



CODERS.BAY

# **SELEKTOREN & TYPOGRAPHIE**

## SCHRÖDINGER KAPITEL 04

- Cascading = stufenförmig hintereinander gereiht
- bedeutet: ein Element "erbt" sein Aussehen vom umgebenden Element  
das span im p erbt die Anweisungen

```
p { color: #707; background-color: yellow;}  
.farbe_1 { background-color: #000; }  
.farbe_2 { background-color: #456; }  
  
<p>  
  Gelb gelb gelb <span class="farbe_1">sind alle</span> meine Kleider...  
  Olivgrün <span class="farbe_2">sind alle</span> meine Kleider...  
</p>
```

Gelb gelb gelb sind alle meine Kleider... Olivgrün sind alle meine Kleider...

# KASKADIERENDE UND NICHT KASKADIERENDE ELEMENTE

kaskadiert	kaskadiert nicht
color	background und background-*
font und font-*	border und border-*
letter-spacing	display
list-style und list-style-*	height und width
text-align	margin und margin-*
text-indent	min- und max-height und -width
text-transform	overflow
visibility	padding und padding-*
white-space	position
word-space	top, bottom, left, right
	text-decoration
	vertical-align
	z-index

## ➤ Schriftgröße verändern

## ➤ Größenangaben

### ▶ absolute Größen

#### > **px** (Pixel)

- Größe eines Pixels ist nicht konstant

#### > **pt** (Point)

- passen nicht mit anderen Größenangaben zusammen da es eigentlich aus dem Druck kommt

### ▶ Schlüsselwörter

#### > **xx-small**

#### > **x-small**

#### > **small**

#### > **medium**

#### > **large**

#### > **x-large**

#### > **xx-large**

- Browser legt Größe fest – keine eigene Kontrolle

## ▶ relative Größen

### > **%** (Prozentangabe)

- verändert die Schrift gegenüber dem umgebenen Element

### > **em**

- ähnlich, wird als Multiplikator angegeben

### > **rem** (Root em)

- ähnlich wie em
- relativ zur Schriftgröße
- Angabe nicht relativ zum umgebenen Element

## › font-style:

- ▶ normal
  - ▶ italic (kursiv)
- 
- › Elemente, die direkt in einem anderen Enthalten sind, nennt man Kinder (children) des Elements.
  - › Alle Elemente, die in einem anderen Element enthalten sind, egal, wie tief verschachtelt, nennt man die Nachfahren (descendants) des Elements

```
p {  
  font-size: 1.2em;  
}  
h1 em, h2 em {  
  font-style: italic;  
}  
  
<h1>Die ist ein <em>titel</em></h1>  
<p>Das ist ein <em>Absatz</em></p>
```

`<em>` ist ein Tag, das den enthaltenen Text besonders hervorhebt. -> semantische Bedeutung

- Schriftarten, deren Namen Leerzeichen enthält, müssen immer in Anführungszeichen gesetzt werden  
`font-family: "Klingon New", "Comic Sans", Arial, sans-serif;`
- **sans-serif**
  - ▶ generische Schriftart
  - ▶ keine echte Schriftart
  - ▶ bezeichnet Gruppe von Schriftarten, die keine Serifen besitzen
- **cursive**
  - ▶ Schriften die einer Handschrift ähnlich sehen
  - ▶ dekorativer Nutzen
- **fantasy**
  - ▶ Schriften die dekorativ sind
  - ▶ dekorativer Nutzen
- **monospace**
  - ▶ Schriftarten, in denen alle Zeichen die gleiche Breite haben
  - ▶ für Fließtext
- **serif**
  - ▶ Schriftart mit Serifen
  - ▶ Serifen sind die "Füße", die Buchstaben haben
- Die Liste von Schriftarten für **font-family** sollten immer mit einem dieser fünf generischen Typen enden, sonst gibt es keine Garantie, was der Browser benutzt

