` und `-bottom` können die Abstände oben, links, rechts und unten jeweils separat definiert werden
- Innen- und Außenabstand sind transparent - sie übernehmen die Hintergrundfarbe des Inhaltsbereiches bzw. des umgebenen Elements
- ! `margin-top` und `margin-bottom` gelten nicht für Inline-Elemente

BOX-MODELL: INNEN- UND AUSSENABSTAND (2)

padding bzw. margin sind Kurzschreibweisen, mit denen man – top, –left, –right und –bottom gleichzeitig setzen kann (im Uhrzeigersinn startend bei top)

Beispiele (margin analog):

```
/* top=10px, right=15px, bottom=5px, left=20px */  
#beispiel1 { padding: 10px 15px 5px 20px; }  
/* top=15px, right=left=5px, bottom=10px */  
#beispiel2 { padding: 15px 5px 10px }  
/* top=bottom=15px, right=left=10px */  
#beispiel3 { padding: 15px 10px }  
/* top=right=bottom=left=15px */  
#beispiel4 { padding: 15px }
```

BOX-MODELL: RAHMEN

Rahmendicke: **border-width**

Numerischer Wert

Rahmenstil: **border-style**

Mögliche Werte u.A.: `none` (kein Rahmen), `dotted` (gepunktet), `dashed` (gestrichelt), `solid` (durchgezogen), `double` (doppelter Rahmen)

Rahmenfarbe: **border-color**

Farbwert

BOX-MODELL: RAHMEN (2)

- Mit den Varianten `border-top`, `border-left`, `border-right` und `border-bottom` können die vier Seiten jeweils separat definiert werden
- `border` ist eine Kurzschreibweise, mit der `border-width`, `border-style` und `border-color` gleichzeitig definiert werden können

Beispiel:

```
/* width=5px, style=dotted, color=red */  
#beispiel { border: 5px dotted red; }
```


BOX-MODELL: BEISPIEL

style.css

```
p.warnung {  
  border: 5px solid red;  
  padding: 1em;  
  margin-bottom: 50px;  
  background-color: lightcoral;  
}  
  
p.hinweis {  
  border: 3px dotted blue;  
  padding: 1px;  
  margin: 25px;  
  background-color: skyblue;  
}
```

