

Abstände bei Überschriften, Textabsätzen, Bilderausrichtung mittels CSS-Formatierungen festlegen

Das Festlegen von Abständen in einem Text- oder Webseitenlayout mag auf den ersten Anschein kein Problem zu sein. In der Praxis stellen sich jedoch schnell Abweichungen ein, gerade wenn man verschiedene Absatzformatierungen und Überschriften benutzt.

Natürlich führen immer mehrere Wege nach Rom. Es sei mir verziehen dass ich hier eine römische Stadt benutzt habe und keine griechische! Aber so lautet eben das Sprichwort.

Es ist daher sinnvoll sich zu überlegen welche Texte und Überschriften miteinander kombiniert werden. D.h. die Festlegung eines Konzeptes. Ich möchte daher mit meinen nachfolgenden Tipps nur eine Hilfestellung geben und soll daher auch kein unbedingtes Muss darstellen.

Fangen wir mit einem einfachen Container (div) an, in den ich beliebig Textabsätze, Überschriften und Bilder setzen möchte. Diesen Container weise ich als Bsp. die ID #content zu.

Formatierung des Containers:

Oftmals dient ein Container der Eingrenzung von Inhalten wie z.B. Text etc. Gerne wird auch diesem Container ein Rand zur optischen Eingrenzung gegeben. Schnell könnte man jetzt auf die Idee kommen, den Textabsätzen `<p>` einfach rechts und links einen Abstand nach aussen (margin) oder nach innen (padding) zu geben.

```
<div id="content">
    <p>Dies ist mein Beispieltext, der innerhalb des Containers
    #content sitzt.</p>
</div>
```

Stylesheet:

```
#content { width: 600px; border: 1px solid blue; }
p { margin: 0 20px 0 20px; }
```

Sieht dann ungefähr so aus:

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt.

Jetzt stört noch, dass der Text oben und unten direkt am Rand des Containers klebt. Gut, erweitere ich nun einfach die Formatierung der Abstände für oben und unten.

```
p { margin: 20px 20px 20px 20px; }
```

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt.

Jetzt möchte ich noch unterhalb des Absatzes ein Bild einsetzen.

```
<div id="content">
    <p>Dies ist mein Beispieltext, der innerhalb des Containers
    #content sitzt.</p>
    
</div>
```

Was passiert nun:

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt.



Das Bild rutscht jetzt an den linken Seitenrand und ich habe einen Versatz der nicht gewünscht war. Jetzt könnte ich natürlich dem Bild auch ein margin zuweisen um dies auszugleichen.

Füge ich jetzt noch eine h1, h2, h3 usw. Überschrift ein, muss ich die auch wieder entsprechend einrücken.

Gibt es nicht eine andere Möglichkeit dies zu umgehen, somit den Formatierungsaufwand zu reduzieren?

Ja natürlich!

Fügen wir einfach einen inneren Abstand mittels padding, für den Container #content ein. So erspare ich mir die ganzen extra Formatierungen für die Randabstände.

```
#content {
width: 600px;
border: 1px solid blue;
padding: 20px 20px 20px 20px;
}
```

Anmerkung: Da bei padding alle vier Abstände den gleichen Wert haben (20px), kann man vereinfacht auch padding: 20px; schreiben.

So erreiche ich, dass sich alle Elemente, egal ob Textabsatz, Bilder, Überschriften oder auch weitere Container sich an diesen Abständen gleich ausrichten.

Praktisch wird man jedoch später auch für die Absätze nach unten hin einen Abstand, z.B. mittels margin festlegen, da in der Regel mehrere Absätze untereinander gesetzt werden.

Absätze und Überschriften:

Für z.B. eine h1 Überschrift ergänze ich jetzt nur noch einen Abstand nach unten.

```
h1 { margin: 0 0 20px 0; }
```

Im oberen margin sind der obere, linke und rechte Abstand auf den Wert Null gesetzt. Ebenso könnte ich auch schreiben, was dann gleichwertig wäre:

