Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

HTML Texte

Stand: 21.07.2022

Text formatieren mit HTML
Inline-Elemente (Zeichenformate) für den Fließtext

Als `Inline`-Elemente werden in **HTML** Elemente bezeichnet, die keine neue Zeile im Textfluss bewirken und deren Breite sich nur über den Raum erstreckt, den der Inhalt einnimmt. In der hier empfohlenen `strict`-Variante
von HTML dürfen `Inline`-Elemente ebenso wenig wie nackter Text direkt im Dateikörper zwischen `<body>` und `</body>` vorkommen, sondern nur innerhalb von `Block`-Elementen oder bestimmten anderen
`Inline`-Elementen.

Erzwungener Zeilenumbruch

Genaugenommen ist der erzwungene Zeilenumbruch weder ein `Inline`-Element noch ein `Block`-Element, weil er gar keinen eigenen Inhalt und Erstreckungsbereich hat. Wir behandeln ihn jedoch an dieser Stelle, da er typischerweise innerhalb
von `Block`-Elementen vorkommt, um dort Zeilenumbrüche an gewünschten Stellen zu bewirken. Ein typischer Anwendungsfall sind Strophen eines Gedichts oder Songs, wie das nachfolgende Beispiel zeigt:

```
<p>Yesterday
<br>
All my troubles seemed so far away
<br>
Now it looks as though they're here to stay
<br>
Oh I believe in yesterday...</p>
```

Info Das alleinstehende Tag `<br>` bewirkt einen einfachen Zeilenumbruch. Falls Sie in XHTML schreiben, dürfen Sie nicht vergessen, dass dort `<br />` notiert werden muss.

Inline-Auszeichnungen mit semantischer Bedeutung

Mit der Zielsetzung, möglichst sinnvolles semantisches Markup zu erzeugen, sollten Sie die nachfolgenden Inline-Auszeichnungen bevorzugt benutzen.

Inline-Auszeichnung	Bedeutung
`<em>...</em>`	Betont, hervorgehoben

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Inline-Auszeichnung	Bedeutung
`<strong>...</strong>`	Noch stärker betont, besonders hervorgehoben
`<code>...</code>`	Quelltext-Fragment, z.B. aus Programmiersprachen
`<samp>...</samp>`	Beispiel, z.B. bei Ausgaben eines Programms oder Scripts
`<kbd>...</kbd>`	Tastatureingaben des Anwenders
`<var>...</var>`	Variable, verwendbar z.B. auch für Namen
`<cite="">...</cite>`	Zitat
`<q cite="">...</q>`	Quotation mit Zitathinweis
`<abbr>...</abbr>`	Als Buchstabenfolge gesprochene Abkürzungen
`<acronym>...</acronym>`	Als Wort gesprochene Abkürzungen

Info Einige dieser **Elemente** sind vor allem für Softwaredokumentation interessant, wie etwa die Elemente `code`, `samp`, `kbd` und `var`. Etwas Interpretationsfreiheit ist jedoch erlaubt.
Es spricht nichts dagegen, wenn Sie beispielsweise mit dem `var`-Element unverlinkte URL-Adressen auszeichnen oder auch Produktbezeichnungen. Wichtig ist, dass Sie solche Auszeichnungen durchgängig gleich und in allen Texten eines Projekts
konsequent verwenden.

Etwas erklärungsbedürftig sind die Elemente

`abbr` und `acronym`. Unter `abbr` fallen Abkürzungen wie „WWW“, „HTML“ oder „FBI“, also Abkürzungen, die sich nur Buchstabe für Buchstabe aussprechen lassen. Unter `acronym` fallen dagegen Ausdrücke wie „NASA“
oder „Modem“, also als Wort gesprochene Ausdrücke, die jedoch eigentlich eine Abkürzung darstellen. Wie ein Browser solche semantischen Auszeichnungen im Text darstellt, bleibt ihm überlassen. Die meisten modernen grafischen Browser stellen beispielsweise
Inhalte, die mit `<em>...</em>` ausgezeichnet sind, in Kursivschrift dar, `<strong>...</strong>` dagegen in Fettschrift, `<dfn>...</dfn>` oftmals

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

gepunktet unterstrichen und `<code>...</code>` in dicktengleicher Schrift. Verlassen sollten Sie sich jedoch in dieser Hinsicht auf nichts. Semantische Elemente schreien geradezu danach, mithilfe von CSS ordentlich
formatiert zu werden. Nachfolgend noch ein Beispiel, das den Einsatz semantischer `Inline`-Auszeichnungen zeigt:

```
<p><q cite="Heraklit">Alles fließt</q>, sprach <var
class="person_name">Heraklit</var>, für den die <dfn>arche</dfn>, also der
Urgrund allen Seins, in der Bewegung des ewig Werdenden lag. Genaugenommen
stammt der Leitspruch, dass alles fließt, zwar <em>nicht</em> von <var
class="person_name">Heraklit</var>, sondern wird ihm nur von
<var class="person_name">Platon</var> zugeschrieben - doch es drückt seine Lehre
so gut aus, dass man über diese kleine Ungenauigkeit gerne hinweg sieht.</p>
```

Info Das Beispiel zeigt, wie Sie mit Hilfe des universell einsetzbaren `class`-Attributs noch weitere semantische Genauigkeit im Text erzielen können.

Inline-Auszeichnung ohne semantische Bedeutung

Bei diesen Elementen die untenstehende Tabelle zeigt, handelt sich um solche, die direkt die **Textformatierung** beeinflussen.

Inline-Auszeichnung	Bedeutung
`<i>...</i>`	Kursiv
`<b>...</b>`	Fett
`<tt>...</tt>`	Dicktengleich
`<big>...</big>`	Größer als normal
`<small>...</small>`	Kleiner als normal
`<sub>...</sub>`	Tiefgestellt
`<sup>...</sup>`	Hochgestellt
`<span>...</span>`	Inline-Element ohne Wirkung

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Info Die Tabelle enthält ausschließlich Elemente, die im `strict`-Standard von HTML vorkommen. Dennoch gelten vor allem die beiden erstgenannten Elemente, also das `i`– und das b-Element, bei Hardlinern des semantischen
Markup als verpönt, weil sie direkte Formatieranweisungen darstellen. Bei Elementen wie dem `tt`-Element, das einfach nur die Verwendung dicktengleicher Schrift anweist, sind die Ansichten bereits geteilter, da seine mögliche Einsatzzwecke
nicht unbedingt durch das verfügbare Repertoire an semantischen Inline-Auszeichnungen abgedeckt werden kann. Und für Elemente wie `sub` und `sup` gibt es ohnehin keine semantische Entsprechung.
Besondere Beachtung verdient das `span`-Element.
Es bewirkt optisch rein gar nichts, wird aber vom Parser des Browsers registriert. Das `span`-Element eignet sich durch diese Eigenschaft besonders zur Formatierung mit **CSS**.