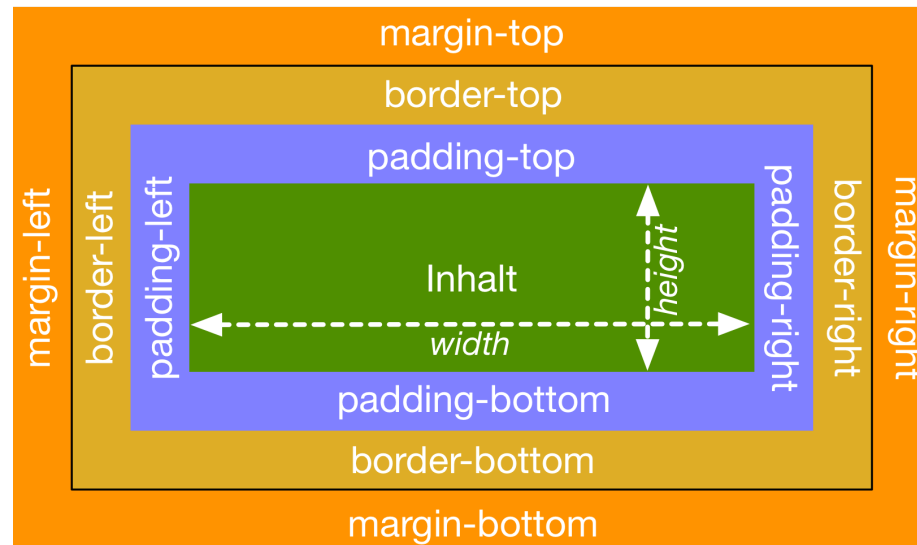
seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <p class="warnung">  
      Ein Absatz mit einer Warnung.  
    </p>  
    <p class="hinweis">  
      Ein Absatz mit einem Hinweis.  
    </p>  
  </body>  
</html>
```

Ein Absatz mit einer Warnung.

Ein Absatz mit einem Hinweis.

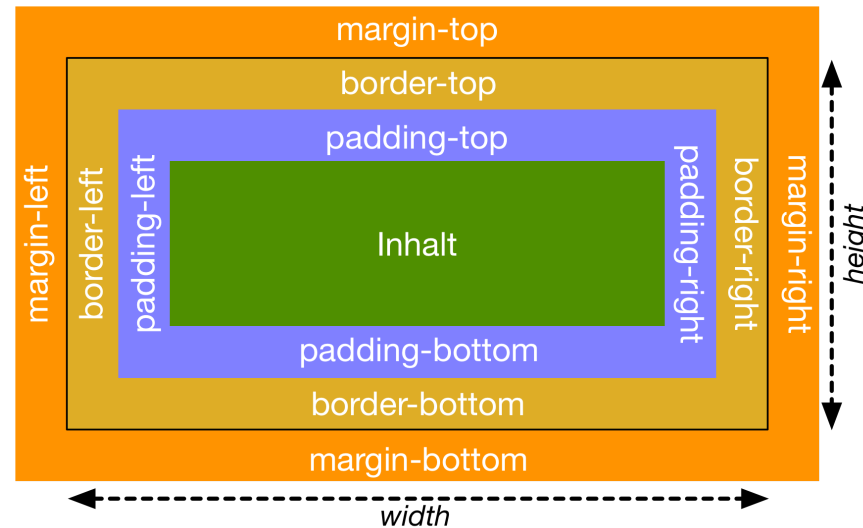
BOX-MODELL: GRÖSSE



Breite und Höhe eines Elements: **width**, **height**

- Nur für Blockelemente, nicht für Inline-Elemente
- ! Durch Setzen der Eigenschaften `width` und `height` wird lediglich die Größe des Inhaltsbereiches modifiziert!
- Für den tatsächlichen Platzbedarf eines Elements werden die Werte von Innenabstand, Rahmen und Außenabstand dazugerechnet, d.h.:
Gesamtbreite = `margin-left` + `border-left` + `padding-left` + `width` + `padding-right` + `border-right` + `margin-right` (Gesamthöhe analog)

BOX-MODELL: GRÖSSE (2)



Mit der Eigenschaft **box-sizing** kann die Bedeutung von **width** und **height** beeinflusst werden:

- Wert **content-box** (Standard) bewirkt das klassische Verhalten ($\text{width/height} = \text{Breite/Höhe des Inhalts}$)
- Wert **border-box** bewirkt, dass **width** und **height** auch den Innenabstand und Rahmen umfassen (Inhalt wird entsprechend verkleinert)

BOX-MODELL: ZU GROSSER INHALT

Ist der Inhalt größer als der Inhaltsbereich, so fließt dieser "aus der Box heraus", Beispiel:

style.css

```
p.box {  
  border: 1px solid;  
  width: 100px;  
  height: 100px;  
}
```

seite.html

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Titel meiner Web-Seite</title>  
<meta charset="utf-8">  
<link rel="stylesheet" type="text/css" href="style.css">  
</head>  
  
<body>  
<p class="box">  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed  
  do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
</p>  
</body>  
</html>
```

Lorem ipsum dolor
sit amet,
consectetur
adipiscing elit, sed
do eiusmod tempor
incidunt ut
labore et dolore
magna aliqua.

BOX-MODELL: ZU GROSSER INHALT (2)

Dieses Verhalten kann mit der Eigenschaft **overflow** beeinflusst werden:

- Wert `visible` (Standard): Inhalt wird komplett angezeigt
- Wert `hidden`: "Überfließender" Inhalt wird abgeschnitten und ausgeblendet
- Wert `scroll`: "Überfließender" Inhalt wird abgeschnitten und ausgeblendet, innerhalb der Box kann jedoch gescrolled werden
- Wert `auto`: Browser entscheidet (meistens: `scroll`)

Varianten **overflow-x** und **overflow-y**:

Gleiche Werte möglich, jedoch spezifisch für links/rechts (`overflow-x`) bzw. oben/unten (`overflow-y`) "herausfließenden" Text

ZU GROSSER INHALT: BEISPIEL

style.css

```
p.box {  
border: 1px solid;  
width: 100px;  
height: 100px;  
}  
  
p.hide { overflow: hidden; }  
  
p.scroll { overflow: scroll; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Titel meiner Web-Seite</title>  
  <meta charset="utf-8">  
  <link rel="stylesheet" type="text/css" href="style.css">  
</head>  
  
<body>  
  <p class="box hide">  
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed  
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
  </p>  
  <p class="box scroll">  
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed  
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
  </p>  
</body>  
</html>
```

