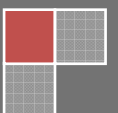


2008

CSS-Tutorial

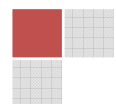
Grundlegende CSS-Kenntnisse

Dieses Tutorial gibt Ihnen verschiedene CSS-Kenntnisse mit an die Hand, damit Sie Ihre Webseite ansprechend gestalten können.

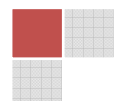


Inhaltsverzeichnis

1. Einleitung.....	4
2. CSS einsetzen, und wie es funktioniert	5
2.1 CSS-Syntax	5
2.2 Die CSS-Syntax anwenden	5
2.2.1 Methode 1: als Tag (In-document).....	5
2.2.2 Methode 2: Als Attribut (In-line).....	6
2.2.3 Methode 3: Extern (per Style Sheet).....	6
2.3 Mehrere Eigenschaften miteinander kombinieren.....	7
3. Farben und Hintergründe	8
3.1 Hintergrundfarbe.....	8
3.2 Vordergrundfarbe (oder auch: Textfarbe)	8
3.3 Hintergrundbilder.....	8
3.4 Hintergrundbild-Wiederholung definieren	9
3.4.1 Horizontale Wiederholung	9
3.4.2. Vertikale Wiederholung	9
3.4.3 Horizontale und vertikale Wiederholung.....	9
3.4.4 Keine Wiederholung.....	9
3.5 Fixierung des Hintergrundbildes	10
3.5.1 Hintergrundbild fixieren	10
3.5.2 Hintergrundbild nicht fixieren	10
3.6 Positionierung eines Hintergrundbildes.....	10
4. Textformatierung	11
4.1 Schriftart auswählen.....	11
4.2 Texte formatieren.....	11
4.2.1 Text ausrichten	11
4.2.2 Text einrücken.....	12
4.2.3 Texte und Schriftarten formatieren	12
4.3 Verweise (Links) gestalten.....	13
5. Gruppierung von Elementen	15
5.1 Gruppieren von Elementen in Klassen	15
5.2 Einzelne Elemente hervorheben	15



5.3 Weitere Gruppierungsmöglichkeiten.....	16
6. Kästchenmodell.....	17
6.1 Außenabstand verändern.....	17
6.2 Innenabstand verändern.....	18
6.3 Rahmen erstellen und verändern.....	18
7. Das Floating-Modell.....	20
7.1 Ein float-Element definieren.....	20
7.2 float stoppen.....	20
8. Elemente auf der Webseite positionieren.....	22
8.1 Absolute und relative Positionierung.....	22
8.2 Ebenen definieren.....	23
9. Anhang.....	25
9.1 Wo sind die Bilder hin?.....	25
9.2 Urheberrechtshinweis.....	25



1. Einleitung

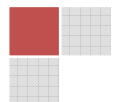
Sie haben also Ihre erste Webseite ins Netz gestellt. Doch irgendwie wirkt alles ein wenig trocken und farblos. HTML bringt zwar von sich aus einige Gestaltungsmöglichkeiten für die Webseite mit sich, aber vieles lässt sich mit HTML auch wiederum nicht machen.

Genau hier setzen die *Cascading Style Sheets*, oder auch CSS abgekürzt, an. Mit CSS ist sehr leicht, ihrer Webseite ein ansprechendes Layout zu verpassen. Jeder, der etwas mehr als nur eine ganz einfache, rein private Webseite möchte, kommt nicht um CSS herum.

Dieses Tutorial wurde geschrieben, Ihnen einen Einblick in CSS und die Möglichkeiten, die sich damit eröffnen, zu verschaffen. Um mit CSS zu beginnen, sollten Sie sich bereits ein wenig mit HTML auskennen. Dazu können wir auch an dieser Stelle unser HTML-Tutorial empfehlen.

Genau wie beim Erstellen von Webseiten brauchen Sie auch für CSS nur einen Texteditor und einen Browser, um sich das Ergebnis hinterher anzuschauen.

Und nun wünsche ich Ihnen viel Spaß beim Lesen dieses Tutorials und beim Ausprobieren der Beispiele, die Sie in dieser Lektüre finden.



2. CSS einsetzen, und wie es funktioniert

CSS ist HTML gar nicht mal so unähnlich. Viele Eigenschaften von CSS gleichen denen von HTML. Doch gibt es auch einige Unterschiede, in der Anwendung zum Beispiel. In diesem Kapitel erkläre ich Ihnen, wie CSS angewandt wird.

2.1 CSS-Syntax

Die CSS-Syntax wird in drei Teilen betrachtet und ist wie folgt aufgebaut:

```
selector { eigenschaft: wert; }
```

Beachten Sie dabei bitte die Verwendung der spitz zulaufenden Klammern ([ALT Gr]+[7] bzw. [ALT Gr]+[0]) und auf das Semikolon am Ende jedes Wertes.

Beispiel:

```
body { background-color: lightblue; }
```

2.2 Die CSS-Syntax anwenden

Wie schon erwähnt, ist CSS ähnlich aufgebaut wie HTML. Dabei gibt es mehrere Möglichkeiten, CSS in Ihre Webseiten einzubauen:

2.2.1 Methode 1: als Tag (In-document)

Mit dem HTML-Tag `<style>...</style>` lässt sich CSS auf das gesamte Dokument anwenden.

Beispiel:

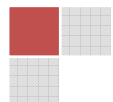
```
<html>
<head>
  <title>CSS In-document einbinden</title>
  <style type="text/css">
    body {background-color: lightblue;}
  </style>
</head>

<body>
  ...
</body>

</html>
```

Erläuterung:

Mit diesem Beispiel wird eine Webseite erzeugt, die einen blauen Hintergrund hat. Wählen Sie statt des body-selectors beispielsweise den für einen Absatz (*p*), so würde diese Auswahl für das gesamte HTML-Dokument gelten.



