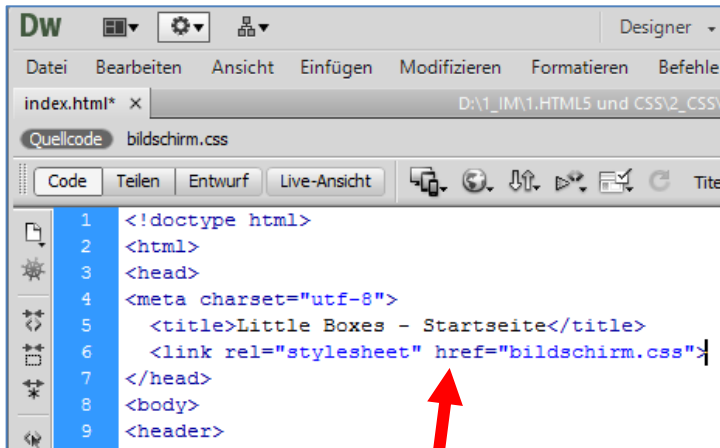


## Grundlagen-Beispiel CSS

Öffne im Ordner „2.css.uebung“ die vorhandenen Dateien „index.html“ und **erstelle** eine neue CSS-Datei mit dem Namen **„bildschirm.css“**.



```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Little Boxes - Startseite</title>
6 <link rel="stylesheet" href="bildschirm.css">
7 </head>
8 <body>
9 <header>
```

Füge im <head> Bereich folgende Verlinkung ein:

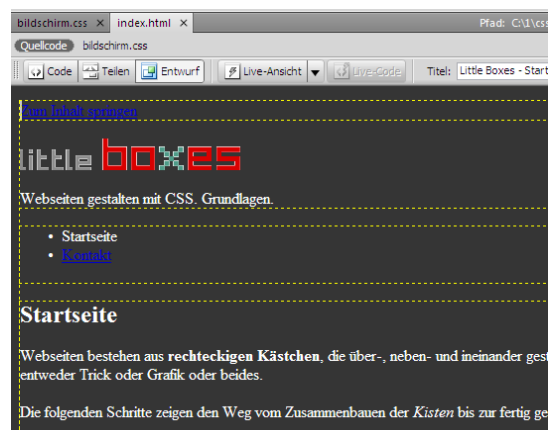
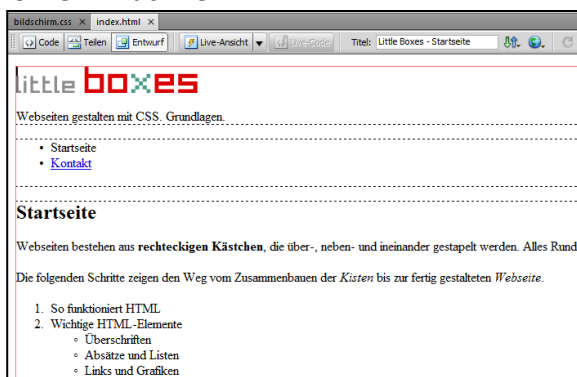
```
<link rel="stylesheet" href="bildschirm.css"/>
```

Nun werden in der „bildschirm.css“ zuerst die Hintergrundfarben und Schriftarten festgelegt. Der Hintergrund des kompletten <body> soll grau sein und die Schriftfarbe „weiß“.



```
1 /* Stylesheet der "Little Boxes" */
2
3
4 body {
5     background-color:#353535;
6     color:white;
7 }
8
```

vorher – nachher:



## Mit Klassen und IDs arbeiten

ba

### 1) id:

Im Quelltext der HTML-Datei können damit **eigene Namenszusätze** vergeben werden. Damit können die verschiedenen ID's einzeln in der CSS-Datei angesprochen und mit Eigenschaften versehen werden.