Auszeichnungen für Änderungsmarkierungen

Möglicherweise kennen Sie aus Textverarbeitungsprogrammen die Möglichkeit, Dokumente beim Bearbeiten mit Änderungsmarkierungen zu versehen. Dabei werden gelöschte Textpassagen meist rot und durchgestrichen dargestellt, eingefügte dagegen grün. Gedacht
sind solche Markierungen für den Austausch etwa für Review-Prozesse oder Lektorat. In HTML stehen diese Funktionen ebenfalls zur Verfügung: Mit `<ins>...</ins>` markieren Sie Textpassagen als neu eingefügt – `ins`steht für `insert` (zu Deutsch: einfügen) – und mit `<del>...</del>` können Sie Passagen zum Löschen markieren – `del` steht für `delete` (zu Deutsch: löschen). Mit Hilfe zweier Attribute
können Sie Auszeichnungen dieser Art außerdem datieren und begründen. Dazu ein Beispiel:

```
<ins datetime="2005-06-25T11:00+01:00"
cite="http://www.w3.org/TR/html4/appendix/changes.html#19991224">
Die Meta-Angabe für Refreshs wird von der HTML-Spezifikation mittlerweile nicht
mehr empfohlen. Stattdessen wird auf die Möglichkeit hingewiesen, serverseitige
Redirects einzusetzen.</ins>
```

Info Das `datetime`-Attribut erwartet eine Datum-Zeit-Angabe in der dargestellten Form (der Wert im Beispiel bedeutet: 25. Juni 2005, 11:00 Uhr bei plus 1 Stunde gegenüber Greenwich-Zeit). Die beim `cite`-Attribut angegebene
URL-Adresse verweist auf eine Stelle in der HTML-4.01-Spezifikation, in der Änderungen gegenüber der älteren HTML-4.0-Spezifikation aufgelistet werden. Die Elemente `ins` und `del` weisen übrigens eine Besonderheit auf: Obwohl sie
selbst zu den `Inline`-Elementen gehören, dürfen Sie sowohl `Block`– als auch weitere `Inline`-Elemente enthalten. Ein Konstrukt wie das Folgende ist also durchaus zulässig:

```
<ins>
<h3>Eine eingefügte Überschrift</h3>
<p>und ein eingefügter Absatz</p>
</ins>
```

Fließtext

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Widmen wir uns dem sichtbaren Dateikörper des HTML-Dokuments, also dem, was im Browser-Fenster angezeigt wird. Es handelt sich um alles, was zwischen `<body>` und `</body>` notiert wird.

HTML stellt ein ganzes Arsenal an Elementen bereit, um Inhalte ordentlich zu strukturieren. Aufgrund historischer Entwicklungen sind einige dieser Elemente heute veraltet oder sie gehören noch zum Standard, werden jedoch in der Praxis
nicht mehr verwendet. Wir werden uns in diesem Abschnitt daher auf eine Auswahl an solchen Elementen beschränken, die modernes HTML im Sinne von „semantischem Markup“ repräsentieren und praxisrelevant sind.

> Dabei sei noch mal ausdrücklich daran erinnert,

dass diese Elemente nur der inhaltlichen Strukturierung dienen. Stören Sie sich beim Ausprobieren nicht an der Darstellung Ihrer Versuche im Browser. Die Feinheiten der Darstellung werden später mithilfe von CSS realisiert. Es ist sogar eine durchaus
empfehlenswerte Vorgehensweise, bei inhaltslastigen Projekten nicht mit dem optischen Design zu beginnen, sondern mit dem logischen Markup in HTML. Wenn die Strukturierung der Inhalte stimmt, dann lässt sich die Struktur mithilfe von CSS auch in ein angemessenes
und schickes Layout bringen. In HTML wird zwischen `Block`– und `Inline`-Elementen unterschieden. Aus Textverarbeitungsprogrammen sind Ihnen möglicherweise Begriffe wie Absatz- und Zeichenformate geläufig. Diese Begriffe können zumindest
als Verständnishilfe dienen: `Block`-Elemente entsprechen Absatzformaten und `Inline`-Elemente entsprechen Zeichenformaten. Dass in HTML allerdings nicht von Absatz- und Zeichenformaten die Rede ist, sondern eben von `Block`–
und `Inline`-Elementen, hängt mit der strengeren Strukturierungsphilosophie von HTML und dem Boxmodell der Ergänzungssprache CSS zusammen. Ein `Block`-Element in HTML erstreckt sich über die gesamte verfügbare Breite und nimmt so
viel Höhe ein wie vom Inhalt her erforderlich. „Verfügbare Breite“ ist dabei relativ zu verstehen.

Logische Bereiche

Wenn Webseiten aus logisch unterscheidbaren Bereichen bestehen, die später auch optisch als eigener Bereich erkennbar erscheinen sollen, dann sollten Sie diese Bereiche im **Markup** von vorneherein berücksichtigen. Ein Komplettlisting soll
dies demonstrieren:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="de">
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
<title>Bereiche</title>
</head>
<body>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<div id="Navigation">
<a name="aktuelleSeite" id="aktuelleSeite">Startseite</a>
<a class="Navi" href="produkte.html">Produkte</a>
<a class="Navi" href="php/shop.php">Shop</a>
<a class="Navi" href="unternehmen.html">Unternehmen</a>
<a class="Navi" href="jobs.html">Jobs</a>
<a class="Navi" href="impressum.html">Impressum</a>
<a class="Navi" href="sitemap.html">Sitemap</a>
</div>
<div id="Inhalt">
<div id="Seitenkopf">
<h1>Willkommen!</h1>
</div>
<div id="Seitentext">
<p>Für unsere Kunden bieten wir Produktinfos mit Direktbestellmöglichkeit (Shop)
an. Unsere Geschäftspartner können sich über unser Unternehmen informieren.
Besuchen Sie auch unsere Job-Seite!</p>
</div>
</div>
</body>
</html>
```

Info Die Darstellung der Seite im Browser ist noch sehr spartanisch. Aber dennoch enthält der **HTML-Quelltext** bereits eine Menge in Hinblick auf ein späteres Seitenlayout, indem darin logische sinnvolle Bereiche definiert sind.
Auf der Hierarchieebene unterhalb des `body`-Elements sind zwei Bereiche definiert: einer für die Navigation, eingeleitet durch `<div id="Navigation">`, und einer für den Seiteninhalt (`<div id="Inhalt">`).
Mit `<div>...</ div>` werden in HTML Bereiche für logisch zusammenhängende Inhalte definiert. Das `div` steht für `division` (zu Deutsch: Abteilung, Bereich). Das Attribut `id=` mit
den frei wählbaren Namen ist nicht zwingend erforderlich. Doch da wir die Bereiche zu einem späteren Zeitpunkt mithilfe von CSS zum Errichten eines Seitenlayouts benötigen, setzen wir das Attribut jetzt bereits ein. Außerdem erhalten unsere Bereiche durch
die Namenszuweisung gleich ihre logische Bedeutung, was das Verständnis des Quelltextes verbessert.
Das `div`-Element mit dem `id`-Namen Inhalt besteht im Beispiel selber wieder aus zwei untergeordneten Bereichen, die jeweils eigene
Namen erhalten: Seitenkopf und Seitentext. Es ist also erlaubt und durchaus gängige Praxis, Bereiche ineinander zu verschachteln. Treiben Sie die Verschachtelung allerdings nicht auf die Spitze. Es gibt auch Webseiten, die sich „modern“ schimpfen, im
Grunde aber nur aus einer `div`-Orgie bestehen, die jeder logischen Nachvollziehbarkeit entbehrt. Auch in unserem Beispiel ist ein `div`-Bereich nicht wirklich zwingend logisch begründbar, nämlich der Bereich Seitenkopf. Da dieser
nichts weiter als eine Überschrift enthält, könnte man ihn auch weglassen und z.B. der Überschrift selber den `id`-Namen Seitenkopf geben. Im Beispiel sähe das so aus:

```
<h1 id="Seitenkopf">Willkommen</h1>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Dennoch haben wir entschieden, die Überschrift in einen `div`-Bereich einzuschließen. Der Grund ist, dass wir zu einem späteren Zeitpunkt noch eine Grafikreferenz mit in diesen Bereich packen wollen, nämlich ein Logo.