2.2.2 Methode 2: Als Attribut (In-line)

Eine weitere Möglichkeit, CSS einzubinden, ist per Attribut. Dazu wird dem HTML-Tag die entsprechende CSS-Definition als Attribut hinzugefügt.

Beispiel:

```
<html>
<head>
  <title>CSS In-line einbinden</title>
</head>

<body style="background-color: lightblue;">
...
</body>

</html>
```

Erläuterung:

Auch mit diesem Beispiel wird wieder eine Webseite mit einem hellblauen Hintergrund erzeugt. Der Unterschied bei dieser Methode ist, dass CSS hier als Attribut dem HTML-Tag hinzugefügt wird. So lassen sich beispielsweise für einzelne Absätze verschiedene Layouts definieren.

2.2.3 Methode 3: Extern (per Style Sheet)

Die dritte Methode, CSS in Ihre Webseite zu bringen, ist per externes Style Sheet. Diese Methode hat den Vorteil, dass auf all Ihren HTML-Dokumenten dieses Style Sheet als Layout-Definition dienen kann, ohne dass Sie immer wieder einzelne Tags mit Attributen versehen, oder in jeden Kopf Ihr komplettes Style Sheet einfügen müssen. Das externe Verfahren ist auch jenes, welches wir Ihnen ans Herz legen möchten.

Ein weiterer Vorteil ist nämlich, dass wenn Sie einmal Ihr Design verändern möchten, Sie nur die CSS-Datei anpassen brauchen. Schon ändert sich auf Ihrer kompletten Webseite das Layout. Ohne, dass Sie dafür dutzende Dateien bearbeiten bräuchten.

Um die externe Methode anwenden zu können, müssen Sie eine neue Textdatei erstellen, die Sie mit der Dateiendung `.css` abspeichern. Diese CSS-Datei legen Sie einfach auf Ihren Webserver ab. In den HTML-Dokumenten muss nun auf dieses externe Style-Sheet verwiesen werden.

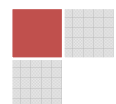
Beispiel:

```
<html>
<head>
  <title>CSS per Style Sheet einbinden</title>
  <link rel="stylesheet" type="text/css" href="stylesheet.css" />
</head>

<body>
...
</body>

</html>
```

In der CSS-Datei selbst definieren Sie anschließend das Layout Ihrer Webseite in der *In-document*-Methode. Fortan werden alle Webseiten, die dieses Style Sheet als Vorlage benutzen, von dieser einen, zentralen Datei aus gesteuert, was das Layout angeht.



2.3 Mehrere Eigenschaften miteinander kombinieren

Selbstverständlich kann auch in CSS mehrere Eigenschaften eines *selectors* zusammen kombiniert werden.

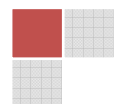
Beispiel:

```
body {  
    background-color: darkred;  
    background-image: url("background.jpg");  
    background-repeat: no-repeat;  
}
```

```
body { background: darkred url("background.jpg") no-repeat; }
```

Beide Beispiele zeigen das gleiche Prinzip: Verschiedene Eigenschaften werden unter einem *selector* (in diesem Fall: *body*) zusammengefasst.

Im Endeffekt müssen Sie dabei selbst entscheiden, was Sie für sich übersichtlicher empfinden. Ich selbst präferiere das erste dieser Beispiele. Aber das ist Geschmackssache.



3. Farben und Hintergründe

Da Sie nun CSS anwenden können, möchten Sie sicherlich auch schnell die ersten Dinge ausprobieren. Neben dem Inhalt ist natürlich auch das Aussehen der Webseite wichtig. Was dabei als erstes Auffällt, sind die Farben der Webseite und die Hintergrundmotive. Deren Einsatz wollen wir nun gemeinsam betrachten.

Bitte beachten Sie, dass ab hier alle CSS-Beispiele nur noch als CSS-Element angegeben werden, ohne HTML-Aufbau. Dieses CSS-Element können Sie so, wie es ist, in Ihre Webseite übernehmen. Entweder im In-line Verfahren oder, was ratsamer wäre, über ein externes Style Sheet.

3.1 Hintergrundfarbe

Mit der Eigenschaft `background-color`: können Sie die Hintergrundfarbe eines Elementes bestimmen.

Beispiel:

```
body { background-color: lightblue; }
```

```
h1 { background-color: darkred; }
```

Erläuterung:

Mit diesen beiden Beispielen wird eine Webseite mit einem hellblauen Hintergrund und dunkelroter Hinterlegung der Überschriften angezeigt.

3.2 Vordergrundfarbe (oder auch: Textfarbe)

Die Vordergrundfarbe in CSS wird mit der Eigenschaft `color`: definiert.

Beispiel:

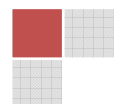
```
h1 { color: darkred; }  
h2 { color: red; }  
h3 { color: lightred; }
```

Erläuterung:

In diesem Beispiel werden für die drei Überschrift-Arten verschiedene Farben definiert. Die Überschrift erster Ebene wird dabei dunkelrot, die Überschrift zweiter Ebene rot und die Überschrift dritter Ebene hellrot angezeigt.

3.3 Hintergrundbilder

Um ein Hintergrundbild in Ihre Webseite einzufügen, müssen Sie die CSS-Eigenschaft `background-image`: verwenden.