Vergib unter dem <body> eine ID namens „container“, der erst vor dem Ende des <body> wieder aufhört:

```
8 <body>
9 <div id="container">
10 <header>
11 <h1> <!-- Ende fussbereich -->
45 </div> <!--Ende Container-->
46 </body>
47 </html>
```

Dann in der CSS-Datei:

```
9 #container {
10     background-color:white;
11     color:black;
12 }
```

Da der Container innerhalb des <body> fast den kompletten Bereich ausmacht, ist die Veränderung massiv.

Aber: der Wrapper wird nur so hoch, wie der Inhalt der Website ist, darunter ist die schwarze Hintergrundfarbe des <body> erkennbar.

Ergebnis:



**2) „Class“ dient zur Gruppierung.** Damit können Mehrfachbenutzungen durchgeführt werden, da „class“-Namen öfter im HTML-Quelltext vergeben werden können (im Gegensatz zu ID's) und in der CSS-Datei damit gemeinsam angesprochen werden können.

### Beispiel:

Im Quelltext der HTML-Datei „index.html“ kommen mehrere `<p>` (Paragraphs) vor. So kann man mehrere mit „class“ versehen und dann gemeinsam „formatieren“.

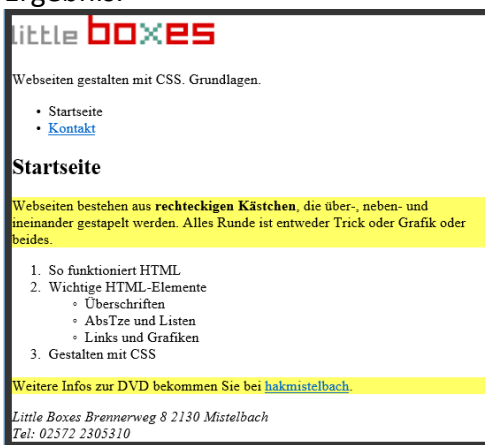
Dafür müssen die beiden kurz erweitert werden mit **class** und einem Namenszusatz. Dieser kann dann sehr einfach direkt in CSS angesprochen und mit Eigenschaften versehen werden. Anders bei ID's: hier müsste man unterschiedliche Namen wählen und daher beide extra in der CSS-Datei ansprechen – mehr Tipparbeit.

```
21 <article>
22 <h2>Startseite</h2>
23 <p class="infobox">Webseiten bestehen aus
  <strong>, die über-, neben- und ineinander
  ist entweder Trick oder Grafik oder bei
24 <ol>
25   <li>So funktioniert HTML</li>
26   <li>Wichtige HTML-Elemente
27     <ul>
28       <li>Überschriften</li>
29       <li>AbsTze und Listen</li>
30       <li>Links und Grafiken</li>
31     </ul>
32   </li>
33   <li>Gestalten mit CSS</li>
34 </ol>
35 <p class="infobox">Weitere Infos zur DVD
  "http://www.hakmistelbach.ac.at" title="
  hakmistelbach</a>.</p>
36 </article> <!-- Ende textbereich -->
```

