



CODERS.BAY

CSS

>

- ▶ hat keine Funktion im Browser wie <a>
- ▶ hat keine Auswirkung auf die Textformatierung wie ein <p>
- ▶ verändert nicht das Aussehen des Textes der damit markiert ist
- ▶ funktioniert in allen Browsern
- ▶ hat keine semantische Bedeutung
- ▶ markiert den enthaltenen Text

```
<p>  
    Weit hinten, hinter den Wortbergen, fern der  
    <span>  
        Länder Vokalien und Konsonantien  
    </span>  
    leben die Blindtexte.  
</p>
```

INLINE STYLES UND FARBEN

- Inline Styling = schnellste und einfachste Methode, CSS in HTML zu bringen, ist auch die unpraktischste
- **color** und **background-color** sind zwei Style-Eigenschaften
 - ▶ **color** legt die Farbe der Schrift fest
 - ▶ **background-color** legt die Farbe des Hintergrunds fest

style-Attribute enthält die Formatangaben für das HTML-Element.

Style-Eigenschaften bestehen aus einem Namen, gefolgt von einem Doppelpunkt, einem Wert und schließt mit einem Strichpunkt

```
<p>  
    Weit hinten, hinter den Wortbergen, fern der  
    <span style="color: red; background-color: gray;">  
        Länder Vokalien und Konsonantien  
    </span>  
    leben die Blindtexte.  
</p>
```

INLINE STYLES UND FARBEN

- Inline Styling = schnellste und einfachste Methode, CSS in HTML zu bringen, ist auch die unpraktischste
- **color** und **background-color** sind zwei Style-Eigenschaften
 - ▶ **color** legt die Farbe der Schrift fest
 - ▶ **background-color** legt die Farbe des Hintergrunds fest

```
.custom-bg {  
  color: red;  
  background-color: gray;  
}
```

```
<p>  
  Weit hinten, hinter den Wortbergen, fern der  
  <span class="custom-bg">  
    Länder Vokalien und Konsonantien  
  </span>  
  leben die Blindtexte.  
</p>
```

- Style-Eigenschaften global zu definieren ist eine bessere Möglichkeit
- dort ändert man nur einmal den Wert
- gehört zum guten Stil die Dokumentenstruktur und den Inhalt von den Angaben zum Aussehen zu trennen
- Vorteil: einfacheres bearbeiten, bessere Wiederverwendbarkeit, Seiten werden barrierefreier
- Styles werden austauschbar
- zwei Möglichkeiten die im `<head>`-Bereich des HTML-Dokuments deklariert werden:
 - ▶ in der .html-Datei
 - > `<style>`
 - ▶ in einer eigenen .css-Datei
 - > `<link>`
 - ist kein Hyperlink
 - Attribut: `href`
 - kann eine absolute oder relative URL sein
 - Attribut: `rel="stylesheet"`
 - kurz für Relationship
 - gibt die Beziehung an

- `rel="shortcut icon"`
 - ▶ zeigt auf ein Bild mit iCO-Format
- `rel="alternate" type="application/rss+xml"`
`rel="alternate" type="application/atom+xml"`
 - ▶ verlinkt zum RSS- oder Atom-Feed der Seite

Stylesheet-Dateien enden typischerweise auf **.css**
Texteditoren mit Unterstützung für HTML helfen auch mit CSS weiter

- Selektor kommt vom englischen to select, auswählen
- Selektoren wählen Elemente aus
- Tag-Selektor funktioniert in allen Browsern

Den gesamten Block aus Selektor und einer oder mehreren Eigenschaften nennt man eine Style- oder CSS-Regel

ein einfacher Selektor für alle ``-Tags

```
span {  
  color: red;  
  background-color: gray;  
}
```

Und alle Eigenschaften, die zu einem Selektor gehören, werden in geschweiften Klammern gefasst.

Style Eigenschaften werden genauso definiert wie bei Inline Styles. Auch hier das Semikolon nicht vergessen!