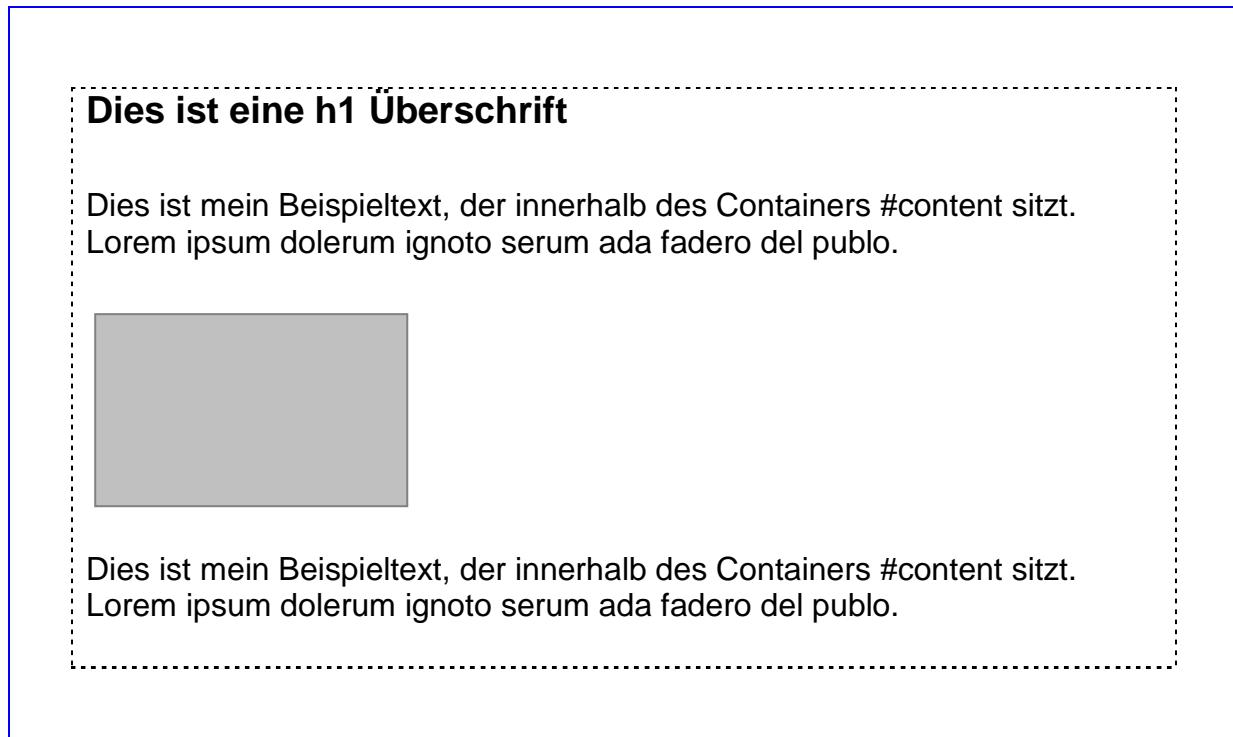
```
h1 { margin-bottom: 20px; }
```

womit ich nur direkt auf den unteren Abstand verweise bzw. ihn formatiere.

Das gleiche führe ich jetzt mit dem Absatz-Tag <p> und dem Img-Tag durch.

```
p { margin-bottom: 15px; }  
img { margin-bottom: 15px; }
```

Als Beispiel könnte dies jetzt so aussehen:



Die im oberen Beispiel gezeigte gestrichelte Linie ist nicht sichtbar und soll nur den inneren Abstand des Containers verdeutlichen und ist auch nicht maßstäblich gezeichnet.

Ich habe hierbei absichtlich nur Abstände nach unten eingeführt. Würde ich auch nach oben einen Abstand einfügen und verschiedene Elemente kombinieren würde es unerwünschte Effekte, d.h. verschieden große Abstände geben.

Positionierung, Formatierung von Bildern:

Bei der o.g. Formatierung des Bild-Tags per `img { margin-bottom: 15px; }` ist zu beachten, dass sich dies auf **alle** Bilder im HTML-Dokument auswirkt. Dies ist meistens nicht gewünscht. Um nur einem Bild ein bestimmtes Aussehen oder bestimmte Abstände zu zuweisen, führt man hierzu eine Klasse ein. Dies könnte wie folgt aussehen.

```
<p>

Weiterer Text ...
...
...
</p>
```

Im Stylesheet erweitere ich hierzu die `img` Anweisung mit einem Punkt gefolgt von dem Klassen-Namen:

```
img.mitrahmen {
margin: 0 0 10px 10px;
padding: 5px;
float: right;
border: 1px solid #ccc;
}
```

Mit `float: right;` erreiche ich eine Ausrichtung nach rechts innerhalb des übergeordneten Elternelements, welches ein Absatz `<p>` oder auch ein Container `<div>` sein kann.

Mit der `Border`-Anweisung und dem `padding` von 5px erreiche ich eine Haarlinie (helles grau) um das gesamte Bild herum. Die Margin Angaben bewirken einen linken und unteren Abstand zum umlaufenden Text.

