

CSS 3 - Zusammenfassung

1. Das Boxensystem

- Die Organisation der Boxen einer Website ist im index.php festgelegt.
- Die Eigenheiten der einzelnen Boxen sind in der CSS – Datei festgelegt.
- Der Inhalt einer Box wird mittels PHP-Einschüben eingefügt.

2. PHP-Einschübe

In der Datei index.php befindet sich das Template mit dem die verschiedenen Boxen und PHP-Einschübe zusammengeknüpft werden. Im Prinzip sind alle Einschübe freiwillig. Wenn z.B. kein Menü gewünscht ist, kann der {\$menu} – Einschub weggelassen werden, etc. Beispiel der PHP-Einschübe von CMBasic:

{ \$title }	steuert den Seitentitel
{ \$menu }	erzeugt das Hauptmenu
{ \$content }	Platzhalter für den Inhaltsbereich
{ \$topmenu }	erzeugt das obere Menü, falls gewünscht
{ \$path }	erzeugt den Navigationspfad, falls gewünscht
{ \$felog }	generiert den ebenfalls freiwilligen Frontend-Einlogbereich
{ \$meta }	wird unterhalb des TITLE-Tags notiert und bindet die Meta-Description ein
{ \$head }	wird oberhalb des HEAD-Ausschalt-Tags notiert und bindet zusätzliche CSS- bzw. RSS-Links ein
{ \$rssicon }	wird an beliebiger Stelle im BODY notiert und blendet ggf. ein RSS-Symbol mit Feedlink ein

3. CSS-Datei für Screen-Betrieb

- Die normale CSS-Datei wird bei der Anzeige im Browser verwendet. Sie wird gegenüber der Print CSS-Datei mittels Medien-Eintrag „screen“ in der Head-Zeile unterschieden:

```
<link rel="stylesheet" href="normal.css" media="screen">
```

- In der normalen CSS – Datei sind die Eigenheiten der einzelnen Boxen festgelegt.
- Margin (Aussenrand) wenn möglich auf 0 lassen, da sie die Boxen vergrößern
- Padding (Innenrand) nur wo nötig verwenden, da sie die Boxen verkleinern!

4. CSS-Datei für Printer-Betrieb

- Die Print-CSS-Datei wird beim Ausdrucken einer Seite verwendet wird. Sie wird gegenüber der normalen CSS-Datei mittels Medien-Eintrag „print“ in der Head-Zeile unterschieden:

```
<link rel="stylesheet" href="print.css" media="print">
```

- **Alle Boxinformationen** der normalen CSS-Datei sind hier **weggelassen**.
- Schriftfarbe ist auf schwarz und **# navigation** auf „none“.

5. Quirks-Modus

Der Quirks-Modus ist der unkorrigierte Browser-Modus mit all seinen CSS-Fehlern. Am Schlimmsten wirkt sich der Quirks-Modus des Internet Explorer aus. Damit ein Browser nicht in diesen Modus fällt, müssen die HTML-Seiten wie folgt eröffnet werden:

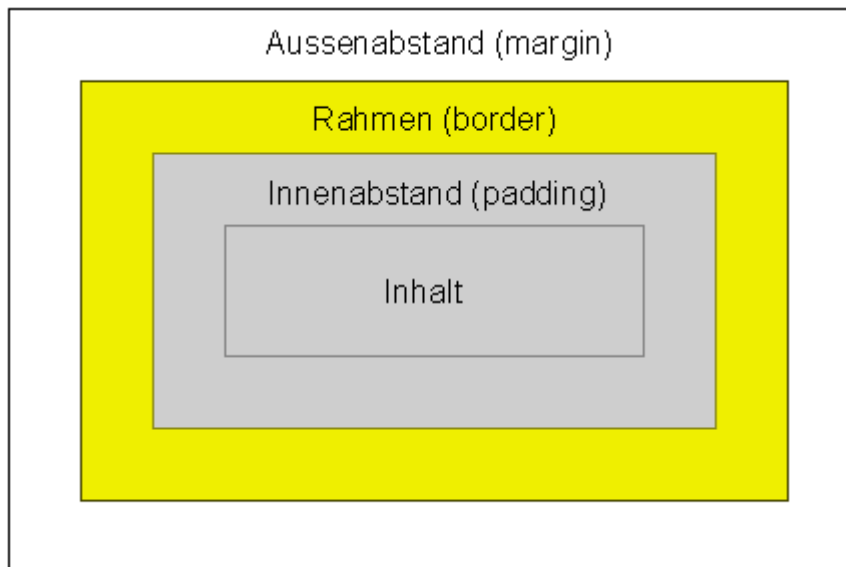
Bis html 4:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Etwelche Eröffnungen mit `<?xml version="1.0"?>` weglassen.