Beispiel:

```
body { background-image: url("image.jpg"); }
```

```
body { background-image: url("../grafiken/background.jpg"); }
```

```
body { background-image: url("http://www.ihreurl.de/background.jpg"); }
```

Erläuterung:

Je nachdem, wo die Bilddatei, die Sie als Hintergrundbild verwenden möchten, abgelegt worden ist, können Sie entsprechend den Pfad oder die URL verwenden. Liegt die Datei im gleichen Verzeichnis wie die CSS-Datei, können Sie auch einfach nur den Namen der Grafik angeben.

3.4 Hintergrundbild-Wiederholung definieren

CSS bietet gegenüber HTML viele verschiedene Möglichkeiten zur Wiederholung des Hintergrundbildes.

3.4.1 Horizontale Wiederholung

Mit CSS ist es möglich, das Hintergrundbild nur horizontal wiederholen zu lassen.

Beispiel:

```
body { background-repeat: repeat-x; }
```

3.4.2. Vertikale Wiederholung

Genau so ist es möglich, das Hintergrundbild nur vertikal zu wiederholen.

Beispiel:

```
body { background-repeat: repeat-y; }
```

3.4.3 Horizontale und vertikale Wiederholung

Ähnlich wie in HTML, lässt sich natürlich auch ein Kacheffekt erzeugen, indem das Hintergrundbild sowohl horizontal als auch vertikal wiederholt wird.

Beispiel:

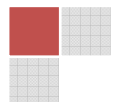
```
body { background-repeat: repeat; }
```

3.4.4 Keine Wiederholung

Selbstverständlich ist es mit CSS auch möglich, das Hintergrundbild nur einmal, ohne Wiederholung anzuzeigen.

Beispiel:

```
body { background-repeat: no-repeat; }
```



3.5 Fixierung des Hintergrundbildes

Es ist auch möglich, das Bild zusammen mit dem Text scrollen zu lassen, oder es auf dem Hintergrund zu fixieren. Der Unterschied ist der, dass bei einer Fixierung im Hintergrund das Hintergrundbild an der vorgegebenen Stelle stehen bleibt, während der Besucher über die Seite scrollt. Ist das Bild nicht fixiert, scrollt es zusammen mit den restlichen Elementen „nach oben“.

3.5.1 Hintergrundbild fixieren

Um das Hintergrundbild zu fixieren, also um es an der Stelle zu lassen, wo es steht muss folgende CSS-Eigenschaft verwendet werden.

Beispiel:

```
body { background-attachment: fixed; }
```

3.5.2 Hintergrundbild nicht fixieren

Soll das Hintergrundbild zusammen mit der Seite scrollen, wird hingegen folgende CSS-Eigenschaft verwendet.

Beispiel:

```
body { background-attachment: scroll; }
```

3.6 Positionierung eines Hintergrundbildes

Etwas, was mit reinem HTML so nicht möglich ist, wird durch den Einsatz von CSS hingegen kinderleicht: Die Positionierung eines Hintergrundbildes.

In der Standardeinstellung wird das Hintergrundbild im Browser links oben positioniert. Jedoch ist es auch möglich, mithilfe von CSS das Bild mittig oder rechts unten im Fenster darzustellen – oder eben an jener Position, die Sie gerne hätten. Das funktioniert über den Tag `background-position`:

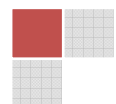
Beispiel:

```
body { background-position: 100px 200px; }  
  
body { background-position: 100% 100%; }  
  
body { background-position: bottom left; }
```

Erläuterung:

Die Angabe der Koordinaten kann dabei als fester Wert erfolgen (in Pixel oder Zentimeter), in Prozent oder mit festen Angaben in Worten. Dabei wird im ersten Wert die Position vom linken Bildschirmrand, im zweiten Wert die Position vom oberen Bildschirmrand angegeben.

Für das erste Beispiel bedeutet dies, dass das Hintergrundbild 100 Pixel vom linken Bildschirmrand und 200 Pixel vom oberen Bildschirmrand entfernt ist. Im zweiten Beispiel hingegen ist das Bild sowohl 100% vom linken, als auch vom rechten Bildschirmrand entfernt (sprich: rechts unten). Das dritte Beispiel hingegen sagt, dass das Hintergrundbild unten links am Bildschirmrand positioniert sein soll.



4. Textformatierung

Im Gegensatz zu reinem HTML bietet CSS natürlich ebenso eine breite Palette zur Textformatierung an, ähnlich wie es bei Grafiken und Farben der Fall ist. In diesem Teil des Tutorials möchte ich Ihnen zeigen, was so alles mit dem Text angestellt werden kann.

4.1 Schriftart auswählen

Schriftarten lassen sich in CSS recht einfach über die Eigenschaft `font-family`: auswählen.

Beispiel:

```
p { font-family: arial; }  
h1 { font-family: verdana, arial, sans-serif; }  
h2 { font-family: "times new roman", serif; }
```

Erläuterung:

Diese drei Beispiele bewirken unterschiedliches. Das erste Beispiel wählt einfach die Schriftart *Arial* für alle Standard-Texte aus.

Das zweite Beispiel hingegen wählt für alle Überschriften bevorzugt die Schriftart *Verdana* aus. Ist *Verdana* auf dem Rechner des Betrachters nicht verfügbar, wird die Schriftart *Arial* ausgewählt. Ist auch *Arial* nicht verfügbar, wird ein Zeichensatz aus den serifenlosen Schriften ausgewählt.

Beispiel drei zeigt, dass für die Überschrift die Schriftart *Times New Roman* gelten soll. Schriftarten, die Leerzeichen enthalten, müssen dabei in Anführungszeichen stehen. Ist *Times New Roman* nicht verfügbar, würde in diesem Beispiel eine andere Schriftart mit Serifen genommen werden.