Ein Beispiel wie dies aussieht:

Dies ist eine h1 Überschrift

Dies ist mein Beispieltext, der innerhalb des Containers `#content` sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo. Dies ist mein Beispieltext, der innerhalb des Containers `#content` sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo.

Dies ist mein Beispieltext, der innerhalb des Containers `#content` sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo.

Ein netter Effekt mit dem ich gerne arbeite, ist eine kleine Vorschaugrafik mittels der Hover-Anweisung zu erweitern. Auf vielen Websites werden kurze Anrisse von News, Artikeln mit einer Teasergrafik (Vorschaubild) dargestellt.

Wäre es nicht schön, wenn sich diese Teasergrafik vergrößert, wenn man mit der Maus darüber fährt? Um das noch zu erweitern möchte ich eine noch größere Ansicht im Browserfenster öffnen, wenn ich auf die Grafik klicke (per Anker, Hyperlink).

Hierzu benötigte ich dann 2 separate Grafiken, eine größere Vorschaugrafik, z.B. 250 Pixel breit und eine Großversion 800 Pixel breit. Die große Version verlinke ich im Anker, Hyperlink und die kleine Version sitzt im img-Tag.

```
<p>
<a href="bilder/meinbildgross.jpg">

</a>
Weiterer Text ...
...
...
</p>
```

Erweitern bzw. schreiben wir das Stylesheet für das Bild noch wie folgt um.

```
#content a img.mitrahmen {
margin: 0 0 10px 10px;
padding: 5px;
float: right;
border: 1px solid #ccc;
width: 100px;
height: 80px;
}
```

Man beachte, dass ich hier die Vorschaugrafik (welche eigentlich 250 Pixel breit ist) auf eine fixe Breite von 100 Pixel und Höhe von 80 Pixel einzwänge. Bilder im Querformat werden dadurch Proportional dargestellt. Hochformatige Bilder werden allerdings gestaucht in der Vorschau angezeigt.

Jetzt benötige ich noch die Anweisung für den Hover-Effekt, somit wenn ich mit der Maus über das Vorschaubild fahre.

```
#content a img.mitrahmen:hover {
margin: 0 0 10px 10px;
padding: 5px;
float: right;
border: 1px solid #ccc;
width: 250px;
height: auto;
}
```

Mit width: 250px erweitere ich die Vorschaugrafik beim Hovern auf die eigentliche Größe. Die Festlegung der Höhe, per height: auto gesetzt, überlasse ich hierbei dem Browser. So werden dann auch hochformatige Bilder proportional dargestellt.

Ein Beispiel wie dies aussieht → ohne Maus (Hover) auf dem Bild:

Vom Aussehen hat sich nichts geändert.

Dies ist eine h1 Überschrift

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo. Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo.

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo.

Jedoch, wenn ich mit der Maus auf das Bild fahre ...

→ **Maus-Hover;**

Dies ist eine h1 Überschrift

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo. Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo.

Dies ist mein Beispieltext, der innerhalb des Containers #content sitzt. Lorem ipsum dolorum ignoto serum ada fadero del publo.

So kann man jetzt noch weitere Texte mit Bildern in Blöcken untereinander setzen. Bei diesem Verfahren sollte man beachten, dass die img-Tag mit dem Anker innerhalb des Absatzes <p> gesetzt wird.

Sicherlich kann man auch die <a> und Anweisung ausserhalb, z.B. vor dem p-Tag setzen. Bei mehreren Artikeln oder Textblöcken wird allerdings die nachfolgende Vorschaugrafik unschön links von dem Hoverbild verschoben, umgebrochen.

Da zerhaut es dann ein bissl das Aussehen. Um dies zuverhindern empfehle ich daher immer das Bild ins p-Tag rein zusetzen.

Zum guten Schluß

Umseitige CSS- und HTML-Formatierungen sollen bitte nur als Beispiele verstanden werden. Je nach Layout der Website können sich auch ganz andere Anforderungen für die Anordnung der Elemente ergeben.

Designtechnisch sind die Beispiele vielleicht auch nicht das Optimum. Ich wollte damit nur die Grob-Prinzipien und Möglichkeiten darstellen. Vielleicht auch als Erinnerungsstütze und Nachschlagewerk, da ich mir auch nicht alle Einzelheiten merken kann.

Im einzelnen habe ich auch nicht die beschriebenen CSS-Formatierungen im Browser getestet. Mag sein, dass sich evtl. hier und da einige Abweichungen ergeben.

Von der Typography her, kann man natürlich seine Überschriften und Texte entsprechend noch aufpeppeln. Hierzu eignen sich Anweisungen wie font-style, font-weight, font-size, letterspacing usw. und generell die Wahl von verschiedenen Schriftarten.