Ab html 5: `<!DOCTYPE html>`

6. Das Boxen-Prinzip



Margin (Aussenabstand) löst den Kistenrand vom Boxenrand und macht den eine Ebene tiefer liegenden Hintergrund sichtbar. Margin ist für alle Seiten oder für einzelne Seiten (top, right, bottom, left) einstellbar. Margin: auto mittet Box im vorliegenden Feld ein

Border (Rahmen) zeigt den Kistenrand an. Die Randbreite geht zulasten der Innengröße. Ist ähnlich wie padding, nur wird der Border mit einer Farbe gefüllt. Border ist für alle Seiten oder für einzelne Seiten (top, right, bottom, left) einstellbar.

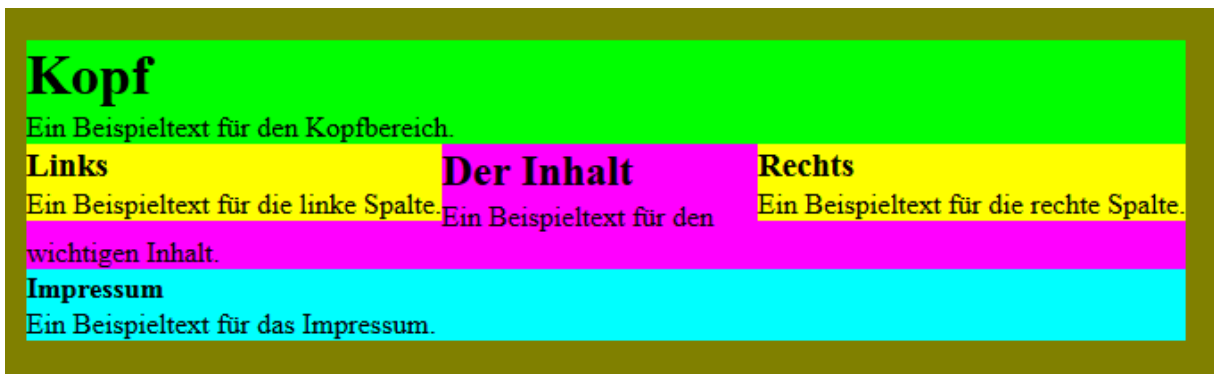
Padding (Innenabstand) löst klebende Texte vom Rand und macht Hintergrund auf gleicher Ebene sichtbar. Padding ist für alle Seiten oder für einzelne Seiten (top, right, bottom, left) einstellbar.



Bemerkung: Kiste ist Box ohne Margin, Padding und Rand.

7. Floating Boxes

Mittels Floating Boxes lassen sich einzelne Boxen an den rechten bzw. linken Rand verschieben. Der Raum dazwischen wird von unten her aufgefüllt.



```
#rechts {
    float: right;
}

#neutral {
    clear: both;
}
```

- Floating Boxen werden im index.php immer zuerst aufgeführt; die daneben liegenden Boxen erst anschliessend (Grund: sonst stimmt der obere Rand nicht!) und ohne floating-Angabe aber dafür mit margin-Einträgen.
- Nach floating-Boxen kommt immer ein Abschluss mit `clear: both` in der gleichen Box.
- Mehrere linke bzw. rechte Boxen übereinander können nur durch Unterzeilen- bzw. Abschnitte in der gefloateten Box gemacht werden.

8. Boxen-Gestaltung

8.1 Ränder

```
#Kiste    {
    margin:.....
    padding:.....
    border width: 6px;
    border-style: solid;
    border-color: white;
}
```

8.2 Schattenränder

```
#Kiste    {
    margin:.....
    padding:.....
    box-shadow: 0 1px 2px rgba(0,0,0,25);
    -webkit-box-shadow: 0 1px 2px rgba(0,0,0,25);
    -moz-box-shadow: 0 1px 2px rgba(0,0,0,25);
}
```

Die grün markierten Zeilen sind Anpassungen für ältere Browser.