4.2 Texte formatieren

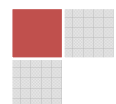
CSS besitzt einige Möglichkeiten, die gewählten Schriftarten bzw. Texte zu formatieren.

4.2.1 Text ausrichten

Mit der Eigenschaft `text-align`: lässt sich der Text grundlegend ausrichten. Ähnlich wie bei dem HTML-Attribut lässt sich mit `text-align`: der Text in CSS linksbündig, rechtsbündig, zentriert oder in Blockschrift ausrichten.

Beispiel:

```
p { text-align: right; }  
p { text-align: left; }  
p { text-align: center; }  
p { text-align: justify; }
```



4.2.2 Text einrücken

Mit CSS können Sie es sogar bewerkstelligen, dass Ihr Text in der ersten Zeile um einen bestimmten Wert eingerückt wird.

Beispiel:

```
p { text-indent: 20px; }
```

Erläuterung:

Mit diesem Beispiel würde jeder Text, der mit dem HTML-Tag `<p>` beginnt, um 20 Pixel eingerückt werden.

4.2.3 Texte und Schriftarten formatieren

CSS bietet verschiedene Möglichkeiten, Texte hervorzuheben. Dazu gibt es zwei Eigenschaften zur Auswahl: `text-decoration`, `font-style` oder `font-weight`.

Beispiel:

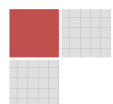
```
h1 { font-style: italic;
      font-weight: bold;
      text-decoration: overline;
    }
```

Erläuterung:

Mit diesem Beispiel würde die Überschrift kursiv, fett und überstrichen (ja, man kann Text mit CSS auch "überstreichen") dargestellt werden.

Zusammengefasst habe ich eine Liste mit möglichen Textformatierungen zusammengestellt:

CSS-Eigenschaft	Erläuterung
text-decoration: underline;	Unterstrichener Text
text-decoration: overline;	„Überstrichener“ Text
text-decoration: line-through;	Durchgestrichener Text
text-decoration: blink;	Blinkender Text
font-style: italic;	Kursiver Schriftstil
font-style: oblique;	Schräggestellter Schriftstil
font-weight: bold;	Fettgedruckt
font-weight: bolder;	Extrafettgedruckt
font-weight: lighter;	Dünngedruckt
font-weight: 100, 200, ... , 900	Druckstil extra dünn (100) bis extra fett (900) in 100er-Schritten
font-weight: normal;	normales Schriftgewicht
font-variant: small-caps;	Schriftvariante kapital (kleine Großbuchstaben)
font-size: xx-small;	winzige Schriftgröße
font-size: x-small;	sehr kleine Schriftgröße
font-size: small;	kleine Schriftgröße



font-size: medium;	mittlere Schriftgröße
font-size: large;	große Schriftgröße
font-size: x-large;	sehr große Schriftgröße
font-size: xx-large;	riesige Schriftgröße
	Natürlich sind auch prozentuale, oder andere relative Angaben möglich, wie zum Beispiel:
font-size: 1em;	1em wäre die vom Benutzer im Browser standardmäßig eingestellte Schriftgröße
word-spacing: 0.5em;	Wortabstand 0.5em
letter-spacing: 0.3em;	Zeichenabstand 0.3em
	Hier sind auch feste Angaben wie px, oder mm möglich.
text-transform: capitalize;	Wortanfänge sind Großbuchstaben
text-transform: uppercase;	Nur Großbuchstaben
text-transform: lowercase;	Nur Kleinbuchstaben

4.3 Verweise (Links) gestalten

Auch Links lassen sich mit CSS relativ einfach gestalten. In der folgenden Liste werden die Eigenschaften, die man verwenden kann, aufgelistet und erläutert. Prinzipiell kann man die Links nach seinen eigenen Vorstellungen mit den bereits bekannten Textformatierungen gestalten. Nach der Liste erhalten Sie ein Anwendungsbeispiel.

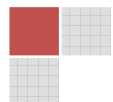
CSS-Eigenschaft	Erläuterung
a:link	Verweis zu noch nicht besuchten Seiten
a:visited	Verweis zu bereits besuchten Seiten
a:focus	Link, der gerade den Fokus hat (z.B. beim „Durchspringen“ der Links mit der Tabulator-Taste)
a:hover	Link, auf dem gerade der Mauszeiger ruht
a:active	Gerade aktiver Link

Beispiel:

```
a:link { text-decoration: none; color: red; }
a:visited { text-decoration: none; color: red; }
a:focus { text-decoration: none; color: red; }
a:hover { text-decoration: underline; color: blue; text-transform: uppercase; }
a:active { text-decoration: underline; color: blue; text-transform: uppercase; }
```

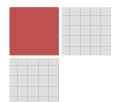
Erläuterung:

Mit diesem Beispiel würde ein Link auf Ihrer Seite im Standardzustand nicht unterstrichen und in roter Farbe dargestellt werden. Das gilt dabei sowohl für bereits besuchte, als auch für noch nicht besuchte Seiten.



Führt der Benutzer jetzt aber mit dem Mauszeiger über den Link oder klickt ihn an, wird der Link unterstrichen und in blauer Farbe dargestellt. Dazu kommt, dass alle Buchstaben des Verweises in Großbuchstaben dargestellt werden.

Wie Sie sehen, lassen sich links so prinzipiell sehr frei formatieren. Das funktioniert übrigens mit so manchen Eigenschaften auch für andere *selectoren*. Beispielsweise könnten Sie Tabellenzellen beim Überfahren des Mauszeigers mit der Eigenschaft `td:hover` mit einer anderen Hintergrundfarbe versehen. Ihrer Fantasie sind dabei keine Grenzen gesetzt.



