

ZUSAMMENFASSUNG CSS Grundkurs – Wichtige CSS Eigenschaften

EINFÜHRUNG

Link zur CSS-Eigenschaften-Übersichtsseite von w3schools:

<https://www.w3schools.com/cssref/>

FONTS

Generell ist es ratsam, für die eigene Website zu Beginn eine Schriftart festzulegen. Das kann über die HTML-Property font-family umgesetzt werden.

Beispiel:

```
* {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

Im Beispiel werden durch den *-Selektor alle HTML-Elemente selektiert und die gewünschte Schriftart Arial hinzugefügt. Helvetica wird automatisch hinzugefügt, falls die erst gewählte Schriftart aus jeglichen Gründen nicht geladen werden sollte. Falls keine der zuvor genannten Schriftarten geladen werden kann, wird eine Schriftart aus der Kategorie sans-serif gewählt.

WICHTIG

Schriftarten, die aus mehreren Wörtern bestehen, werden zwischen zwei einzelne Anführungszeichen geschrieben.

Beispiel:

```
* {  
  font-family: Georgia, 'Times New Roman', Times, serif;  
}
```

Serif vs. Sans Serif

- Serif: Kategorie der Serifenschriften -> **Hallo (Times New Roman)** -> verschnörkelt
- Sans Serif: Kategorie der serifenlosen Schriften -> **Hallo (Arial)** -> clean

font-size

Mit dieser Eigenschaft kann man die Schriftgröße anpassen.

Beispiel:

```
p {  
  font-size: 18px;  
}
```

font-weight

Mit dieser Eigenschaft wird der Text dicker gedruckt.

Beispiel:

```
header span {  
    font-weight: 900;  
}
```

font-style

Mit dieser Eigenschaft wird der Text kursiv gedruckt.

Beispiel:

```
header span {  
    font-weight: 900;  
    font-style: italic;  
}
```

MEHR FONTS: GOOGLE FONTS IMPLEMENTIEREN

Wenn man zusätzliche Schriftarten in seinem Projekt verwenden möchte, kann man diese beispielsweise mittels Google Fonts einbinden.

Link zu Google Fonts: <https://fonts.google.com/>

Bei Google Fonts ist es ratsam, immer mehr als einen Style einer Schriftart zu importieren. Zum Beispiel einen Regular-Style und einen Bold-Style. Das liegt daran, dass der Browser, in welchem die HTML-Datei angezeigt wird, sonst selbst versuchen müsste, die Schrift fett gedruckt bestmöglich darzustellen.

HINTERGRUND

background-color

Die Hintergrundfarbe kann via CSS über die background-color Eigenschaft gesetzt werden.

Beispiel:

```
div {  
    width: 800px;  
    height: 500px;  
    background-color: green;  
}
```

background-image

Über diese CSS-Eigenschaft kann ein Hintergrundbild gesetzt werden.

Beispiel:

```
background-image: url(/img/programmieren-starten-logo.png);
```

background-repeat

Mit dieser Eigenschaft kann man festlegen, dass sich ein Hintergrundbild wiederholt, falls es zu klein für den Hintergrundbereich sein sollte.

repeat-x

Das Bild wiederholt sich durch diesen Wert in x-Richtung.

Beispiel:

background-repeat: **repeat-x**;

repeat-y

Das Bild wiederholt sich durch diesen Wert in y-Richtung.

Beispiel:

background-repeat: **repeat-y**;

no-repeat

Das Bild wird nicht mehr wiederholt angezeigt.

Beispiel:

background-repeat: **no-repeat**;

background-size

Damit lässt sich die Größe des Hintergrundbildes anpassen.

Beispiel:

background-size: 1500px;

cover

Mithilfe dieses Wertes wird das Bild immer auf die Größe des Containers skaliert.

Beispiel:

background-size: **cover**;

background-position

Mithilfe dieser CSS-Eigenschaft kann man das skalierte Bild verschieben.

bottom

Durch diesen Wert wird das Bild so verschoben, dass man nur noch den unteren Teil sieht.

Beispiel:

background-position: **bottom**;

center

Möchte man die Mitte des Bildes sehen, wählt man den Wert center.

Beispiel:

background-position: **center**;

Verschiedene Werte können auch miteinander **kombiniert** werden.

Beispiel:

background-position: **bottom right**;

background-attachment

Mit der CSS-Eigenschaft background-attachment kann festgelegt werden, wie der Hintergrund mit dem eigentlichen Inhalt zusammen angezeigt werden soll. Der default-Wert heißt „scroll“.

scroll

Beispiel:

background-attachment: **scroll**;

Mit diesem Wert verschwindet das Bild, sobald man die Seite entlang nach unten scrollt.

fixed

Beispiel:

background-attachment: **fixed**;

Durch diesen Wert wird das Bild im Hintergrund an der aktuellen Position fixiert. Wenn man nun also scrollt, verschwindet das Bild nicht, sondern bleibt immer an der gleichen Stelle sichtbar.

RÄNDER

Mithilfe von CSS kann man sämtliche Ränder von HTML-Elementen nach Belieben modifizieren.

border-width

Mit dieser CSS-Eigenschaft definiert man, wie dick in Pixel der Rand sein soll.

Beispiel:

```
<style>
  div {
    border-width: 3px;
  }
</style>
```

border-color

Mit dieser CSS-Eigenschaft legt man die Farbe des Randes fest.

Beispiel:

border-color: black;

border-style

Mit dieser CSS-Eigenschaft kann man die Art des Randes festlegen.

Beispiel:

border-style: solid;

dashed

Wenn man der Eigenschaft border-style den Wert dashed zuweist, wird der Rand gestrichelt dargestellt.

Beispiel:

border-style: **dashed**;

dotted

Wenn man der Eigenschaft `border-style` den Wert `dotted` zuweist, wird der Rand gepunktet dargestellt.

Beispiel:

`border-style: dotted;`

Es gibt eine abgekürzte Schreibweise, in der man alle drei CSS-Eigenschaften in einer CSS-Regel setzen kann:

```
div {  
    border-width: 2px;  
    border-color: orange;  
    border-style: dotted;  
}  
p {  
    border: 3px solid green;  
}
```

border-radius

Mit dieser CSS-Eigenschaft lassen sich die Ecken der Ränder abrunden.

Beispiel:

```
div {  
    border-width: 2px;  
    border-color: orange;  
    border-style: dotted;  
    border-radius: 10px;  
}
```

border-bottom-right-radius

Damit lässt sich beispielsweise nur die rechte untere Ecke des Randes abrunden.

Beispiel:

```
p {  
    border: 3px solid green;  
    border-bottom-right-radius: 10px;  
}
```

border-bottom

Im folgenden Beispiel wird nur unter der h1-Überschrift ein Rand gesetzt:

```
h1 {  
    border-bottom: 3px solid red;  
}
```

FARBEN SPEZIFIZIEREN

Über RGB-Werte und Hexadezimal-Farbcodes können verschiedene Farbtöne spezifiziert werden.

RGB-Werte

Ein RGB-Wert setzt sich aus den drei Grundfarben Rot, Grün und Blau zusammen. Aus diesen Farben kann jede beliebige Farbe gemischt werden. Die Stärke jeder dieser drei Farben kann man mit einer Ziffer zwischen 0 und 255 bestimmen. 0 ist der kleinste Wert, während 255 der größte Wert ist.

Mit der ersten Zahl gibt man den Rot-Anteil an der Gesamtfarbe an, mit der zweiten Zahl den Grün-Anteil und mit der dritten Zahl gibt man den Blau-Anteil an der Gesamtfarbe an.

Beispiel:

background-color: **rgb(0,0,0);**

Der Hintergrund wird durch dieses Beispiel schwarz dargestellt.

Beispiel 2:

background-color: **rgb(100,120,80);**

Der Hintergrund wird durch dieses Beispiel in einen Olivgrünnton gefärbt.

Link zu einer Website mit Farbcodes: <https://www.color-hex.com/>

Hexadezimal-Farbcode

Die Werte von Hexadezimal-Farbcodes haben zu Beginn ein #-Zeichen und sind in Hexadezimalzahlen angegeben.

Die ersten beiden Zeichen stehen für den Rot-Anteil der Gesamtfarbe, die mittleren beiden Zeichen für den Grün-Anteil und die letzten beiden Zeichen stehen für den Blau-Anteil an der Gesamtfarbe. Wie auch beim RGB-Wert kann beim Hex-Wert jeder Farbanteil zwischen 0 und 255 liegen.

Der Wert 0 entspricht hier 00 und der Wert 255 entspricht dem Wert ff.

TIPP: GOOGLE CHROME DEVELOPER TOOLS

Shortcut, um Google Chrome Developer Tools zu öffnen:

Mac: **Command+Option+I**

Windows/Linux: **F12 oder Control+Shift+I**

BOX MODEL, MARGIN UND PADDING

Box Model

Das Box Model bildet die Grundlage eines Layouts. Jedes Block-Element (p, div, h1, ...) bildet eine rechteckige Box, die frei formatiert werden kann. Das Element an sich, dessen Rand und die Abstände via Padding und Margin ergeben gemeinsam das Konzept des Box Models. Mithilfe von Google Chrome Developer Tools kann das Box Model besser verstanden werden.

Margin

Mit Margin legt man den Abstand vom **Rand** des Elements zu **anderen Elementen** fest, die um das Element herum angeordnet sind.

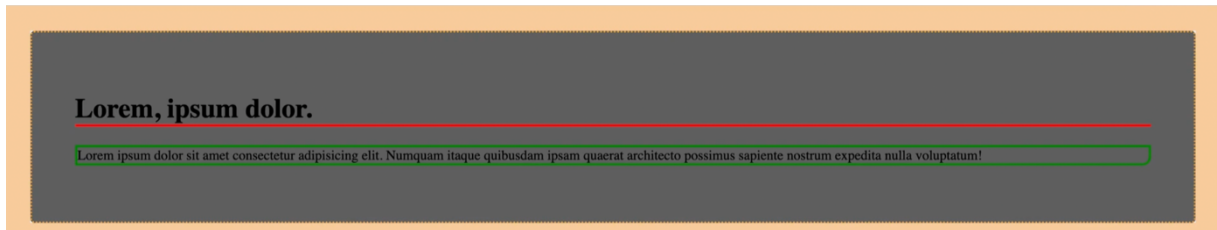
Beispiel 1:

margin: 30px;

Beispiel 2:

margin-top: 30px;

Auf der Abbildung ist die Margin in einem hellen Orange dargestellt:



Padding

Mit Padding legt man den Abstand zwischen dem **Element-Inhalt** und dem **Element-Rand** fest.

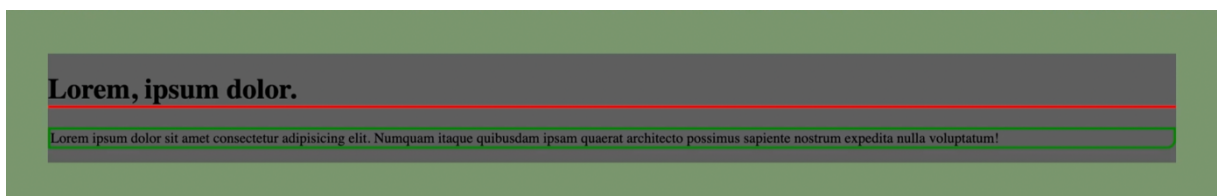
Beispiel 1:

padding: 50px;

Beispiel 2:

padding-left: 20px;

Auf der Abbildung ist das Padding grün dargestellt:



Alle Werte in **einer CSS-Regel** definieren:

```
p {  
    border: 3px solid green;  
    border-bottom-right-radius: 10px;  
    padding: 10px 20px 30px 40px;  
}
```

1. Wert: 10 Pixel (padding-top)
2. Wert: 20 Pixel (padding-right)
3. Wert: 30 Pixel (padding-bottom)
4. Wert: 40 Pixel (padding-left)

DIE CSS-EIGENSCHAFT BOX-SIZING

Standardmäßig vergrößern Padding und die border-width einen bestehenden Container sowohl in der Höhe als auch in der Breite. Das liegt daran, dass die CSS-Eigenschaft **box-sizing** standardmäßig auf den Wert **content-box** gesetzt ist.

border-box

Der Wert border-box sorgt dafür, dass die festgelegte Höhe und Breite des divs dennoch eingehalten wird, denn sowohl das Padding als auch ein potenzieller Rand werden mit eingerechnet. Stattdessen wird der Content-Bereich innen kleiner.

WICHTIG

Wenn man die CSS-Eigenschaft box-sizing nicht explizit setzt, dann ist diese standardmäßig immer auf den Wert **content-box** gesetzt. Das bedeutet, dass sich die width- und height-Attribute nur auf den Inhalt des divs beziehen. Fügt man noch ein Padding oder einen Rand hinzu, werden die gesamte width und height des divs dadurch nochmals vergrößert.

EXKURS: PSEUDO-KLASSEN (AM BEISPIEL VON LINKS ERKLÄRT)

Eine Pseudo-Klasse wird verwendet, um ein Element anzusprechen, wenn es eine bestimmte Eigenschaft bzw. einen bestimmten Status besitzt. Man kann Elemente also je nachdem, welchen Status sie gerade aufweisen, unterschiedlich stylen.

Die Pseudoklasse hover

Im folgenden Beispiel wird festgelegt, dass sich der Text rot färben soll, wenn man mit der Maus über einen Link hovers/fährt.

Beispiel:

```
a:hover {  
    color: red;  
    text-decoration: none;  
}
```

Die Pseudoklasse active

Im folgenden Beispiel wird festgelegt, dass sich der Text vergrößert, solange man auf den Link klickt.

Beispiel:

```
a:active {  
    font-size: 50px;  
}
```

Die Pseudoklasse link

Im folgenden Beispiel werden durch die Pseudoklasse link alle Links selektiert, die noch **nicht besucht** wurden.

Beispiel:

```
a:link {  
    color: black;  
}
```

Die Pseudoklasse visited

Im folgenden Beispiel werden mithilfe der Pseudoklasse visited alle Links selektiert und in grün dargestellt, die **bereits besucht** wurden.

Beispiel:

```
a:visited {  
    color: green;  
}
```

TABELLEN STYLEN

border-collapse

Die Eigenschaft border-collapse legt fest, ob Rahmen benachbarter Tabellenelemente getrennt oder zusammenfallend dargestellt werden.

Durch das folgende Beispiel wird die Tabelle mit einer einfachen Umrandung dargestellt.

Beispiel:

```
.custom-table {  
    border-collapse: collapse;  
}
```

width

Durch die Eigenschaft width wird die Breite eines Elements festgelegt. Im folgenden Beispiel wird festgelegt, dass die Tabelle eine Breite von 90 % der verfügbaren Fläche einnehmen soll.

Beispiel:

```
.custom-table {  
    border-collapse: collapse;  
    width: 90%;  
}
```

text-align

Durch diese Eigenschaft wird die horizontale Ausrichtung von Text und inline-Elementen in Blockelementen festgelegt.

Im folgenden Beispiel wird der Text innerhalb der Kopfzeile linksbündig ausgerichtet.

Beispiel:

```
.custom-table thead tr {  
    text-align: left;  
}
```

Die Pseudoklasse nth-child

Der :nth-child-Selektor wählt das n-te Element zu einem Elternelement aus. Im folgenden Beispiel soll jede gerade Zahl selektiert und damit jede Zeile mit einer geraden Zeilennummer in der angegebenen Farbe dargestellt werden.

Beispiel:

```
.custom-table tbody tr:nth-child(even) {  
    background-color: rgb(236, 238, 248);  
}
```

Im zweiten Beispiel werden durch „odd“ nur die ungeraden Zahlen selektiert:

Beispiel 2:

```
.custom-table tbody tr:nth-child(odd) {  
    background-color: rgb(236, 238, 248);  
}
```

Die Pseudoklasse first-of-type

Mithilfe dieser Pseudoklasse wird im folgenden Beispiel das erste th-Element, das auftritt, selektiert und ein Radius wird darauf angewendet.

Beispiel:

```
.custom-table th:first-of-type {  
    border-top-left-radius: 10px;  
}
```

DISPLAY: INLINE, BLOCK, INLINE-BLOCK UND NONE

- Der Display-Value von divs ist standardmäßig auf block gesetzt.
- Der Display-Value von spans ist standardmäßig auf inline gesetzt.

display

Trotz alldem ist es möglich, diesen Wert über die CSS-Eigenschaft display zu verändern.

Beispiel:

```
#span-1 {  
    display: block;  
    background-color: green;  
}
```

Bilder sind standardmäßig auf den Wert inline-block gesetzt.

Unterschied zwischen inline und inline-block

Bei inline-block-Elementen kann man die Breite (width) und Höhe (height) manuell festlegen. Bei Elementen mit dem Display-Value inline hingegen funktioniert das nicht.

none

Durch den Display-Value „none“ verhält sich das Element so, als wäre es vollständig aus dem HTML-Element gelöscht worden.

Beispiel:

```
#div-3 {  
    display: none;  
}
```