Größe, Abstände, Rahmen und das Boxmodell

Die Größe eines **Blockelements** mit Text richtet sich normalerweise nach der Menge an Text. Sie können allerdings die Breite exakt angeben. Dazu dient der CSS-Befehl `width`, der eine beliebige CSS-Maßeinheit akzeptiert:

```
<p style="width: 200px">Text im Absatz mit festgelegter Größe. Um dies zu sehen, muss genug Text vorhanden und das Browserfenster nicht zu klein sein.</p>
<p>Text im Absatz ohne festgelegte Größe. Um dies zu sehen, muss genug Text vorhanden und das Browserfenster nicht zu klein sein.</p>
```

Neben der Größe kann ein Blockelement auch einen Rahmen haben. Hierfür dienen die `border`-Befehle:

- `border-width` gibt die Rahmenstärke an.
- `border-color` gibt die Rahmenfarbe an. `transparent` ist ein durchsichtiger Rahmen.
- `border-style` steuert, wie der Rahmen aussieht. `dashed` ist gestrichelt, `dotted` gepunktet, `solid` durchgezogen, `double` eine doppelte Linie, `ridge` ist gewölbt und `groove` wirkt reliefartig. `none` bedeutet ohne Rahmen, `hidden`, dass der Rahmen unsichtbar ist.

Hier ein einfaches Beispiel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- rahmen.html -->
<html>
<head>
<title>Rahmen</title>
</head>
<body>
<p style="width: 200px; border-width: 1px; border-style: solid; border-color: black;">
Text im Absatz mit Rahmen außen.</p>
<p style="border-width: 5px; border-style: groove; border-color: blue;">
Text im Absatz mit anderer Rahmenart.</p>
</body>
</html>
```

Info Sie können für den **Rahmen** auch den Sammelbefehl `border` verwenden. Er enthält Stil, Breite und Farbe und zwar in beliebiger Reihenfolge, z.B.

```
border: 5px groove blue;
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Rahmen lassen sich auch für jede der vier Seiten einzeln festlegen. Dazu verwenden Sie die Befehle `border-left`, `border-top`, `border-right` und `border-bottom`. Diese vier Seitenangaben gibt es wiederum auch
in Verbindung mit den drei Einzelbefehlen. Um also z.B. die Dicke der rechten Seite festzulegen, verwenden Sie `border-right-width`. Sie können die vier Seiten allerdings auch in den normalen `border`-Befehlen angeben, indem Sie
zwei bis vier Werte hintereinander schreiben:

- Bei zwei Werten gilt der erste für oben und unten, der zweite für links und rechts.
- Bei dreien gilt der erste für oben, der zweite für links und rechts und der dritte für unten.
- Bei vieren geht die Reihenfolge oben, rechts, unten und links. Hier ein Beispiel:

```
border-color: red green blue red;
```

Bis jetzt „pappt“ der Text direkt am Rahmen. Dies können Sie allerdings mit dem Innenabstand des Blockelements steuern. Für den Innenabstand ist der **CSS-Befehl** `padding` zuständig. Den Abstand nach außen, das heißt vom Rahmen
zu umliegenden Elementen, regelt `margin`. Wie beim Rahmen können Sie auch hier jeweils alle vier Seiten einzeln steuern. Diese zwei bilden zusammen mit dem Rahmen das so genannte Boxmodell, das für jedes Blockelement gilt: Der Nullpunkt ist
die linke obere Ecke des Blockelements, an der es positioniert wird.
Info Das Boxmodell wird nicht in jedem Browser korrekt interpretiert. Der Internet Explorer 5.0 und 5.5 zählen zur Breite eines Elements Innenabstand, also `padding`, und Rahmen, also `border`, dazu. Das heißt, dass diese
Browser eine Box schmaler interpretieren als standardkonforme Browser. Der Internet Explorer 6 macht alles richtig, wenn ein XHTML-DOCTYPE angegeben ist. Wenn Sie Ihr Design völlig gleich gestalten wollen, können Sie einen so genannten Hack, also einen
CSS-technischen Trick, verwenden, um diesen Effekt zu vermeiden. Allerdings ist das in der Praxis nur bedingt empfehlenswert.

Aufzählungslisten

Darunter sind Listen zu verstehen, deren Listenpunkte mit einem so genannten **Bullet-Symbol** (mancherorts auch „Böller“ genannt) beginnen. Die HTML-Spezifikation redet einfach nur von „unsortierten Listen“ (im Gegensatz zu den nummerierten
Listen). Gedacht sind die Aufzählungslisten für alle Arten von zusammengehörigen Daten, Argumenten oder dergleichen, wobei die Reihenfolge keine Rolle spielt.

Eine einfache Aufzählungsliste hat in HTML folgendes Aussehen:

```
<ul>
<li>Listenpunkt</li>
<li>anderer Listenpunkt</li>
</ul>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Info Eingeschlossen wird die Liste in `<ul>...</ul>`. Das steht für unordered list (zu Deutsch: unsortierte Liste). Jeder einzelne Listenpunkt wird durch `<li>...</li>` markiert
– die Abkürzung bedeutet `list item` (zu Deutsch: Listeneintrag). Das `ul`-Element darf nichts anderes als `li`-Elemente enthalten, also keinen nackten Text oder andere Elemente außerhalb der `li`-Elemente.
Die `li`-Elemente dürfen dagegen Text, andere Block-Elemente und andere Inline-Elemente enthalten.
HTML-Quelltext dazu sieht folgendermaßen aus:

```
<ul>
<li>Listenpunkt</li>
<li>Unterliste:
<ul>
<li>Listenpunkt</li>
<li>Listenpunkt</li>
</li>
</ul>
```

Info Zu beachten ist dabei, dass die untergeordnete Liste als normaler Listenpunkt mit führendem Text beginnt und daran anschließend die komplette untergeordnete Liste notiert wird. Am Ende wird der Listenpunkt, der die untergeordnete Liste enthält,
ordnungsgemäß mit `</li>` abgeschlossen. Bei verschachtelten Listen wechseln die Browser üblicherweise das Bullet-Symbol. Um Einfluss darauf zu nehmen, welches Symbol angezeigt wird, müssen Sie **CSS** verwenden.

Nummerierte Listen

Nummerierte Listen ähneln vom syntaktischen Aufbau her den Aufzählungslisten. Der Unterschied besteht nur darin, dass sie mit `<ol>...</ol>` definiert werden, wobei `ol` für `ordered list` (zu Deutsch: geordnete Liste) steht. Die folgende Beispielnotation bewirkt eine einfache Nummerierung:

```
<ol>
<li>Aufstehen</li>
<li>Duschen</li>
<li>Frühstücken</li>
</ol>
```

Info Das Verschachteln von nummerierten Listen ist genauso möglich wie bei **Aufzählungslisten**. Die Hoffnung, dass dadurch automatisch Nummerierungen wie 1.1, 1.2 usw. entstehen, wird jedoch enttäuscht. Solche Automatisierungsmechanismen
unterstützt HTML leider nicht und nur durch spezielle CSS-Notationen lässt sich etwas Derartiges bewerkstelligen. Ebenfalls mit CSS lässt sich die Art der Nummerierung beeinflussen (z.B. alphabetische Nummerierung anstelle der numerischen).

Physische Auszeichnungen im Text

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

In HTML gibt es physische und logische Elemente zur Auszeichnung von Text. Physische Textauszeichnungen haben Bedeutungen wie „fett“ oder „kursiv“, stellen also direkte Angaben zur gewünschten **Schriftformatierung** dar. Bei physischen Elementen
sollte der Webbrowser eine Möglichkeit finden, den so ausgezeichneten Text entsprechend darzustellen. Ebenso wie die logischen Elemente zur Auszeichnung von Text, gehören die hier beschriebenen Elemente zuden Inline-Elementen.

HTML-Elemente für physische Auszeichnung im Text

Es stehen verschiedene HTML-Elemente zur Verfügung, um Text physisch auszuzeichnen. Inline-Elemente für Auszeichnungen im Text müssen – zumindest in der HTML-Variante „Strict“ – innerhalb anderer Blockelemente vorkommen. Im nachfolgenden Beispiel ist
ein Textabsatz notiert und innerhalb davon stehen mehrere physische Textauszeichnungen. Am Anfang des Textbereichs, der ausgezeichnet werden soll, wird ein einleitendes Tag (im Beispiel die Tag und) eingefügt.
Am Ende des gewünschten Textbereichs wird ein entsprechendes Abschluss-Tag eingefügt (z.B. bzw.).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Fett und schief</h1>
<p>Das Schwein ist <b>fett</b> und der Turm von Pisa ist <i>schief</i>.
<br>
Und was ist <b><i>fett und schief</b></i>?</p>
</body>
</html>
```

Info Physische Textauszeichnungen lassen sich auch kombinieren. Folgende Elemente dieser Art stehen zur Verfügung:

... Zeichnet einen Text als fett aus.