In der CSS-Datei:

```
13
14 .infobox {
15     background-color:#FF6;
16 }
```

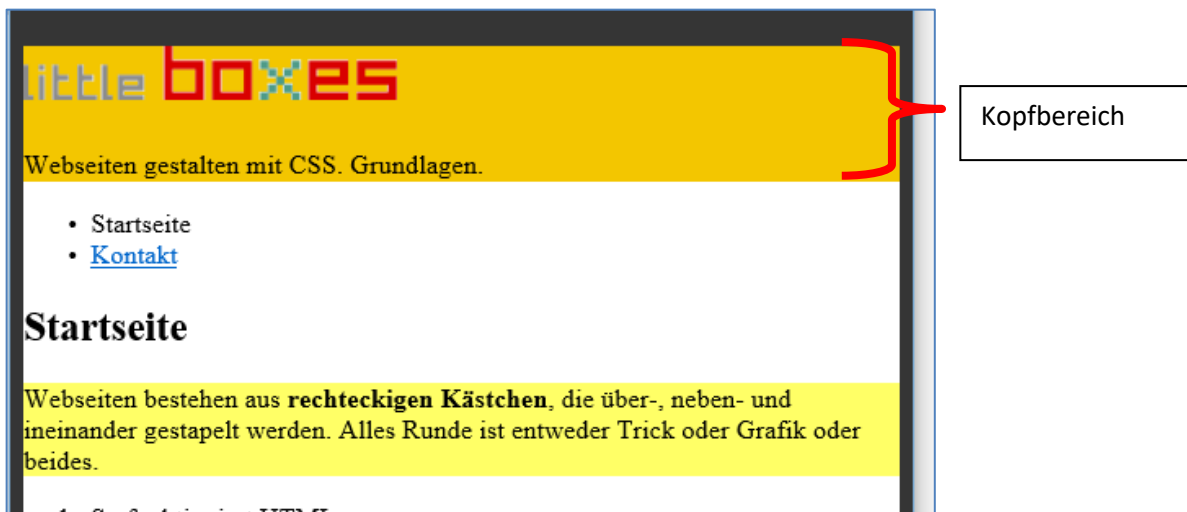
Ergebnis:



## Auf die semantischen Elemente zugreifen

Nun folgt die Formatierung des Kopfbereiches – es genügt den Namen des Bereiches, hier `<header>` in der CSS-Datei anzusprechen:  
Hintergrund gelb-orange, Schriftfarbe schwarz.

```
18 header {
19     background-color:#f3c600;
20     color:black;
21 }
```



## Logo formatieren

Da das Logo auch mit einer ID extra angesprochen werden kann, kann es auch separat formatiert werden.

```
8 <body>
9 <div id="container">
10 <header>
11 <h1></h1>
12 <p>Webseiten gestalten mit CSS. <span>Grundlagen</span>.</p>
13 </header> <!-- Ende kopfbereich -->
14 <nav>
15 </div>
```

Das Logobild „little boxes“ hat einen transparenten Hintergrund, der jetzt auch gelb-orange geworden ist. Dieser soll aber wieder weiß formatiert sein:

```
22 ,
23 #logo {
24     background-color:white;
25 }
```


Ergebnis:



## Einstellung der Schriftart und Schriftgröße:

Füge folgende Zeilen ein. Der %-Satz bei den Überschriften (z.B. h1, h2) bezieht sich auf die oben eingestellte Schriftgröße „small“.

```
4  body {
5      background-color:#353535;
6      color:white;
7      font-family:Verdana, Geneva, sans-serif;
8      font-size:small;
9  }
10
11 #container {
12     background-color:white;
13     color:black;
14 }
15
16 .infobox {
17     background-color:#FF6;
18 }
19
20 header {
21     background-color:#f3c600;
22     color:black;
23 }
24
25 #logo {
26     background-color:white;
27 }
28
29 h1 {font-size:150%; }
30 h2 {font-size:130%; }
31
```

The image shows a code editor with CSS code. Two red arrows point to the 'font-family' and 'font-size' properties in the 'body' selector. Another two red arrows point to the 'font-size' properties in the 'h1' and 'h2' selectors.

## Element „footer“ formatieren:

Soll in der Mitte stehen (center).

```
37  
38 <footer>  
39 Little Boxes Brennerweg 8 2130 Mistelbach  
40 <br>  
41 Tel: 02572 2305310  
42 </footer> <!-- Ende fussbereich -->  
43 </div> <!--Ende Container-->
```

Wähle die „Kiste“ footer“ aus und schreibe:

```
33 footer {  
34     text-align:center;  
35     font-size:80%;  
36     font-style:normal;  
37     letter-spacing:2px;  
38 }
```

Font-style: von italic (=kursiv) auf normal umstellen

letter-spacing: Abstand zwischen den Buchstaben wird auf 2 Pixel vergrößert.