5. Gruppierung von Elementen

Sie wollen bestimmte Textabschnitte besonders hervorheben? Sie wollen in Aufzählungslisten bestimmte Elemente extra kenntlich machen? Das ist mit CSS überhaupt kein Problem. Extra für solche Wünsche kann man mit CSS seine verschiedenen Elemente gruppieren. Dazu stehen die Eigenschaften `class` und `id` zur Verfügung.

5.1 Gruppieren von Elementen in Klassen

Nehmen wir für dieses Beispiel an, dass Sie ein Fußball-Fan sein. Auf Ihrer Homepage erstellen Sie eine Liste mit verschiedenen Verweisen zu verschiedenen Vereinen. Jetzt möchten Sie, dass alle Links, die zum Verein Bayern-München verweisen, in rot dargestellt werden und alle Links, die zum Verein Schalke 04 verweisen, in blau dargestellt werden. Mit einer Klassendefinition ist dies überhaupt kein Problem.

Beispiel:

```
<p>Bayern München</p>
<ul>
  <li><a href="kader.htm" class="bayern">Kader von Bayern München</a></li>
  <li><a href="statistik.htm" class="bayern">Statistik des Clubs</a></li>
  <li><a href="tabelle.htm" class="bayern">Tabellenplatzentwicklung</a></li>
</ul>

<p>Schalke 04</p>
<ul>
  <li><a href="kader.htm" class="schalke">Kader von Schalke</a></li>
  <li><a href="statistik.htm" class="schalke">Statistik des Clubs</a></li>
  <li><a href="tabelle.htm" class="schalke">Tabellenplatzentwicklung</a></li>
</ul>
```

In unserer CSS-Datei müssen wir den Klassen nun noch die entsprechenden Eigenschaften zuweisen.

Beispiel:

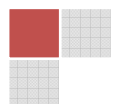
```
a.bayern { color: red; }
a.schalke { color: blue; }
a { color: green; }
```

Erläuterung:

Mit diesem Beispiel würden alle Links, die die Klasse *„bayern“* enthalten, rot angezeigt werden, während alle Links, die die Klasse *„schalke“* enthalten, blau angezeigt würden. Da Sie in diesem Beispiel Werder Bremen-Fan sind, werden alle normalen Links natürlich grün angezeigt.

5.2 Einzelne Elemente hervorheben

Wenn Sie hingegen auf Ihrer Seite nur ein einzelnes Element hervorheben möchten, müssen Sie die Eigenschaft `id` nutzen.



Beispiel:

```
<h1 id="css">CSS-Tutorial</h1>
  <h2 id="k1">Kapitel 1</h2>
  <h2 id="k2">Kapitel 2</h2>
```

In der CSS-Datei müssen wir nun die ID entsprechend formatieren.

Beispiel:

```
#css { color: darkred; text-transform: uppercase; }
#k1 { color: darkblue; text-transform: lowercase; }
#k2 { color: yellow; text-transform: lowercase; }
```

Erläuterung:

Mit `id` ist es möglich, einzelne Elemente innerhalb Ihrer Webseite zu gestalten. Dabei kann jedes ID-Element nur einmal vorkommen. Der Einsatz dieser Eigenschaft ist dann sinnvoll, wenn Sie vereinzelt ein bestimmtes Element auf Ihrer Seite hervorheben oder anders gestalten möchten.

5.3 Weitere Gruppierungsmöglichkeiten

Um nicht bestimmte Elemente von vornherein klassifizieren zu müssen, ist auch der Einsatz von `` und `<div>` hilfreich. Diese beiden HTML-Tags fügen der Webseite nichts hinzu. So lassen sich einzelne Textpassagen oder ganze Listen einfach formatieren, ohne dass man dazu in der Webseite ein extra Element, wie z.B. eine Aufzählungsliste, anlegen müsste.

Beispiel:

```
<p><span class="red">Hervorhebungen</span> sind eine feine Sache, um dem
Leser etwas <span class="red">deutlich</span> zu machen.</p>

<div id="linksbuendig">
<p>Jedoch ist es natürlich auch möglich, komplette Bereiche, mit allem was
dazugehört, auf einmal zu formatieren.</p>
</div>

<div id="rechtsbuendig">
<p>Oder man macht es halt ganz anders. Das ist einem völlig selbst
überlassen.</p>
</div>
```

Die CSS-Datei sieht dabei dann wie folgt aus:

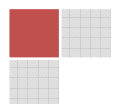
Beispiel:

```
span.red { color: red; font-weight: bold; }

#linksbuendig { text-align: left; }
#rechtsbuendig { text-align: right; }
```

Erläuterung:

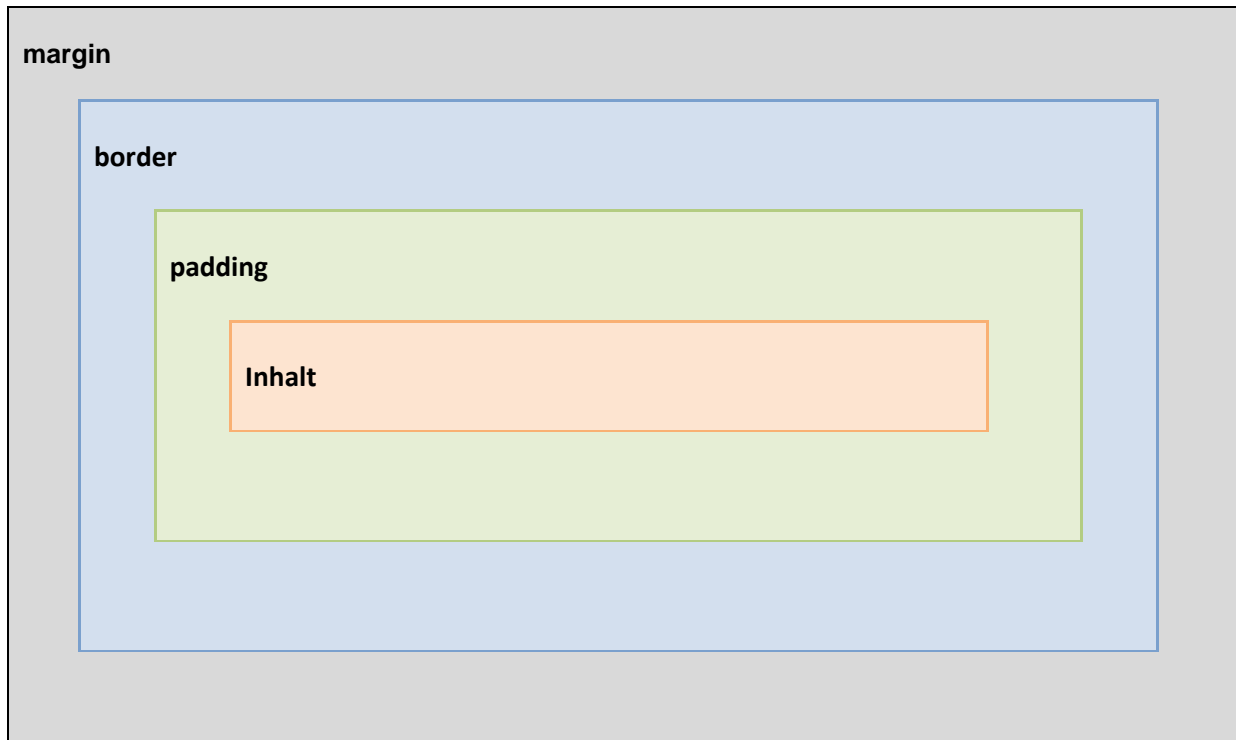
`` wird meistens dann eingesetzt, wenn es einzelne Dinge innerhalb eines Elementes zum Hervorheben gibt. `<div>` hingegen gebraucht man dann, wenn es komplette Elemente zum formatieren gibt. Mit Hilfe dieser beiden Tags lassen sich jedoch noch viel mehr Dinge anstellen. Doch dazu mehr in den kommenden Lektionen.



6. Kästchenmodell

CSS definiert ein Kästchenmodell, oder auch Box-Modell genannt, um die einzelnen HTML-Elemente herum. Dieses Modell besteht dabei aus mehreren Elementen, die ich nun näher erläutern möchte.

Anzeigebeispiel:



Diese Abbildung zeigt, dass jedes HTML-Element (hier als „Inhalt“ gekennzeichnet) von verschiedenen Kästchen umgeben ist. CSS kann dabei jedes dieser Kästchen ansteuern und verändern, so dass sich individuelle und vielfältige Gestaltungsmöglichkeiten ergeben. Dabei ist es egal, was Sie verändern möchten. Sie können den Außenabstand (*margin*), die Ränder (*border*), Innenabstände (*padding*) und den Inhalt (*content*) frei gestalten.

6.1 Außenabstand verändern

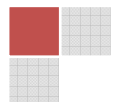
Der Außenabstand eines HTML-Elementes lässt sich mit dem Wert *margin* verändern.

Beispiel:

```
body { margin-top: 200px;
      margin-right: 100px;
      margin-left: 100px;
      margin-bottom: 200px;
}
```

Erläuterung:

Wie Sie sehen können, besteht der Außenabstand aus vier Teilen: Oben, Unten, Links und Rechts. Jeder dieser Werte lässt sich verändern. In diesem Beispiel würde die Webseite erst bei 200 Pixeln



unter dem oberen Bildschirmrand beginnen, dabei zum linken und zum rechten Seitenrand jeweils 100 Pixel Abstand halten und nach unten noch einmal 200 Pixel Abstand haben.

6.2 Innenabstand verändern

Auch die Innenabstände der Elemente lassen sich verändern. Dazu dient der Wert *padding*.

Beispiel:

```
td { background: lightblue;
      padding-top: 10px;
      padding-right: 120px;
      padding-left: 100px;
      padding-bottom: 10px;
    }
```

Erläuterung:

In diesem Beispiel würde alles, was in eine Tabellenzelle hineingeschrieben wird, 10 Pixel Abstand zum oberen Rand der Zelle haben, 120 Pixel Abstand zum rechten Rand, 100 Pixel zum linken Rand und noch einmal 10 Pixel zum unteren Rand halten. Dazu kommt, dass der Tabellenzellen-Hintergrund in hellblau dargestellt wird.

6.3 Rahmen erstellen und verändern

Mit CSS ist es ein Leichtes, verschiedene Rahmen um einzelne HTML-Elemente herum zu bauen.

Beispiel:

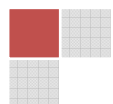
```
h1 { border-style: dotted;
      border-color: red;
      border-width: thick;
    }
```

Erläuterung:

Mit diesem Beispiel werden alle Überschriften mit einem gepunkteten, dicken Rahmen versehen. Der Rahmen hat dabei eine rote Farbe.