- Alle Eigenschaften in einem Stylesheet müssen zu einem Selektor gehören, sie können nicht verweist im Stylesheet stehen.
- Indem man eine background-color für `<body>` setzt, legt man den Hintergrund der gesamten Seite fest

- **id**-Selektor funktioniert in allen Browsern

```
#zweiter-absatz {  
    color: red;  
}  
  
p {  
    color: green;  
}
```

```
<body>  
  <p>  
    Dieser Absatz ist grün.  
  </p>  
  <p id="zweiter-absatz">  
    Dieser Absatz wird rot.  
  </p>  
  <p>  
    Und dieser ist wieder grün.  
  </p>  
</body>
```

- Selektor kommt vom englischen to select, auswählen
- Selektoren wählen Elemente aus
- Ausgabebeispiel auf der nächsten Folie

```
.hobby {  
  color: red;  
}  
  
.work {  
  color: green;  
}  
  
.other {  
  color: orange;  
}
```

```
<body>  
  <p class="other">  
    Heute Picknick im Park &lt;3  
  </p>  
  <p class="hobby">  
    Seit langem mal wieder auf der PS4 gespielt. Macht Spaß.  
  </p>  
  <p class="work">  
    Drei Stunden nach einem Buch gesucht. Lag beim Kollegen unter dem  
    Schreibtisch.  
  </p>  
  <p class="hobby">  
    Ich kann immer noch nicht pokern, aber es macht Spaß.  
  </p>  
  <p class="work">  
    Mittwoch wird viel Stoff durchgenommen  
  </p>  
</body>
```

- Klassennamen, genau wie **ids**, dürfen nicht aus beliebigen Zeichen bestehen. Sie müssen mit einem Buchstaben anfangen und können danach Buchstaben, Ziffern, _ und - enthalten.
- **class** ist ein ganz einfaches HTML-Attribut



BEISPIEL AUSGABE



```
.hobby {  
  color: red;  
}  
  
.work {  
  color: green;  
}  
  
.other {  
  color: orange;  
}
```

Heute Picknick im Park <3

Seit langem mal wieder auf der PS4 gespielt. Macht Spaß.

Drei Stunden nach einem Buch gesucht. Lag beim Kollegen unter dem Schreibtisch.

Ich kann immer noch nicht pokern, aber es macht Spaß.

Mittwoch wird viel Stoff durchgenommen

- › Ein Element kann mehrere Klassen angehören
- › alle Klassen müssen im Attribut `class` angeführt werden und durch Leerzeichen getrennt sein

```
.hobby {  
  color: red;  
}  
.work {  
  color: green;  
}  
.night {  
  background-color: black;  
}
```

Seit langem mal wieder auf der PS4 gespielt. Macht Spaß.

Drei Stunden nach einem Buch gesucht. Lag beim Kollegen unter dem Schreibtisch.

Ich kann immer noch nicht pokern, aber es macht Spaß.

Mittwoch wird viel Stoff durchgenommen

```
<body>  
  <p class="hobby">  
    Seit langem mal wieder auf der PS4 gespielt. Macht Spaß.  
  </p>  
  <p class="work night">  
    Drei Stunden nach einem Buch gesucht. Lag beim Kollegen unter dem  
    Schreibtisch.  
  </p>  
  <p class="work">  
    Ich kann immer noch nicht pokern, aber es macht Spaß.  
  </p>  
  <p class="hobby night">  
    Mittwoch wird viel Stoff durchgenommen  
  </p>  
</body>
```

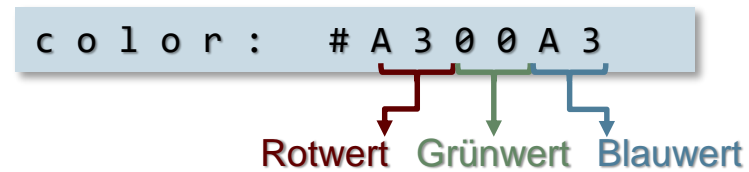
Klassen-, Tag- und id-Selektoren können kombiniert werden.