Ergebnis:



## Hyperlinks gestalten

Hyperlinks werden in „Kisten“ mit **a** dargestellt, z.B.

```
<a href="kontakt.html" title="Zum Kontaktformular">Kontakt</a>
```

Unbesuchte Hyperlinks werden von den Browsern standardmäßig blau, besuchte Hyperlinks werden lila dargestellt.

**Pseudo-Klassen** helfen bei der Definition der Zustände von Hyperlinks:

- **a:link** = ein unbesuchter Hyperlink (kein Zwischenraum vor und nach dem Doppelpunkt)
- **a:visited** = besuchter Link
- **a:hover** = wenn sich die Maus darüber befindet
- **a:focus** = wenn der Hyperlink per Tab-Taste ausgewählt wurde
- **a:activ** = im Moment des Anklickens

Dabei sollte man sich an genau diese Reihenfolge der Pseudolinks halten!

**text-decoration: none** = entfernt die Hyperlink-Unterstreichung

**text-decoration: normal** = bringt wieder die Unterstreichung zurück

Als Alternative zur Unterstreichung kann man auch folgendes verwenden, welches den Vorteil hat, dass man mehr Einfluss nehmen kann und dass der Unterstrich etwas weiter entfernt vom Text ist:

**border-bottom: 1px solid #d90000** = solid (=durchgehend), Farbe beliebig

Gib folgende Styles ein:

```
40 a {text-decoration:none; }
41 a:link {color:#d90000; }
42 a:visited {color:#cc6600; }
43 a:hover, a:focus {border-bottom: 1px solid #d90000; }
44 a:active {
45     background-color:#d90000;
46     color:white; }
```

Ergebnis:



Styles können mit Beistrich getrennt kombiniert werden, Eigenschaften gelten dann für diese alle.

Wird gerade angeklickt, daher wird a:active verwendet.

## Verschachtelte Selektoren:

Dabei werden mehrerer Selektoren hintereinander geschrieben, z.B. soll nur im <article> die Hyperlink-Darstellung wieder anders lauten.

```
34 </ol>
35 <p class="infobox">Weitere Infos zur DVD bekommen Sie bei <a href=
"http://www.hakmistelbach.ac.at" title="Startseite unserer Schule">
hakmistelbach</a>.</p>
36 </article> <!-- Ende textbereich -->
37
```

Dazu wähle in der CSS-Datei den Selektor oder Bereich mit „article“ aus und wähle dann das **a** für die Hyperlinks, gefolgt von den Eigenschaften:

```
47
48 article a {
49     border-bottom:dotted;
50 }
51
```

Ergebnis:

Startseite

Webseiten bestehen aus **rechteckigen Kästchen**, die über-, ist entweder Trick oder Grafik oder beides.

1. So funktioniert HTML
2. Wichtige HTML-Elemente
  - Überschriften
  - AbsTze und Listen
  - Links und Grafiken
3. Gestalten mit CSS

Weitere Infos zur DVD bekommen Sie bei hakmistelbach.

Little Boxes Brennerweg 8  
Tel: 02572 23

gepunktet

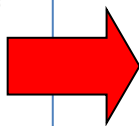


### Weiteres Beispiel für verschachtelte Selektoren:

Die Aufzählungszeichen in der Art von Kreisen sollen durch gefüllte Quadrate ersetzt werden. Alle ungeordneten Listen im Textbereich sollen ein quadratisches Symbol erhalten.

```
50 }
51 article ul li {
52     list-style-type:square;
53 }
```

1. So funktioniert HTML
  2. Wichtige HTML-Elemente
    - Überschriften
    - AbsTze und Listen
    - Links und Grafiken
  3. Gestalten mit CSS
- Weitere Infos zur DVD bekommen



1. So funktioniert HTML
  2. Wichtige HTML-Elemente
    - Überschriften
    - AbsTze und Listen
    - Links und Grafiken
  3. Gestalten mit CSS
- Weitere Infos zur DVD bekommen