Weitere Anzeigemerkmale finden Sie in der folgenden Referenzliste:

CSS-Eigenschaft	Erläuterung
border-style: dotted;	Gepunkteter Rahmen
border-style: dashed;	Gestrichelter Rahmen
border-style: solid;	Durchgezogener Rahmen
border-style: double;	Doppelter, durchgezogener Rahmen
border-style: groove;	Schattenwurf nach oben
border-style: ridge;	3D-Effekt
border-style: inset;	Äußerer Schatten
border-style: outset;	Innerer Schatten
border-width: thin;	Dünnere Rahmen



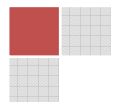
border-width: medium;	Mittelstarker Rahmen
border-width: thick;	Dicker Rahmen
	Die Rahmenstärke lässt sich auch mit einer px-Angabe definieren.
border-top-color: blue;	Obere Rahmenlinie ist blau
border-right-color: blue;	Rechte Rahmenlinie ist blau
border-left-color: blue;	Linke Rahmenlinie ist blau
border-bottom-color;	Untere Rahmenlinie ist blau
border-color: red blue green black;	Verkürzte Version der eben genannten Gestaltungsmöglichkeiten*

Tipp:

Ähnlich wie bei der `border-color` lässt sich auch für `border-width` und `border-style` die Angaben für die oberen, unteren, linken und rechten Rahmen separat definieren. So stehen einem viele verschiedene Gestaltungsmöglichkeiten zur Auswahl.

* folgende Regeln sind bei dieser verkürzten Version zu beachten:

- Zwei Angaben: Die erste Angabe bestimmt den Rahmentyp für oben und unten, die zweite Angabe den Rahmentyp für links und rechts.
- Drei Angaben: Die erste Angabe bestimmt den Rahmentyp für oben, die zweite den Rahmentyp für links und rechts und die dritte Angabe den Rahmentyp für unten.
- Vier Angaben: Die erste Angabe bestimmt den Rahmentyp für oben, die zweite für rechts, die dritte für unten und die vierte den Rahmentyp für links.



7. Das Floating-Modell

Sie kennen das aus einer Zeitung: Links oben im Artikel ist ein Bild abgedruckt. Rechts daneben beginnt der Text. Ist der Text länger, als das Bild hoch ist, wird der Text unter dem Bild fortgesetzt. Sie möchten ein solches Erscheinungsbild auch für Ihre Seiten einrichten? Mit CSS kein Problem. Die Eigenschaft `float` sorgt für einen Effekt, der genau so ist.

7.1 Ein float-Element definieren

Beispiel:

```
<div id="floating">
  
</div>
```

In unserer CSS-Datei definieren wir nun die floating-Box:

Beispiel:

```
#floating {
    float:left;
    width: 200px;
}
```

Erläuterung:

Wenn wir dieses Beispiel benutzen, um den Text um unser Bild herum "floaten" zu lassen, ergibt sich dabei der Effekt, dass der Text sich rechts vom Bild befindet. Das hängt damit zusammen, dass wir das Bild als „links floatend“ eingestuft haben, oder anders ausgedrückt: Das Bild befindet sich durch die Angabe `float:left;` auf der linken Bildschirmseite.

Die Eigenschaft `float` kann entweder den Wert *left* oder den Wert *right* haben.

7.2 float stoppen

Manchmal ist es auch sinnvoll, nur eine Überschrift neben einem „floatenden“ Bild zu positionieren, den Text aber wieder unter das Bild zu stellen. Dazu muss die Eigenschaft `clear` benutzt werden.

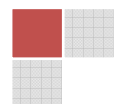
Beispiel:

```
<div id="floating">
  
</div>

<h1>Dies ist ein Beispielbild</h1>

<p class="floatingstopp">Dieser Text wird wieder unter dem Bild
dargestellt.</p>
```

Unsere CSS-Datei ist dabei wie folgt aufgebaut:



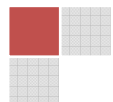
Beispiel:

```
#floating {  
    float: left;  
    width: 200px;  
}  
  
.floatingstopp { clear: both; }
```

Erläuterung:

Mit diesem Beispiel würde neben unserem Bild die Überschrift *“Dies ist ein Beispielbild”* erscheinen. Der Text hingegen wird nicht mehr neben dem Bild dargestellt, sondern unter dem Bild fortgesetzt.

Die Eigenschaft `clear` kann folgende Werte haben: *both*, *left* oder *right*.



8. Elemente auf der Webseite positionieren

Ein weiterer, designtechnisch sehr nützlicher Vorteil von CSS ist, dass man Elemente frei auf der Webseite positionieren kann. Und das ist dabei noch nicht einmal schwer.

Dabei kann man entweder eine absolute oder eine relative Positionierung wählen. Der Unterschied zwischen den beiden Möglichkeiten ist folgender:

Die relative Positionierung wird von der eigentlichen Stelle des Elementes aus berechnet, an dem es auf die Seite gesetzt wurde. Das heißt, wenn Sie beispielsweise ein Bild links oben in die Ecke setzen und dann die Position des Bildes relativ bestimmen, bewegen Sie dieses Bild von seiner ursprünglichen Koordinate aus nach oben, unten, links oder rechts. Dabei wird ein kleiner Freiraum hinterlassen. Die Positionierung orientiert sich an der ursprünglichen Position des Elementes.

Bei der absoluten Positionierung würde das Bild quasi „auf die Seite drauf geworfen“. Sie können es frei positionieren, ohne dass dabei eine Lücke hinterlassen wird. Die Positionierung orientiert sich also am Browserfenster.

In beiden Versionen können dabei andere Webseiten-Elemente überlagert werden.