... Zeichnet einen Text als kursiv aus.

<tt>...</tt> Zeichnet einen Text als dicktengleich formatiert aus (tt = Teletyper = Fernschreiber).

<u>...</u> Zeichnet einen Text als unterstrichen aus.

<strike>...</strike> Zeichnet einen Text als durchgestrichen aus.

<s>...</s> Zeichnet einen Text als durchgestrichen aus.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

<big>...</big> Zeichnet einen Text als größer als normal aus.

<small>...</small> Zeichnet einen Text als kleiner als normal aus.

^{...} Zeichnet einen Text als hochgestellt aus.

_{...} Zeichnet einen Text als tiefgestellt aus.
Info Sie können auch – wie im obigen Beispiel – Elemente dieser Art verschachteln. Dadurch sollten sich die Effekte addieren. Ein Text, der also beispielsweise als fett und innerhalb davon als kursiv ausgezeichnet wird, sollte fettkursiv dargestellt
werden.

Physische Elemente für Textauszeichnung zusätzlich mit CSS formatieren

Mit **Stylesheets** können Sie auch physische Elemente nach weiteren Wünschen formatieren. Maßgeblich sind im hier beschriebenen Zusammenhang z.B. die CSS-Eigenschaften Schriftformatierung, Hintergrundfarben und -bilder.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Fett und schief mit Stil</h1>
<p>Das Schwert ist <b style="background-color:#FFCCCC">fett</b> und der Turm von
Pisa ist <i style="font-weight:bold; background.color:#FFFF00">schief</i>.</p>
</body>
</html>
```

Info Mit CSS erweitern Sie die Darstellung von Text um vielfältige Optionen, wie etwa eine farbliche Hinterlegung. Die beiden Elemente für fette und kursive Schriftauszeichnung erhalten im Beispiel zusätzlich unterschiedliche Hintergrundfarben.
Das `i` -Element wird außerdem fett ausgezeichnet – mit Hilfe von CSS.

Blöcke mit präformatiertem Text

Der Ausdruck „**präformatierter Text**“ ist etwas gewöhnungsbedürftig. Gemeint ist damit, dass Text im Browser so wiedergegeben wird wie im Editor eingegeben, mit allen Leerzeichen, Zeilenumbrüchen, Einrückungen usw. Dieses Verhalten entspricht
ja nicht den normalen Editierregeln von HTML. Manchmal ist ein solches Verhalten jedoch sehr wünschenswert, vor allem, wenn man in HTML-Texten Blöcke mit Programmlistings oder rein mit ASCII-Zeichen formatierte Tabellen oder Kunstwerke („ASCII-Art“)

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

im Text darstellen möchte.

In HTML können Sie solche Blöcke mit `<pre>...</pre>` auszeichnen. Dabei steht `pre` für *preformatted* (präformatiert). Alles, was Sie zwischen diesen beiden Tags eingeben, wird so interpretiert wie im Editor eingegeben. Beachten Sie, dass der Browser dabei eine dicktengleiche Schriftart wie z.B. Courier verwendet, also eine Schriftart, bei der alle Zeichen die gleiche Breite haben. Denn nur so lassen sich viele bei präformatiertem Text gewünschte
Effekte überhaupt sinnvoll darstellen. Ein Beispiel:

```
/-----------------------------------------------\
|          |        Average    | other    |  Misc  |
|          |-------------------| category |--------|
|          |  height |  weight |          |        |
|-----------------------------------------|--------|
| males    | 1.9     | 0.003   |          |        |
|-----------------------------------------|--------|
| females  | 1.7     | 0.002   |          |        |
\-----------------------------------------------/
```

Info Diese mit Strichen und positioniertem Text „gemalte“ Tabelle können Sie in HTML mithilfe des `pre`-Elements so wie sie ist im Browser repräsentieren. Das `pre`-Element darf durchaus auch andere **HTML-Elemente**enthalten. Nicht selten wird das `pre`-Element auch dazu benutzt, um HTML-Quelltext im Browser darzustellen. Dabei müssen Sie jedoch wie bereits gelernt penibel darauf achten, dass Sie alle HTML-eigenen Zeichen maskieren. Andernfalls würde
der Browser den HTML-Code interpretieren statt ihn darzustellen. Wir zeigen das Ganze am Beispiel der zuvor behandelten nummerierten Listen:

```
<ol>
<li>Aufstehen</li>
<li>Duschen</li>
<li>Frühstücken</li>
</ol>
```

Info Im Browser wird dann der HTML-Quelltext der nummerierten Liste angezeigt, mit allen Einrückungen und Zeilenumbrüchen wie eingegeben.

Sonstige Block-Elemente für den Fließtext

Ergänzend zu den Standard-Blockelementen für Überschriften, Textabsätze, Listen und präformatierten Text bietet **HTML** noch ein paar weitere `Block`-Elemente zur Textgestaltung an, die bei konsequenter Verwendung zu einem reicheren
semantischen Markup beitragen: Durch `<blockquote>...</blockquote>` markieren Sie ein Zitat, das als eigener Absatz dargestellt werden soll. Die meisten Browser stellen den Text eines so ausgezeichneten Inhalts eingerückt
dar, was viele Möchtegern-Designer dazu verleitet hat, das `blockquote`-Element dazu zu verwenden, um

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Einrückungen jeder Art im Text zu realisieren. Solche Zweckentfremdungen, die auf eher zufälligen Default-Darstellungsweisen von Browsern
beruhen, gehören aus heutiger Sicht zu den schlimmsten Verbrechen an HTML, die man begehen kann. Hüten Sie sich also davor, mit dem `blockquote`-Element irgendetwas anderes auszuzeichnen als ein alleinstehendes Zitat, und rücken Sie Texte mithilfe
von CSS ein! Falls Sie im HTML-Quelltext auch die Quelle eines Zitats nennen möchten, so steht dafür das Attribut `cite` zur Verfügung. Ein Beispiel:

```
<blockquote cite="Cluetrain Manifest, http://www.cluetrain.de/, These 7"
title="aus dem Cluetrain Manifest, http://www.cluetrain.de/">
Hyperlinks untergraben Hierarchien.
</blockquote>
```

Info Das `cite`-Attribut hat jedoch keine sichtbare Wirkung im Browser. Deshalb haben wir im Beispiel außerdem noch das `title`-Attribut notiert, damit der Anwender die Quelle des Zitats erfährt, wenn er mit der Maus über
den Text fährt. Ein weiteres `Block`-Element stellt HTML mit `<address>...</address>` zur Verfügung. Dieses Element ist zur Auszeichnung von Autorenangaben mit Kontaktmöglichkeit gedacht, z. B.:

```
<address>
Name Vorname, info@email.de
</address>
```

Info Drei weitere `Block`-Elemente sind schließlich für den Texttyp „Glossar“ vorgesehen. Ein Glossar ist eine Art Liste, deren Einträge jeweils aus einem Glossareintrag („Term“) und einer Definition bestehen. In HTML sieht das wie im
folgenden Beispiel aus:

```
<dl>
<dt>dl-Element</dt>
<dd>dient zur Markierung einer Glossarliste</dd>
<dt>dt-Element</dt>
<dd>dient zur Markierung eines Terms im Glossar</dd>
<dt>dd-Element</dt>
<dd>dient zur Markierung einer Definition im Glossar</dd>
</dl>
```

Info Das Beispiel erklärt sich bereits selbst: Mit dem `dl`-Element schließen Sie die gesamte Glossar-Liste ein. Innerhalb davon werden abwechselnd je ein `dt`– und ein `dd`-Element notiert. Das `dt`-Element markiert einen Glossareintrag und das anschließende `dd`-Element die zugehörige Definition. Erwähnt werden soll abschließend noch das `hr`-Element: Durch Notieren von `<hr>` (in XHTML: `<hr />`)
fügen Sie eine horizontale Trennlinie ein. Dies kann sinnvoll sein, um logische Textbereiche voneinander zu trennen.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Textabsätze definieren

Um „Absatzschaltungen“ in HTML zu erzeugen, müssen Sie das dafür vorgesehene Element verwenden. Schlampereien der Sorte „zweimal Enter eingeben und Leerzeile erzeugen“ funktionieren nicht, da der Zeilenvorschub in **HTML** als Whitespace-Zeichen
gilt und mehrere Whitespace-Zeichen einfach zu einem Leerzeichen zusammengefasst werden. Gewöhnen Sie sich deshalb von vorneherein an, für Absätze in längerem Fließtext das p-Element zu benutzen:

```
<p>Hier der Text des Absatzes.</p>
<p>Und hier der nächste Absatz.</p>
```

Info Das p steht für **paragraph** (zu Deutsch: Absatz). Wenn Sie wie empfohlen die `strict`-Variante von HTML verwenden, müssen übrigens alle Inhalte zwischen <body> und </body>in `Block`-Elemente eingeschlossen sein. Es ist also nicht erlaubt, zwischen <body> und </body> an irgendeiner Stelle direkt Text ohne umgebendes `Block`-Element zu notieren. Den Browsern sind
solche Fehler zwar egal, doch die Validität des HTML-Dokuments geht durch solche Fehler unnötigerweise verloren.

> Wenn Sie also sonst keine Strukturelemente im Dateikörper benötigen,

beispielsweise, weil Sie eine reine Textdatei in HTML abbilden wollen, dann verwenden Sie dort, wo in der Textdatei Absätze durch Leerzeilen simuliert werden, das p-Element. Für das p-Element wie für alle weiteren noch vorgestellten
Elemente gilt wie für Überschriften: Die Default-Darstellung im Browser-Fenster können Sie später mit CSS beliebig ändern. Bisher haben Sie Text, der im Anzeigebereich des Browsers ausgegeben werden soll, einfach in den Gültigkeitsbereich des `body`-Elements
geschrieben. In früheren HTML-Versionen war dies auch vollkommen in Ordnung. Seit der Version 4.0 (mit der wir uns befassen) ist dies aber nicht mehr HTML-konform. Zwar stellen die gängigsten Browser einen solchen Text fehlerfrei dar, dies bedeutet aber
nicht, dass es kein Fehler ist, da Browser in der Regel über eine sehr hohe Fehlertoleranz verfügen. Anstatt den Text einfach im `body`-Element zu notieren, wird dafür das p-Element (p = engl. paragraph, dt. Absatz) verwendet. Dieses Element bietet auch sogleich einige Vorteile:

- Nach einem p-Element ist ein deutlicher Absatz zum folgenden Element zu erkennen.
- Mit einem speziellen Attribut lässt sich der Text wie in Textverarbeitungsprogrammen links, rechts, zentriert oder als Blocksatz ausrichten.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- Beispiel für das p-Element -->
<html>
<head>
<title>Absatz</title>
</head>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<body>
Textabsatz
<p>Dieser Text steht in einem p-Element.</p>
<p>Dies ist ein weiterer Text in einem zweiten Absatz.</p>
</body>
</html>
```

Info Wenn Sie das Beispiel mit den Zeilenumbrüchen in Ihren Editor übertragen, werden Sie bei der Darstellung im Browser zwei Dinge erkennen: den deutlichen Abstand zwischen beiden Absätzen und – obwohl der Text im zweiten Absatz im HTML-Dokument
in drei Zeilen steht – dass der Browser den Text in nur einer Zeile darstellt. Diesen Effekt werde ich Ihnen an späterer Stelle noch einmal genauer erklären.

Absätze aufmöbeln

Bisher bezogen sich alle **Formatierungen** direkt auf den Text. Sie können aber auch die **Blockelemente** selbst formatieren und den Text in den Blockelementen ausrichten. Dies zeigen wir Ihnen hier hauptsächlich am Beispiel
von Absätzen. Das Gesagte zum Ausrichten und zu den Rahmen gilt allerdings auch für andere Blockelemente wie Überschriften.

Ausrichten

Text im **Absatz** lässt sich recht einfach ausrichten. Dazu dient das Attribut `align` im <p>-Tag. Dieses Attribut kennen allerdings auch andere Blockelemente wie Überschriften und `<div>`-Elemente.
`align` kann folgende Werte annehmen:

- `left` für linksbündig. Dies ist der Standardwert.
- `center` für zentriert
- `right` für rechtsbündig
- `justify` für Blocksatz. Dieser klappt allerdings in älteren Browsern nicht und führt oftmals zu komischen Ergebnissen, da Browser nicht die Darstellungsqualitäten von Satzprogrammen besitzen und so unschöne Lücken entstehen, da HTML keine
  Silbentrennung beherrscht.

Hier ein Beispiel für den Einsatz:

```
<p align="right">Text im Absatz mit rechtsbündiger Ausrichtung. Um dies zu
sehen, muss genug Text vorhanden und das Browserfenster kleiner sein.</p>
<p align="justify">Text im Absatz mit Blocksatz. Um dies zu sehen, muss genug
Text vorhanden und das Browserfenster kleiner sein.</p>
```

In CSS gibt es als Entsprechung zum **HTML-Attribut** `align` den Befehl `text-align`. Er akzeptiert dieselben Werte wie `align`.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<p style="text-align: right">Text im Absatz mit rechtsb ündiger Ausrichtung. Um
dies zu sehen, muss genug Text vorhanden und das Browserfenster kleiner
sein.</p>
<p style="text-align: justify">Text im Absatz mit Blocksatz. Um dies zu sehen,
muss genug Text vorhanden und das Browserfenster kleiner sein.</p>
```

Die CSS-Version ist vorzuziehen, da das HTML-Attribut vom W3C als `deprecated` gekennzeichnet ist. Praktische Unterschiede gibt es allerdings ansonsten nicht.
Info Achtung, im Internet Explorer richtet `text-align` auch verschachtelte Blockelemente aus. Dies ist aber ein fehlerhaftes Verhalten und funktioniert z.B. im Mozilla Firefox nicht.

Einzüge und spezielle Formate

Für Absätze gibt es noch einige Besonderheiten, um Einzüge, die erste Zeile und den ersten Buchstaben zu formatieren. Für einen Einzug der ersten Zeile ist der **CSS-Befehl** `text-indent` zuständig:

```
<p style="text-indent: 10pt;">Text im Absatz mit Einzug. Den Effekt sehen Sie
mit mehr Text.</p>
<p>Text im Absatz ohne Einzug. Den Effekt sehen Sie mit mehr Text.</p>
```

Um speziell die erste Zeile oder den ersten Buchstaben zu formatieren, gibt es zwei so genannte Pseudoklassen. Das sind CSS-Klassen mit vorgegebener Funktionalität. Sie erkennen sie an dem Doppelpunkt statt dem Punkt vor dem Klassennamen. Für diese Klassen
können Sie dann beliebige CSS-Befehle festlegen. Zur Verfügung stehen:

- `:first-letter` für den ersten Buchstaben eines Absatzes.
- `:first-line` für die erste Zeile. Dies ergibt gerade in Verbindung mit `text-indent` interessante Effekte.

Hier ein Beispiel, das alle drei Befehle kombiniert. Allerdings interpretiert der Internet Explorer dies nicht korrekt:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- first_letter_line.html -->
<html>
<head>
<title>Pseudoklassen und Absätze</title>
<style type="text/css"><!--
p {
font-family: Arial, Geneva, Helvetica, sans-serif;
font-size: 10pt;
text-indent: 10px;
}
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
p:first-line {
font-family: Courier, monospace;
color: blue;
}
p:first-letter {
font-size: 30pt;
line-height: 10pt;
color: red;
}
--></style>
</head>
<body>
<p>Pseudoklassen erlauben die Hervorhebung des ersten Buchstabens und der ersten Zeile.Den Effekt sehen Sie mit mehr Text.</p>
</body>
</html>
```

Info Für `first-letter` ist hier die Zeilenhöhe angegeben, da der **Zeilenabstand** durch den größeren Buchstaben sonst für die ganze Zeile zu hoch wird.