8.3 Runde Ecken

```
#Kiste    {
    margin:.....
    padding:.....
    border-top-radius
    webkit-border-top-left-radius: 6px
    -moz-border-radius top-left: 6px;;
    border-top-left-radius: 6px;
    webkit-border-top-right-radius: 6px
    -moz-border-radius top-right: 6px;;
    border-top-right-radius: 6px;
}
```

Die grün markierten Zeilen sind Anpassungen für ältere Browser.

8.4 Hintergrund-Farbe

```
#Kiste    {
    margin:.....;
    padding:.....;
    background-color: .....;
}
```

8.5 Hintergrund-Farbverlauf

```
#Kiste    {
    margin:.....;
    padding:.....;
    background-image: url (z.B. schmaler 10 – 20px-Verlaufsstreifen von oben nach
    unten);
    background-repeat: repeat-x (horizontal vervielfachen) bzw. repeat-y (vertikal
    vervielfachen);
}
```

8.6 Hintergrund-Bild

```
#Kiste    {
    margin:.....;
    padding:.....;
    background-color: .....;
    background-image: url (z.B. Wasserzeichen.gif); *)
    repeat: no repeat; ***)
    background-attachment: fixed; **)
}
```

*) Die Eigenschaft background-image wiederholt das eingebundene Bild horizontal und vertikal und überdeckt die background-color. Es geht mit gif- und jpg-Bildern.

**) Fixiert das Hintergrundbild, so dass Vordergrund darüber läuft (kann weggelassen werden).

***) Verhindert das Wiederholen (ist bei kurzen Seiten nicht unbedingt nötig)

8.7 Zusammengefasste Schreibweise für Boxen

```
#Kiste    {
    background: #CFFFCDD url (wolke.gif) no repeat 50px 20px fixed
}
```

*) Die Eigenschaft background-image wiederholt das eingebundene Bild horizontal und vertikal und überdeckt die background-color.

**) Fixiert das Hintergrundbild, so dass Vordergrund darüber läuft (kann weggelassen werden).

8.8 Boxen Positionierung

```
#Kiste_links {
    margin:.....;
    padding:.....;
    Background-color: .....;
    float: left;
}
```

```
#Kiste_rechts {
    margin:.....;
    padding:.....;
    Background-color: .....;
    float: right;
}
```

```
#Kiste_neutral {
    margin:.....;
    padding:.....;
    Background-color: .....;
    clear: both; *)
}
```

*) nur nötig, wenn vorher rechts oder links gefloatet wurde.

8.9 Überlappung von DIVs

Überlappende DIVs müssen mit einem z-index versehen werden (z.B. DIV 1 mit Angabe z-index:-1 und DIV 2 mit Angabe z-index:1). Es wird diejenige mit der höheren Boxangabe (z-index:1) angezeigt.

```
#Kiste {
    .....
    .....
    z-index:1
}
```

9. Schrift-Gestaltung

9.1 Schrift-Familie

Es gibt folgende Schriftfamilien: serif, sans serif, cursive, monospace, fantasy. Beispiel für serif:

```
#serif {
    font-family: serif;
}
```

9.2 Schriftart

Es gibt zu jeder Schrift-Familie beinahe unendlich viele Schriftarten. Hier einige Beispiele: Arial, Verdana, Courier New, Monotype Corsiva, etc. Beispiel für sans-serif::

```
#sans-serif {
    font-family: Arial, Verdana, sans-serif;
}
```

9.3 Schrift-Grösse

Es wird unterschieden in relative (em) und absolute (px) Grössen. 1 em ist die Höhe des grossen „M“ der aktuellen Schriftart. 1 px ist ein Bildpunkt des Anzeigegerätes, relativ zur Bildschirmauflösung.

```
#div1 {
    font-size: 2em;
}
```

Die Schriftgrösse kann auch vererbt werden vom Elternelement (übergeordnete Box)

```
#div1 {
```

```
    font-size: inherit;
}
```

Bemerkung: Dieses Inherit wird vom Browser automatisch verwendet, sollte aber aus Übersichtlichkeitsgründen verwendet werden.