8.1 Absolute und relative Positionierung

Beispiel absolute Positionierung:

```
#bsp1 {
  position: absolute;
  top: 50px;
  left: 50px;
}

#bsp2 {
  position: absolute;
  top: 50px;
  right: 50px;
}

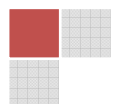
#bsp3 {
  position: absolute;
  bottom: 50px;
  left: 50px;
}

#bsp4 {
  position: absolute;
  bottom: 50px;
  right: 50px;
}
```

Unsere HTML-Datei könnte dann wie folgt aussehen:

Beispiel:

```
<div id="bsp1"><p>Hallo von links oben</p></div>
<div id="bsp2"><p>Hallo von rechts oben</p></div>
<div id="bsp3"><p>Hallo links unten</p></div>
<div id="bsp4"><p>Hallo von rechts unten</p></div>
```



Erläuterung:

Wenn wir dieses Beispiel so in unsere Webseite kopieren, erscheint in jeder Ecke der Webseite – und zwar mit einem Abstand von 50 Pixeln zum jeweiligen Rand – der im Beispiel genannte Text. Dieser Text überlagert dabei auch schon eventuell bereits vorhandene Elemente.

Beispiel relative Positionierung:

```
#bsp1 {
  position: relative;
  top: 50px;
  left: 50px;
}

#bsp2 {
  position: relative;
  top: 50px;
  right: 50px;
}

#bsp3 {
  position: relative;
  bottom: 50px;
  left: 50px;
}

#bsp4 {
  position: relative;
  bottom: 50px;
  right: 50px;
}
```

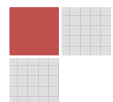
Erläuterung:

Benutzen wir nun eine relative Positionierung der Elemente, so würde sich, wenn wir das Beispiel von vorhin nehmen, der genannte Text um die angegebenen Werte von seiner ursprünglichen Position entfernen und dabei eine Lücke hinterlassen.

Nehmen wir also an, der Text wäre Teil eines längeren Abschnittes. Verwenden wir nun dieses Beispiel, wird dabei der Text aus dem Beispiel vorhin aus dem Abschnitt „herausgeschnitten“ und an die entsprechende andere Position wieder „drauf geschmissen“ – berechnet von seiner ursprünglichen Position aus.

8.2 Ebenen definieren

Mit CSS kann man sogar in die dritte Dimension arbeiten: Die Tiefe. CSS beherrscht eine Ebenenfunktion mit der sich die verschiedenen Elemente einer Webseite überlappen lassen können. Dabei ist es völlig egal, ob es sich um ein Bild, einen Text, eine Tabelle oder sonst irgendwas handelt.



Beispiel:

```
#ebene1 {
  position: absolute;
  left: 100px;
  top: 100px;
  z-index: 1;
}

#ebene2 {
  position: absolute;
  left: 150px;
  top: 150px;
  z-index: 2;
}

#ebene3 {
  position: absolute;
  left: 200px;
  top: 200px;
  z-index: 3;
}
```

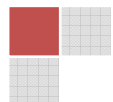
Unsere HTML-Datei sieht wie folgt aus:

Beispiel:

```
<div id="ebene1"></div>
<div id="ebene2"></div>
<div id="ebene3"></div>
```

Erläuterung:

Das Bild „04.jpg“ wird in diesem Beispiel dreimal überlappt – mit jeweils 50px Abstand mehr zur vorigen Ebene. Dabei zu beachten wäre, dass die Ebene mit dem höheren *z-index-wert* die Ebene mit dem niedrigeren *z-index-wert* überlappt.



9. Anhang

9.1 Wo sind die Bilder hin?

„Wo sind die Bilder hin?“, wird sich mit Sicherheit der ein oder andere Leser dieses Tutorials fragen. Die Antwort auf diese Frage ist relativ einfach: „Es gibt keine.“

In diesem Tutorial wurde bewusst auf den Einsatz von Bildern verzichtet. Der Grund ist, dass es meiner Meinung nach mehr Spaß macht, die Codebeispiele selbst auszuprobieren, als alles „vorgekaut“ zu bekommen. Außerdem kann man Anhand von Veränderungen in den Beispielen selbst schnell die Unterschiede feststellen. Das fördert den „Aha“-Effekt und regt zum Experimentieren an. Und experimentieren ist etwas, was man gerade beim Einstieg in CSS sehr häufig tun sollte.

Dieses Tutorial soll kein Werk darstellen, das Ihnen vorschreibt, was Sie zu tun und zu lassen haben. Ich möchte dazu anregen, selbst kreativ zu werden. Daher wurde auf den Einsatz von Bildern zur Veranschaulichung der Codebeispiele weitestgehend verzichtet.

9.2 Urheberrechtshinweis

Dieses Tutorial unterliegt dem deutschen Urheberrecht. Die Vervielfältigung und Verbreitung dieses Tutorials ist ausdrücklich erwünscht. Eine Bearbeitung und/oder Änderung des Inhaltes hingegen, egal in welcher Form, bedarf der schriftlichen Zustimmung des Autors. So lange Satz, Layout und Inhalt nicht bearbeitet wurden, darf dieses Tutorial zu privaten, nichtkommerziellen Zwecken verbreitet und vervielfältigt werden. Ein Vertrieb dieses Tutorials gegen Entgelt ist untersagt. Eine Verbreitung dieses Tutorials als Teil einer kommerziellen Sammlung redaktioneller Inhalte (beispielsweise als Inhalt auf einer CD zu einer Zeitschrift) bedarf des schriftlichen Einverständnisses des Autors.

Die Code-Beispiele, die in diesem Tutorial gezeigt werden, unterliegen dieser Regelung nicht. Die Code-Beispiele dürfen frei angewandt und verändert werden. Dies gilt wiederum nicht für beschreibende Texte wie Hinweise, Tipps und Erklärungen zu diesen Code-Beispielen.

Wenn Ihnen dieses Tutorial so gut gefällt, dass Sie es anderen zur Verfügung stellen möchten, würden ich mich über eine entsprechende Mitteilung natürlich sehr freuen.