Textausrichtung

Standardmäßig wird der mit Elementen der **Textstrukturierung** ausgezeichnete Text linksbündig ausgerichtet. Das heißt, dass ein Text, der sich über mehrere Zeilen erstreckt, an einem imaginären Rand auf der linken Seite ausgerichtet ist
und eine vertikale Linie bildet. Um dies zu beeinflussen, gibt es das Attribut `align` und vier Parameter, um den Text, wie in den meisten Textverarbeitungsprogrammen auch, linksbündig, rechtsbündig, zentriert oder als Blocksatz auszurichten.
Die entsprechenden Parameter lauten:

Wert	Erklärung
`left`	Richtet den Absatz am linken Seitenrand aus.
`right`	Richtet den Absatz am rechten Seitenrand aus.
`center`	Zentriert den Absatz in der Seitenmitte.
`justify`	Richtet den Absatz als Blocksatz aus.

Info Nun können wir das bereits bekannte p-Element mit einer neuen **Ausrichtung** versehen.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- Textausrichtung //-->
<html>
<head>
<title>Textausrichtung</title>
</head>
<body>
<p align="left">Dieser Absatz ist am linken Seitenrand ausgerichtet.</p>
<p align="right">Dieser Absatz ist am rechten Seitenrand ausgerichtet.</p>
<p align="center">Dieser Absatz ist zentriert.</p>
<p align="justify">Dieser Absatz wurde mit Hilfe des align-Attributs als
Blocksatz ausgerichtet und wurde mit Absicht mit einem längeren Text versehen,
um den Parameter justify auch beispielhaft darstellen zu können. Dieser Absatz
wurde mit Hilfe des align-Attributs als Blocksatz ausgerichtet und wurde mit
Absicht mit einem längeren Text versehen, um den Parameter justify auch
beispielhaft darstellen zu können.</p>
</body>
</html>
```

Info Bei der Verwendung des Parameters `justify` ist eine wichtige Tatsache zu berücksichtigen: Damit der Absatz auch wirklich als Blocksatz ausgerichtet wird, muss er sich mindestens über zwei Zeilen erstrecken. Ansonsten wäre zum Parameter
`left` kein optischer Unterschied erkennbar. Berücksichtigen Sie dies, falls Sie der Meinung sind, dass der Absatz vom Browser nicht korrekt ausgerichtet dargestellt wird.

Trennlinien definieren

Trennlinien dienen der optischen Abgrenzung von nicht unmittelbar zusammengehörigen Textabschnitten oder allgemein zur Auflockerung. Eine **Trennlinie** erzeugt einen eigenen Absatz.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<p>Hier ist ein Abschnitt zu Ende.</p>
<hr>
<p>Und hier beginnt etwas Neues.</p>
</body>
</html>
```

Info
<hr> fügt eine Trennlinie ein (hr = horizontal rule = Querlinie). Dabei ist es egal, ob das Tag am Ende der

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Zeile des vorherigen Absatzes steht oder in einer eigenen Zeile (wie im Beispiel), oder am Anfang des folgenden Absatzes.
Wenn Sie XHTML-konform arbeiten, müssen Sie das `hr`-Element als inhaltsleer kennzeichnen. Dazu notieren Sie das allein stehende Tag in der Form `<hr />`. Mit Hilfe diverser HTML-Attribute im `<hr>`-Tag können
Sie eine Trennlinie auffälliger gestalten. Diese Attribute sind allerdings allesamt als deprecated eingestuft und sollen künftig aus dem HTML-Standard entfallen. Empfohlen wird die Gestaltung von Trennlinien mit CSS.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<p>Hier ist ein Abschnitt zu Ende.</p>
<hr noshade width="300" size="3* align="left">
<p>Und hier beginnt etwas Neues.</p>
</body>
</html>
```

Info Durch das Attribut `noshade` erreichen Sie, dass der Browser die Trennlinie massiv und durchgezogen, also nicht schattiert anzeigt (noshade = unschattiert). Wenn Sie XHTML-konform arbeiten, müssen Sie das Attribut in
der Form `noshade="noshade"` notieren, da XML-basierte Sprachen keine Attribute ohne Wertzuweisung erlauben. Durch das Attribut `width=` (width = Breite) erreichen Sie, dass der Browser die Trennlinie so breit anzeigt
wie angegeben. Sie können eine Zahl oder einen Prozentwert angeben. Mit einer Zahl, z.B. 300, erzwingen Sie, dass die Trennline so viel Pixel breit dargestellt wird wie angegeben. Mit einem Prozentwert erreichen Sie, dass die Trennlinie maximal so viel
Breite des Anzeige-Fensters einnimmt wie angegeben. Für eine prozentuale Angabe notieren Sie hinter der Zahl einfach ein Prozentzeichen (%). Durch das Attribut `size=` (size = Größe) können Sie die Höhe (Dicke) der Trennlinie
bestimmen. Die Voreinstellung beträgt 2 Punkt. Mit dem Wert 1 erzwingen Sie also eine besonders dünne Trennlinie, mit Werten über 2 können Sie dickere als die normalen Trennlinien erzeugen. Mit `align="1eft"` erreichen Sie, dass die Trennlinie
linksbündig ausgerichtet wird (align = Ausrichtung left = links). Mit `align="right"` wird die Trennlinie rechtsbündig ausgerichtet (right = rechts) und mit `align="center"` zentriert (Voreinstellung). Das Ausrichten
von Trennlinien ist allerdings nur in Verbindung mit dem Attribut `width=` sinnvoll, da die Trennlinie sonst stets über die gesamte verfügbare Breite geht.

Trennlinien mit CSS gestalten

Das `<hr>`-Tag können Sie auch mit CSS gestalten. Das ist strikt HTML-konform und Sie haben noch

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

deutlich mehr Gestaltungsmöglichkeiten als mit **HTML-Attributen**. Maßgeblich sind im hier beschriebenen Zusammenhang z.B.
folgende CSS-Eigenschaften: Ausrichtung und Absatzkontrolle, Außenrand und Abstand, Rahmen, Positionierung und Anzeige von Elementen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<p>Hier ist ein Abschnitt zu Ende.</p>
<hr noshade style="width:300px; color:yellow; height:5px; text-allign:left;
border:1px solid blue;">
<p>Hier beginnt etwas Neues.</p>
</body>
</html>
```

Info Mit CSS lassen sich Trennlinien weitgehend beeinflussen – allerdings ist der Effekt nur in den neusten Browser-Versionen sichtbar. Das `hr`-Element erhält im Beispiel eine Breite von 300 Pixeln, eine Höhe von 5 Pixeln, wird linksbündig
ausgerichtet, wird in gelber Farbe dargestellt und erhält noch einen dünnen blauen Rahmen drum herum.

Webseiten mit Text und Links

Wer an seiner ersten Webseite bastelt, beginnt meist mit ein wenig Inhalt. Das mag ein Text über sich selbst, die eigene Firma oder das Hobby sein. Wie Sie diesen Text in **HTML** realisieren und mit **CSS** garnieren, sprich
verschönern, erfahren Sie in diesem Artikel.