9.4 Text-Eigenschaften

Dem Text können die verschiedensten Eigenschaften (Schreibrichtung, Ausrichtung, Zeilenabstand, Umbruch, Wortabstand, Farbe, Unterstreichen, etc.) zu Grunde gelegt werden. Beispiel für Textklassen „Haupttext“ und „Textabsatz“:

```
.haupttext {
    font-family: Arial, sans-serif;
    font-style: italic;
    font-size: 1em;
    color: black;
}

.p {
    font-size: 12px;
    font-style: normal;
    color: blue;
}
```

9.5 Zusammenfassung mit Eigenschaft „Font“

Mit der Eigenschaft “font“ können in folgender Reihenfolge alle Schrifteigenschaften zusammengefasst werden:

font: font-style font-variant font-weight *font-size font-family*

Dabei müssen im Minimum die Kursiv angegebene Eigenschaften vorhanden sein!

Hier ein ausgereiftes Beispiel:

```
#schrift {
    font: normal, small-caps lighter 1.2em “Verdana“, “Trebuchet MS“, “Avant-
garde“,
    “Arial“, “sans-serif“;
}
```

9.6 Schriftschatten

```
.text-shadow {
    1px 1px 0 #E6E6E6;
}
```

Erzeugt Schatten der Farbe #E6E6E6 zu einer Schrift

9.7 Textfeld bei Mouse over

<p title=“Hier steht der anzuzeigende Text“>

9.8 Hyperlinks

```
.a {
    text-decoration: none;
    color: inherit;
    background-color: yellow;
    font-style: italic;
}
```

9.9 Hyperlinks mit Pseudoklassen

Bei den Pseudoklassen ist die **Reihenfolge wichtig** und sollte unbedingt eingehalten werden!

```
.a:link {
    text-decoration: none;
    color: inherit;
    background-color: yellow;
    font-style: italic;
}

.a:visited {
    text-decoration: none;
    color: inherit;
    background-color: maroon;
    font-style: italic;
}

.a:hover, a:focus {
    text-decoration: none;
    color: inherit;
    background-color: red;
    font-style: italic;
    outline: none;
}

.a:active {
    text-decoration: none;
    color: inherit;
    background-color: white;
    font-style: italic;
    outline: none;
}
```

Beispiel Linkanzeige:

```
# menu a:link {.....}    Linkanzeige in Ruhezustand
# menu a:hover {.....}   Linkanzeige bei Darüberfahren mit Maus
```


10. HTML-Tag-Formatierung

10.1 Verwendete CSS-Instruktionen:

Mit CSS-Instruktionen lassen sich die Eigenheiten von HTML-Tags einheitlich festlegen:

z.B. Elementsselektor h1: <h1> ... </h1>
zugehörige CSS-Instruktion: h1 {color:red};

Sollte die Einheitsregelung für einzelne Fälle umgangen werden, so erfolgt dies mit dem Beifügen von Klassen-Instruktionen

z.B. Elementsselektor h1: <h1 class="grün"> ... </h1>
zugehörige CSS-Instruktion: .grün = {color:green};

Einmalige Festlegungen für bestimmte Tags lassen sich mit ID-Instruktionen spezifizieren

z.B. Blockselektoren <p>, <div>, <h> : <p id="abc"> ... </p>
zugehörige CSS-Instruktion: #abc {color:black};

Blockelemente wie <h>, <p>, <div> erzeugen um sich herum automatische einen rechteckigen Rahmen und lassen sich wie Boxen verändern. Sie besitzen alle eine Zeilenumbruchfunktion.

Inline-Style Elemente wie <h1 style=".....">, ... werden immer von unsortierten Listen-Elemente (unordered list) wie ... umschlossen. Sie können bezüglich Hintergrund auch unterschiedlich formatiert sein.

Regelung bei mehreren Instruktionen:

- > Je näher eine CSS-Instruktion bei einem Element liegt, umso höher ist deren Gewichtung
- > Stehen mehrere Instruktionen hintereinander , so wirkt immer die letzte

10.2 Syntax zur Formatierung eines bestimmten HTML-Elementes

Die Syntax lautet: „**Box-ID**“ Lücke „**HTML-Tag**“ Lücke „**eingebettetes HTML-Tag**“

Das Festlegen der CSS-Instruktion für ein ganz bestimmtes HTML-Element erfolgt durch hierarchische Aneinanderreihung der relevanten Tags der Box.