RGB-MODELL

- Je höher der Wert einer Farbkomponente, desto intensiver ist diese Komponente in der Farbe vertreten.
- Sind die drei Werte gleich, handelt es sich um einen Grauton, von **#000000** = Schwarz bis **#FFFFFF** = Weiß
- Hexadezimale Farben funktionieren in allen Browsern

color: # A 3 0 0 A 3

Rotwert Grünwert Blauwert



Info

<https://color.adobe.com/de>
www.colorpicker.com
Pick up Hexadezimal

> rgb()

- ▶ nimmt drei Funktionsparameter entgegen
- ▶ Rot, Grün, Blau
- ▶ Werte liegen zwischen 0 und 255 oder zwischen 0% und 100%

```
.blue { color: rgb(0, 0, 255); }  
.blue { color: rgb(0%, 0%, 100%); }
```

> rgba()

- ▶ vierter Wert ist der sogenannte Alpha-Wert
- ▶ darf zwischen 0 und 1 liegen
- ▶ bestimmt die Transparenz
- ▶ wird für background-color verwendet
- ▶ funktioniert im Internet Explorer ab Version 9

```
.blue { color: rgba(0, 0, 255, 0.5); }
```

> opacity

- ▶ gilt für Vordergrund und Hintergrund des Elements, es wird also transparent
- ▶ **rgba()** für **background-color** wirkt sich nicht auf die Vordergrundfarbe aus, nur auf die Hintergrundfarbe

```
.blue {  
  color: #0000FF;  
  opacity: 0.5;  
}
```

BACKGROUND-IMAGE

- funktioniert in allen Browsern
- der Wert von `background-image` ist eine URL zu einer Bilddatei
- die URL muss in Funktionsschreibweise `url()` angegeben werden
- es funktionieren relative und absolute URLs
 - ▶ relative URLs sind relativ zum Stylesheet, nicht zum HTML-Dokument

```
background-image: url("images/bildname.png");
```

› background-repeat

- ▶ default: **repeat**
 - › Bild wiederholt sich auf der horizontalen und vertikalen Achse
- ▶ **no-repeat**
 - › Bild wiederholt sich weder horizontal noch vertikal und wird nur einmal angezeigt
- ▶ **repeat-x**
 - › Bild wiederholt sich nur auf der horizontalen Ebene (x-Achse)
- ▶ **repeat-y**
 - › Bild wiederholt sich nur auf der vertikalen Ebene (y-Achse)
- ▶ **repeat-x** und **repeat-y** nützlich für Farbverläufe

```
background-repeat: no-repeat;
```

- mögliche Werte:
 - ▶ Prozentangaben
 - ▶ Pixelangaben
 - ▶ Schlüsselwörter
- nimmt zwei Werte an
 - ▶ erster Wert: Links-Rechts-Position
 - ▶ zweiter Wert: Oben-Unten-Position
- Schlüsselwörter
 - ▶ `left`
 - ▶ `right`
 - ▶ `center`
 - ▶ `top`
 - ▶ `bottom`

```
/* Beispiel */
.bg {
  background-image: url("images/bildname.png");
  background-repeat: repeat-y;
  background-color: '#8f9cff';
  background-position: center center;
  background-attachment: fixed;
}
```

- legt fest, woran das Hintergrundbild befestigt wird
 - ▶ an der HTML-Seite oder am Browserfenster
 - ▶ default: `scroll`
 - Bild scrollt mit
 - ▶ `fixed`
 - Bild bleibt fix im Hintergrund während der Content gescrollt werden kann

```
background-attachment: scroll;  
background-attachment: fixed;
```


➤ Beispiel

```
.bg {  
  background-image: url("images/bildname.png");  
  background-repeat: repeat-y;  
  background-color: '#8f9cff';  
  background-position: center center;  
  background-attachment: fixed;  
}
```



ÜBUNG...



Erstelle zwei neue HTML-Dokumente, die jeweils zueinander verlinken.
Nimm eine der beiden Seiten, und füge über dem Link drei Überschriften ein, eine `<h1>`, eine `<h2>` und eine `<h3>`.
Dann füge unter dem Link noch ein Bild ein.
Gib die CSS-Regeln an, die du schon kennst um einzelne Textbausteine unterschiedlich einzufärben.
Styling sollte einmal als Inline Styling angegeben werden, einmal mittels `<style>`-Tag im `<head>` Bereich und einmal über ein externes CSS File.
Benutze im CSS File Hexadezimal Angaben und rgb sowie rgba Angaben.
Die Überschrift `<h1>` sollte orange sein und die Quellenangabe in einer Farbe deiner Wahl hinterlegt sein.



CODERS.BAY

ENDE