Lorem ipsum dolor
sit amet,
consectetur
adipiscing elit, sed
do eiusmod tempor
incididunt ut

Lorem ipsum dolor
sit amet,
consectetur
adipiscing elit, sed
do eiusmod tempor
incididunt ut

BOX-EFFEKTE

Schatten: **box-shadow**

Konfigurierbar über bis zu sechs Angaben:

1. Horizontaler Versatz (numerischer Wert)
2. Vertikaler Versatz (numerischer Wert)
3. Verschwommenheit (*blur*, optional, numerischer Wert)
4. Ausbreitung (*spread*, optional, numerischer Wert)
5. Farbe (optional, Farbwert)
6. Wert `inset` (optional, nur als erste oder letzte Angabe)
bewirkt, dass der Schatten innen statt außen ist

Beispiel: `box-shadow: 5px 5px 2px 4px gray;`

BOX-EFFEKTE (2)

Abgerundete Ecken: **border-radius**

Konfigurierbar über eine bis vier Angaben (Eckradius als numerische Werte, im Uhrzeigersinn: oben links, oben rechts, unten rechts, unten links)

Beispiele:

```
/* oben links: 0px, oben rechts: 0px, unten rechts: 10px, unten links: 10px */  
#beispiel1 { border-radius: 0px 0px 10px 10px; }  
/* oben links: 10px, oben rechts & unten links: 5px, unten rechts: 15px */  
#beispiel2 { border-radius: 10px 5px 15px }  
/* oben links & unten rechts: 10px, oben rechts & unten links: 15px */  
#beispiel3 { border-radius: 10px 15px }  
/* alle Ecken: 15px */  
#beispiel4 { border-radius: 10px }
```


BOX-EFFEKTE: BEISPIEL

style.css

```
p.warnung {  
  border: 2px solid red;  
  background-color: lightcoral;  
  box-shadow: 4px 4px 4px gray;  
  border-radius: 10px 5px;  
}  
  
p.hinweis {  
  border: 2px dotted blue;  
  text-align: center;  
  border-radius: 70%;  
}
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <p class="warnung">  
      Ein Absatz mit einer Warnung.  
    </p>  
    <p class="hinweis">  
      Ein Absatz mit einem Hinweis.  
    </p>  
  </body>  
</html>
```

Ein Absatz mit einer Warnung.

Ein Absatz mit einem Hinweis.

EIGENSCHAFTEN FÜR TABELLEN

Rahmen von Zellen zusammenfallen lassen: **border-collapse**

- Wert `separate` (Standard): die Rahmen benachbarter Tabellenzellen werden getrennt angezeigt
- Wert `collapse`: die Rahmen benachbarter Tabellenzellen fallen zusammen

Abstand zwischen Zellen: **border-spacing**

- Entweder ein numerischer Wert (Abstand in alle Richtungen) oder zwei numerische Werte (horizontaler und vertikaler Abstand)
- Wirkt sich nur aus, wenn `border-collapse: separate`; gilt

TABELLEN: BEISPIEL

style.css

```
table, tr, td { border: 1px solid; }

#tabelle1 {
  border-collapse: separate;
  margin-bottom: 1em;
  border-spacing: 15px;
}

#tabelle2 {
  border-collapse: separate;
  margin-bottom: 1em;
  border-spacing: 5px 10px;
}

#tabelle3 { border-collapse: collapse; }
```

seite.html

```
<!DOCTYPE html>
<html>
  <head>
    [...]
  </head>

  <body>
    <table id="tabelle1">
      <tr><td>A</td><td>B</td></tr>
      <tr><td>C</td><td>D</td></tr>
    </table>
    <table id="tabelle2">
      <tr><td>A</td><td>B</td></tr>
      <tr><td>C</td><td>D</td></tr>
    </table>
    <table id="tabelle3">
      <tr><td>A</td><td>B</td></tr>
      <tr><td>C</td><td>D</td></tr>
    </table>
  </body>
</html>
```

A	B
C	D

A	B
C	D

A	B
C	D

KASKADE

Eintrag im [Duden](#)  :

KASKADE, DIE

Bedeutungsübersicht:

- 1. in Form von Stufen künstlich angelegter Wasserfall*
- 2. wagemutiger Sprung, bei dem der Artist einen Absturz vortäuscht*
- 3. (chemische Technik) Anordnung hintereinandergeschalteter, gleichartiger Gefäße*

KASKADE (CSS)

- Wir haben bereits gesehen,
 1. dass CSS-Deklarationen aus verschiedenen Quellen kommen können (z.B. interne und externe Stylesheets), und
 2. dass bei Konflikten die Reihenfolge der Deklarationen wichtig ist
- Stylesheets können also **hintereinandergeschaltet** und miteinander **kombiniert** werden
- Die **Kaskade** ist der Prozess, der Deklarationen aus verschiedenen Quellen miteinander kombiniert und auftretende Konflikte auflöst

KASKADE: ABLAUF

- Die Kaskade verwendet ein Regelsystem, um die anzuwendenden Deklarationen zu bestimmen
- Dabei verwendet die Kaskade folgende Kriterien (mit absteigender Priorität^{*}):
 1. Herkunft und Wichtigkeit
 2. Spezifität
 3. Reihenfolge des Auftretens

^{*} d.h. Kriterium 1 wiegt stärker als Kriterium 2, Kriterium 1 und 2 wiegen stärker als Kriterium 3

KASKADE: ABLAUF

- Die Kaskade verwendet ein Regelsystem, um die anzuwendenden Deklarationen zu bestimmen
- Dabei verwendet die Kaskade folgende Kriterien (mit absteigender Priorität^{*}):
 - 1. Herkunft und Wichtigkeit**
 2. Spezifität
 3. Reihenfolge des Auftretens

^{*} d.h. Kriterium 1 wiegt stärker als Kriterium 2, Kriterium 1 und 2 wiegen stärker als Kriterium 3

HERKUNFT

CSS unterscheidet drei Quellen für Stylesheets:

Autoren-Stylesheet (höchste Priorität)

Auf der Web-Seite durch den Web-Entwickler erstellte und referenzierte Stylesheets (bisherige Beispiele)

Benutzer-Stylesheet (mittlere Priorität)

Eigene Einstellungen des Benutzers (z.B. zur Verbesserung der Barrierefreiheit), Beispiel: [stylish](#) 

Browser-Stylesheet (niedrige Priorität)

Standard-Stylesheet des Browsers

WICHTIGKEIT

- Bei mehrfach definierten Regeln oder Eigenschaften gilt:
Die letzte Deklaration gewinnt
 - Mit dem Schlüsselwort `!important` kann die Priorität einer Deklaration erhöht werden
- ! Empfehlung: Auf `!important` verzichten/sparsam einsetzen (erschwert Lesbarkeit, Debugging, etc.)!

WICHTIGKEIT: BEISPIEL

style.css

```
p { color: orange !important; }  
p { color: blue; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css"  
          href="style.css">  
  </head>  
  
  <body>  
    <p>  
      Normalerweise wäre ich blau, wegen  
      !important bin ich aber orange.  
    </p>  
  </body>  
</html>
```

Normalerweise wäre ich blau, wegen !important bin ich aber orange.

HERKUNFT UND WICHTIGKEIT

Die Kaskade priorisiert Herkunft und Wichtigkeit folgendermaßen
(mit absteigender Priorität):

1. `!important`-Deklarationen in Benutzer-Stylesheets
2. `!important`-Deklarationen in Autoren-Stylesheets
3. Normale Deklarationen in Autoren-Stylesheets
4. Normale Deklarationen in Benutzer-Stylesheets
5. Normale Deklarationen im Browser-Stylesheet

KASKADE: ABLAUF

- Die Kaskade verwendet ein Regelsystem, um die anzuwendenden Deklarationen zu bestimmen
- Dabei verwendet die Kaskade folgende Kriterien (mit absteigender Priorität^{*}):
 1. Herkunft und Wichtigkeit
 - 2. Spezifität**
 3. Reihenfolge des Auftretens

^{*} d.h. Kriterium 1 wiegt stärker als Kriterium 2, Kriterium 1 und 2 wiegen stärker als Kriterium 3