- lädt eine Schriftart von einer URL und gibt ihr einen Namen, der anschließend in einer **font-family-Eigenschaft** benutzt werden kann
- Feature von CSS3 und wird allgemein als Webfonts bezeichnet
- **@font-face** ist kein Selektor
- Um sicherzugehen, dass in jedem Browser die richtige Schriftart verwendet wird, müssen mindestens zwei Formate zur Verfügung stehen: eot und otf

```
@font-face {  
  font-family: "Meine Beispielschrift";  
  src: url('examplefont.eot');  
  src: url('examplefont.woff') format("woff"),  
       url('examplefont.otf') format("opentype");  
}  
p {  
  font-family: "Meine Beispielschrift", "Times New Roman", serif;  
}
```

## > font-weight

- ▶ **normal**
- ▶ **bold** – fett gedruckt
- ▶ **100 – 700**: **400** ist normal, **700** ist **bold**, Rest funktioniert nur selten
- ▶ **bolder**, **lighter**: funktioniert nur selten

## > text-transform

- ▶ **uppercase**: stellt den gesamten Text des Elements nur in Großbuchstaben dar
- ▶ **lowercase**: alles in Kleinbuchstaben
- ▶ **capitalize**: macht den ersten Buchstaben jedes Worte zum Großbuchstaben
- ▶ **none**: schaltet alles ab

## > font-variant

- ▶ **normal**
- ▶ **small-caps**: Buchstaben werden kleiner dargestellt, verwendet Kapitälchen

## > text-decoration

- ▶ **underline**: unterstreichen
- ▶ **overline**: überstreichen
- ▶ **line-through**: durchstreichen

## > text-align

- ▶ **left**: linksbündig
- ▶ **right**: rechtsbündig
- ▶ **center**: zentriert
- ▶ **justify**: Blocksatz

## > text-indent

- ▶ setzt Einrückungen für die erste Zeile in jedem Absatz
- ▶ negative und positive Werte in **px**, **em**, **%**





<BR>



- fügt einen Zeilenumbruch ein bei z.B.: Fließtexten



➤ rechte Mouse Klick -> Element untersuchen

## ➤ Aufgabe 1

- ▶ Schrödinger lernt HTML, CSS und JS Kapitel 4 durcharbeiten (Beispiele machen, ggf. in Aufgabe 2 einfließen lassen)

## ➤ Aufgabe 2

Erstelle **mehrere** Seiten  
Die Seiten sollen beinhalten:

- ▶ Titel (steht dein Name)
- ▶ Verlinkungen zu anderen Seiten
- ▶ unterschiedliche Überschriften
- ▶ Mehrere Bilder
- ▶ Hintergrundbilder
- ▶ Fließtext mit Zeilenumbrüchen

Benutze für das Styling nur noch dein externes Stylesheet.  
Kein Inlinestyling mehr!!

Im externen Stylesheet soll:

- ▶ Min. eine Stylinganweisung soll sich nur auf Tags beziehen
- ▶ Min. eine Stylinganweisung soll sich nur auf IDs beziehen

- ▶ Min. eine Stylinganweisung soll sich nur auf Klassen beziehen
- ▶ Min. eine Stylinganweisung soll nach mehreren Klassen selektieren
- ▶ Min. eine Stylinganweisung soll nach kombinierten Klassen selektieren

Stylinganweisung die in **verschiedenen Kombinationen** verwendet werden müssen:

- ▶ CSS Anweisungen um die Hintergrundbild zu stylen
- ▶ Verschiedene Schriftfamilien (bediene dich hier bei Google Fonts)
- ▶ Verschiedene Schriftgrößen
- ▶ Verschiedene Formatierungen bei den Schriften

## ➤ Aufgabe 3

- ▶ weiterführende Links lesen und in Aufgabe 3 umsetzen

## Abgabemodalität:

Erstelle einen Ordner mit deinem Namen

In diesem Ordner erstelle einen zweiten Ordner mit der Bezeichnung „Aufgabe 03“.

Die .css Datei wird in einem eigenen Ordner namens styling im abzugebenden Ordner gespeichert.

Die Bilder werden in einem eigenen Ordner namens img im abzugebenden Ordner gespeichert.

In dem Ordner „Aufgabe 03“ sind alle Datei, Ordner und alle Bilder die du benutzt, abgespeichert.

Die Aufgabe wird Freitags mit einem USB-Stick eingesammelt. Auf dem USB-Stick kopierst du den Ordner mit deinem Namen.

Überprüfe die Datei am Stick ob alle Verlinkungen ordnungsgemäß funktionieren!



CODERS.BAY

ENDE