#content p {...};	-> für p-Block in Content-Box
#main a {...};	-> für Links in Content-Block
#menu ul li {...};	-> für Menüeintrag in Menü-Block
#menu ul ul li {...};	-> für Untermenüeintrag in Menü-Block

11. Vererbung

Das wichtigste Prinzip der Vererbung ist: “von oben nach unten“. Definiere alles möglichst weit oben (ab dem body- oder container-Element) und du wirst dank der Vererbung eine einheitliche Webseite erhalten.

Globale Einstellungen werden im body-Bereich definiert. Diese Einstellungen betreffen besonders Schriften, Texteigenschaften und Farben. Je nach Sichtweise ergeben sich verschiedene Zusammenhänge.

Boxeigenschaften werden nicht vererbt.

Texteigenschaften (ausg. text-decoration & text-shadow) werden in untergeordnete DIVs (Kinderelemente) vererbt, wenn nichts anderes angegeben ist. Die Verwendung von „inherit“ schafft Übersicht.

Bei der Textvererbung gelten die Regeln der grossen Kaskaden von oben nach unten, welche aus folgenden Elementen besteht:

1. Kleine Kaskade (sammelt, gewichtet, spezifiziert, beachtet Reihenfolge)
2. Vererbung (Unterscheidet ob vererbbar oder nicht vererbbar, Erbt vom Elternteil mittels „inherit“,)
3. setzt Standartwerte des Browsers (wenn keine Regelung gefunden wird)

12. Gestaltungsregeln:

Die Gestaltung erfolgt normalerweise nach folgendem prinzipiellen Ablauf:

1. Anordnung (platzieren) der Box (nur aus Boxensicht)
2. Gestaltung der Box (Rand, Abstände, Füllung)
3. Gestaltung Inhalt der Box (Schriftart, Schriftgrösse, Schriftfarbe)

13. CSS-Medienabfragen (Media Queries / responsive Medienabfrage)

Unter **HTML 5** können CSS-Medienabfragen eingesetzt werden. Diese Medienabfragen sind einfache Filter, die auf CSS-Stile angewendet werden können. Sie ermöglichen, Stile ganz einfach auf Grundlage von Charakteristiken des Geräts, das zur Anzeige der Inhalte genutzt wird, zu ändern, einschließlich Anzeigetyp, Breite, Höhe, Ausrichtung und sogar Auflösung.

Beispielsweise können alle Stile, die zum Drucken notwendig sind, in eine Druckmedienabfrage eingefügt werden:

```
<link rel="stylesheet" href="print.css" media='print'>
```

Medienabfragen ermöglichen auch, eine responsive Erfahrung zu schaffen, bei der bestimmte Stile für kleine Bildschirme, große Bildschirme und alle weiteren Möglichkeiten dazwischen angewendet werden. Die Syntax von Medienabfragen erlaubt auch die Erstellung von Regeln, die abhängig von den Gerätecharakteristiken genutzt werden können.

```
@media (query) {  
  /* CSS Rules used when query matches */  
}
```

Wir können zwar für viele verschiedene Elemente Abfragen (Queries) erstellen, diejenigen, die am häufigsten beim responsiven Webdesign zum Einsatz kommen, sind jedoch min-width, max-width, min-height und max-height. Hier zwei Beispiele aus dem CMBasic:

```
/* bis 800 Pixel Bildschirmbreite, z.B. Tablet */  
@media (max-width: 850px) {  
  #header img { max-width: 100%; height: auto; }  
}
```

```
/* bis 600 Pixel Bildschirmbreite, z.B. Smartphone */
@media (max-width: 600px) {
body { margin: 0; padding: 0;}
#logo {width: 100%;}
#logo h1 {font-size: 3.2em;}
#logo h2 {font-size: 1.5em;}
#topmenu {display: block; float: none !important; width: 100%; }
#content {display: block; float: none !important; width: 95%; }
}
```

14. Erkenntnisse & Vermutungen

- Bei gleichen CSS-Einträgen gilt immer der letzte in der Datei. Er überschreibt den vorderen im Browser-Speicher!
- Ergänzungslinks in der index.php überschreiben den vorliegenden CSS-Eintrag.

15. Tipps & Tricks

- [Siehe CSS4You](#) (unter CSS-Kurs) Tipps & Tricks > mehr