SPEZIFITÄT

- Sind nach der Priorisierung nach Herkunft und Wichtigkeit noch gleichwertige/widersprüchliche Regeln übrig, so werden diese nach **Spezifität** gewichtet
- Die Spezifität ist ein Maß dafür, auf wie viele Elemente eine Regel zutreffen könnte, oder: wie *spezifisch* die Regel (bzw. ihr Selektor) ist
- Regel: Je spezifischer ein Selektor, desto höher ist seine Spezifität, desto höher ist seine Gewichtung/Priorität

SPEZIFITÄT

- Die Spezifität wird in Form von vier Zählern ausgedrückt: (*A*, *B*, *C*, *D*)
- Die Zähler werden folgendermaßen ermittelt:

Zähler	Bedeutung
<i>A</i>	Höchste Gewichtung: Wird auf 1 gesetzt, wenn die Deklaration in einem <code>style</code> -Attribut steht (Inline-Style)
<i>B</i>	+1 für jedes Vorkommen eines ID-Selektors
<i>C</i>	+1 für jedes Vorkommen eines Klassenselektors, Attributselektors oder einer Pseudoklasse
<i>D</i>	Niedrigste Gewichtung: +1 für jedes Vorkommen eines Typselektors oder Pseudoelements

! Universalselektoren und Kombinatoren bekommen keine Gewichtung!

SPEZIFITÄT: BEISPIELE

Selektoren	Spezifität (A,B,C,D)	Beschreibung
<code><p style="...">...</p></code>	(1,0,0,0)	1 Inline-Style
<code>#logo {...}</code>	(0,1,0,0)	1 ID-Selektor
<code>#logo:hover {...}</code>	(0,1,1,0)	1 ID-Selektor, 1 Pseudoklasse
<code>section p.wichtig {...}</code>	(0,0,1,2)	1 Klassenselektor, 2 Typselektoren
<code>* {...}</code>	(0,0,0,0)	1 Universalselektor

SPEZIFITÄT: CODE-BEISPIEL

style.css

```
/* Spezifität (0,0,0,1) */
p { color: green; }
/* Spezifität (0,1,1,1) */
#kurzbeschreibung p.hervorgehoben { color: blue; }
/* Spezifität (0,1,0,2) */
section#kurzbeschreibung p { color: orange; }
/* Spezifität (0,0,1,1) */
p.hervorgehoben { color: red; }
```

seite.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titel meiner Web-Seite</title>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css"
          href="style.css">
  </head>

  <body>
    <section id="kurzbeschreibung">
      <p class="hervorgehoben">Lorem ipsum.</p>
    </section>
    <p class="hervorgehoben">Dolor sit amet.</p>
    <p>Consectetur adipisicing elit.</p>
  </body>
</html>
```

Lorem ipsum.

Dolor sit amet.

Consectetur adipisicing elit.

KASKADE: ABLAUF

- Die Kaskade verwendet ein Regelsystem, um die anzuwendenden Deklarationen zu bestimmen
- Dabei verwendet die Kaskade folgende Kriterien (mit absteigender Priorität^{*}):
 1. Herkunft und Wichtigkeit
 2. Spezifität
 - 3. Reihenfolge des Auftretens**

^{*} d.h. Kriterium 1 wiegt stärker als Kriterium 2, Kriterium 1 und 2 wiegen stärker als Kriterium 3

REIHENFOLGE DES AUFTRETENS

Sind nach der Priorisierung nach Herkunft und Wichtigkeit sowie Spezifität immer noch gleichwertige/widersprüchliche Deklarationen übrig, so hat die zuletzt aufgetretene Deklaration Vorrang

Beispiel:

style.css

```
/* Gleiche Herkunft, Wichtigkeit und Spezifität */  
p[class="hervorgehoben"] { color: blue; }  
p.hervorgehoben { color: orange; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Titel meiner Web-Seite</title>  
    <meta charset="utf-8">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
  
  <body>  
    <p class="hervorgehoben">Dolor sit amet.</p>  
  </body>  
</html>
```

Dolor sit amet.

KASKADENREIHENFOLGE: ZUSAMMENFASSUNG

1. Herkunft und Wichtigkeit

- a. `!important`-Deklarationen in Benutzer-Stylesheets
- b. `!important`-Deklarationen in Autoren-Stylesheets
- c. Normale Deklarationen in Autoren-Stylesheets
- d. Normale Deklarationen in Benutzer-Stylesheets
- e. Normale Deklarationen im Browser-Stylesheet

2. Spezifität

3. Reihenfolge des Auftretens