Sobald der Text fertig ist, wird es sehr schnell notwendig, auf andere Seiten zu verweisen. Sei es, dass die eigene Website aus mehreren Seiten bestehen soll oder dass eine befreundete Website nach einem Link verlangt. Vorsicht bei Links nach außen.

Formatierung

Text landet in HTML, wie Sie schon in CSS-Einführung lesen konnten, immer in einem Blockelement wie z.B. einem `<div>`-Block. Das gebräuchlichste Blockelement sind Absätze, also das `<p>`-Tag, wobei `p` für Paragraph (engl. für Absatz) steht.

Sie können entweder den gesamten Absatz oder auch nur Bereiche eines Textes formatieren. Für Teilformatierungen bieten sich eigene Formatiertags wie `<b>` für fetten Text an. Wollen Sie per CSS formatieren, greifen Sie zum `<span>`-Tag, um einen Textbereich zu markieren.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Schrift

Für die Schriftformatierung gab es früher, also vor der Einführung von CSS, das so genannte `<font>`-Tag. Dies sollten Sie aus zwei Gründen nicht mehr verwenden:

- Zum einen ist es vom HTML-Gralshüter W3C als `deprecated`, also nicht mehr empfohlen gekennzeichnet.
- Zum anderen kommt wirklich jeder relevante Browser mit den viel besseren CSS-Befehlen zur Schriftformatierung zurecht.

Die wichtigste **Schriftformatierung** ist sicherlich die Schriftart, denn sie hat vielleicht die größte Auswirkung auf das optische Erscheinungsbild. Bei der Schriftart gibt es im Web eine große Einschränkung: Der Nutzer muss die Schrift
besitzen, die Sie verwenden. Sie haben also nicht die freie Wahl aus allen auf Ihrem System installierten Schriften, sondern Sie müssen eine der auf (fast) allen Systemen vorhandenen Schriften wie Times New Roman, Arial, Helvetica, Verdana, Tahoma, Courier
etc. verwenden.
Info Downloadbare Schriften sollten dieses Problem beseitigen. So gab und gibt es sowohl von Bitstream mit TrueDoc eine Lösung für Netscape 4.x als auch von Microsoft die Web Embedded Fonts für den Internet Explorer. Beide Lösungen sind aber jeweils
auf die genannten Browser beschränkt und völlig inkompatibel. Insofern ist das Thema downloadbare Schriften heute tot. Wenn Sie eine spezielle Schrift einsetzen möchten, können Sie das in der Praxis nur, indem Sie sie in eine Grafik umwandeln. Die Schriftart
geben Sie in CSS mit dem Befehl `font-family` an. Sie können mehrere **Schriftarten** durch Kommata getrennt angeben. Der Browser sucht dann nach der ersten; findet er sie nicht, nimmt er die zweite und so weiter. Außer einer Schriftart
können Sie auch eine Schriftfamilie nehmen. Das ist eine bestimmte Art von Schriften. Die genaue Schrift legt dann der Browser bzw. das System fest. Tabelle zeigt die möglichen Schriftfamilien.

Schriftfamilie	Schrift	Beschreibung
`sans-serif`	Helvetica, Arial, Verdana, Geneva	Schriften ohne Serifen, das heißt keine Verästelungen an den Enden der Buchstaben
`serif`	Times New Roman, Garamond, Georgia	Schriftart mit Serifen
`monospace`	Courier, Everson Mono	diktengleiche Schriftarten; alle Buchstaben sind gleich breit
`cursive`	Snell Roundhand, Adobe Poetica	handschriftenähnlich, mit verbundenen Buchstaben oder anderen kursiven Charakteristika
`fantasy`	Cottonwood, Alpha Geometrique	außergewöhnliche Schriften z.B. mit Symbolen oder für spezielle Einsatzzwecke

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Hier ein Beispiel für das Festlegen von Schriftarten:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- font_family.html -->
<html>
<head>
<title>Schriftart</title>
</head>
<body>
<p style="font-family: Verdana, Arial, Helvetica, sans-serif;">
Text mit serifenloser Schrift.</p>
<p style="font-family: Courier, monospace;">
Diktengleiche Schrift z.B. für Quellcode-Darstellung.</p>
</body>
</html>
```

Info Im Web sind hauptsächlich serifenlose Schriften wie Arial und Verdana gebräuchlich, da sie gut lesbar sind. Vor allem Verdana ist zurzeit in. Zusammengesetzte Schriftnamen wie Times New Roman werden in Anführungszeichen geschrieben. Wenn Sie
die **CSS-Befehle** im `style`-Attribut in doppelte Anführungszeichen setzen, müssen Sie natürlich einfache Anführungszeichen verwenden.

Schriftgröße und -farbe

Wenn Sie die Schriftart festgelegt haben, können Sie noch Schriftgröße und -farbe festlegen. Für die Schriftgröße ist `font-size` zuständig. Wichtig ist die Einheit, die direkt nach der Zahl (ohne Leerzeichen) folgt. Sie haben hier wie bei
jeder Maßangabe alle **CSS-Maßeinheiten** zur Auswahl. Diese unterscheiden sich in absolute Größen und relative Größen. Dies sind die absoluten Größen:

- `cm` für Zentimeter.
- `mm` für Millimeter.
- `in` ist ein Zoll oder Inch. Ein Inch entspricht 2,54 cm.
- `pt` ist ein Punkt und entspricht 1/72 Inch. Diese Einheit kommt in diesem Artikel für Schriftgrößen zum Einsatz.
- `pc` ist ein Pica und entspricht 12 Punkt.

Und hier die relativen Größen:

- `px` entspricht einem Pixel, also einem Bildschirmpunkt. Die Größe eines Pixels ist abhängig von der Auflösung des Bildschirms (sie reicht im Allgemeinen von 72 bis 96 Pixel pro Inch). Diese Maßeinheit eignet sich hervorragend für Positionierungs-
und Größenangaben. Bei Schriftarten ist sie nicht sinnvoll, da der Internet Explorer Schrift mit Pixelangaben nicht skalieren kann.
- `em` bestimmt die Schriftgröße relativ zur aktuellen Größe. Messgrundlage ist die Schrifthöhe einer

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

bestimmten Schriftart. Ist diese Schrift 14 pt hoch, entspricht dies 1em.
- ex bestimmt die Schriftgröße relativ zur Größe des Buchstabens x. 1 ex entspricht meist einem halben em.
- % für Prozentangaben abhängig von der Fenstergröße oder bei Schriftarten von der normalen Schriftgröße. Wird für Schrift kaum verwendet.

Die Schriftfarbe steuern Sie mit dem `color`-Befehl. Hier ist wichtig zu wissen, wie im Web Farben angegeben werden. Die Grundlage der Farben im Web beziehungsweise auf dem Monitor ist das RGB-Farbmodell. RGB steht für Rot, Grün und Blau. Diese
drei Grundfarben bilden zusammen alle darstellbaren Farben. Eine Farbe setzt sich dabei aus Helligkeitsanteilen von Rot, Grün und Blau zusammen – deswegen nennt man das RGB-Modell auch `additiv`. Von jeder der drei **Grundfarben**gibt es insgesamt 256 Helligkeitsstufen, die von den Zahlen 0 bis 255 dargestellt werden. 255 Rot, 0 Grün und 0 Blau wäre dementsprechend reines Rot. 255 von jeder Grundfarbe ergibt zusammen Weiß, 0 von jeder Grundfarbe ergibt Schwarz.
Info Früher sprach man von 216 websicheren Farben. Websicher deswegen, weil diese Farben sowohl unter Windows als auch am Mac vorhanden waren, wenn der Monitor oder die Grafikkarte nur 256 Farben darstellen konnte. Hätte man andere Farben verwendet,
wären diese zur jeweils nächstähnlichen Farbe verschoben worden, was oft recht psychedelische Effekte hatte. Heutzutage unterstützen fast alle Monitore mehr Farben und die websicheren Farben sind in der Versenkung verschwunden. Die Farbkombinationen lassen
sich in CSS auf folgende Arten schreiben:

- Als **RGB-Farbe** mit dem Farbwert in dezimalen Zahlen:

```
color: rgb(255, 0, 0);
```

- Als **RGB-Farbe** in hexadezimaler Schreibweise. Diese Schreibweise wird auch von HTML verwendet:

```
color: #FF0000;
```

oder bei jeweils zwei gleichen Ziffern auch:

```
color: #F00;
```

Die hexadezimale Schreibweise basiert auf einem Zahlensystem mit der Basis 16 statt mit der Basis 10. Für die Zahlen 10 bis 15 kommen die Buchstaben von A bis F hinzu. Dadurch lässt sich jede Helligkeitsstufe in zwei Ziffern schreiben. Die Umrechnung
erfolgt so: Sie nehmen den RGB-Wert und teilen ihn durch 16. Das ganzzahlige Ergebnis ist die erste Ziffer. Bei 207 ergibt dies 12, also C. Der ganzzahlige Rest der Division ist die zweite Ziffer. Zum Ausrechnen multiplizieren Sie 12 mit 16, was 192 ergibt,
und ziehen das von 207 ab. Die zweite Ziffer ist also 15 und damit F. Daraus ergibt sich, dass 207 in hexadezimaler Schreibweise CF entspricht. Hierzu können Sie natürlich auch den Taschenrechner des Betriebssystems verwenden.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

- Als **Farbname**. CSS sieht sechzehn standardisierte englischsprachige Farbnamen vor wie z.B. `red` für Rot, `blue` für Blau und `green` für Grün. Dazu gibt es noch viele andere Farbnamen, die viele Browser
  unterstützen.

```
color: red;
```

Info
`color` ist der einzige Befehl direkt zur Schriftformatierung, der ohne vorangestelltes `font` auskommt. Hier nun ein Beispiel, das Schriftgröße und Farbangabe miteinander verbindet:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- font_size_color.html -->
<html>
<head>
<title>Schriftart</title>
</head>
<body>
<p style="font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10pt;
color: red;">
Text mit serifenloser Schrift.</p>
<p style="font-family: Courier, monospace; font-size: 2em; color: rgb(255, 51,
76);">
Diktengleiche Schrift z.B. für Quellcode-Darstellung.</p>
</body>
</html>
```

Fett und kursiv

Die Schriftdicke und die Schrägstellung von Textpassagen gehören zu den wichtigsten Auszeichnungsmerkmalen für einzelne Textbereiche. Fett und kursiv gibt es dementsprechend nicht nur in Word, sondern auch in HTML und CSS. Hier handelt es sich um einen
der Fälle, wo die **HTML-Befehle** noch genauso gebräuchlich sind wie die CSS-Pendants. Wir zeigen Ihnen deswegen beide: In HTML verwenden Sie das <b>-Tag, um einen Bereich fett hervorzuheben, und das `<i>`-Tag,
um ihn kursiv zu stellen. Beide lassen sich natürlich auch beliebig kombinieren. Allerdings sollten Sie sie XHTML-konform nur in Blockelementen wie einem Absatz, einem `<div>`-Tag oder einer Tabelle verwenden.

```
<p>Treffen am <b>3.7.2020</b> in <i>New York</i>.</p>
```

In CSS ist für die Schriftdicke `font-weight` verantwortlich. Dieser Befehl verwendet entweder die absolute Angabe `bold` für fett, `relative` Angaben wie `bolder` (dicker) und `lighter` (dünner)

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

oder die Zahlen von 100 bis 700, wobei 400 eine normale Schrift darstellt. In der Praxis ist zu 99 % bold im Einsatz, da die anderen je nach Schriftschnitt keine Auswirkungen haben.

`font-style` sorgt mit dem Wert `italic` für kursive Schrift. Hier ein zur vorherigen HTML-Variante funktionsgleiches Beispiel:

```
<p>Treffen am <span style="font-weight: bold">3.7.2020</span> in <span
style="font-style: italic">New York</span>.</p>
```

Info Es gibt für `font-style` noch den Wert `oblique`, der eine Schrift ebenfalls kursiv stellt. Der Teufel steckt hier im Detail: `italic` verwendet nach Möglichkeit eine kursive Version der Schrift auf dem System (einen so genanten Schriftschnitt). Nur wenn keiner vorhanden ist, wird die Schrift per Berechnung kursiv gestellt. `oblique` berechnet automatisch und ignoriert kursive Schriftschnitte.

Typographische Befehle

Typographie ist die Lehre von Schrift und **Schriftgestaltung**. HTML war ursprünglich aus typographischer Sicht ein Trauerspiel, da es kaum Möglichkeiten gab, auf das Aussehen des Textes Einfluss zu nehmen. Mit CSS hat sich nicht nur einiges
bei den gebräuchlichen Befehlen aus den letzten beiden Abschnitten geändert, sondern auch im Speziellen sind neue Möglichkeiten hinzugekommen:

- `font-variant` wird hauptsächlich dazu eingesetzt, Text mit dem Wert `small-caps` in Kapitälchen umzuwandeln. Kapitälchen heißt, dass alle Kleinbuchstaben als verkleinerte Großbuchstaben dargestellt werden. Je nach Schriftart
  kann es dafür sogar einen eigenen Schriftschnitt geben. Das Gegenstück zu `small-caps` ist der Standardwert `normal`.

```
<p style="font-variant: small-caps;">
Absatz in Kapitälchen.</p>
```

- `text-transform` wandelt einen Text in Kleinbuchstaben (`lowercase`), Großbuchstaben (`uppercase`) oder große Anfangsbuchstaben und dann Kleinschreibung (`capitalize`) um.

```
<p style="text-transform: uppercase;">
Text wird automatisch in Großbuchstaben umgewandelt.</p>
```

- `text-decoration` dient dazu, Text auszuzeichnen. Die Möglichkeiten sind dabei recht weit reichend: `underline` unterstreicht, `line-through` streicht durch und wird z.B. dazu verwendet, ältere, nicht mehr gültige
  Gedanken in Texten festzuhalten. `overline` überstreicht, fügt also eine Linie über dem Text ein. none entfernt jede Art von Strich. Mit dem Wert `blink` können Sie außerdem einen Text zum Blinken bringen. Das klappt allerdings

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

nicht im Internet Explorer.

```
<p style="text-decoration: underline;">Besonders wichtige Gedanken <span
style="text-decoration: line-through;">werden unterstrichen</span> <span
style="text-decoration: none; font-weight: bold;">fett hervorgehoben</span>,
nicht mehr gültige durchgestrichen.</p>
```

Info
`text-decoration` kommt oftmals bei Links zum Einsatz. Mit `text-decoration: none;` lässt sich nämlich die Unterstreichung des Links entfernen. Allerdings sollten Sie dann darauf achten, dass Links nach wie vor gut genug zu erkennen
sind. **Unterstreichungen** im normalen Text sollten Sie weitgehend vermeiden, da sie zu sehr an Links erinnern.

- `line-height` bestimmt die Zeilenhöhe. Die Angabe erfolgt oft in Prozent oder Punkt.

```
<p style="line-height: 200%;">
Textabsatz mit besonders hohem Zeilenabstand.</p>
```

Nur ein Bereich, nämlich hochgestellte und tiefgestellte Buchstaben, ist eher eine Sache von HTML als von CSS. HTML sieht hier nämlich die Tags `<sup>` (hochgestellt) und `<sub>` (tiefgestellt) vor, für die es keine direkte
CSS-Entsprechung gibt.

```
<p>x<sup>2</sup> + a<sub>8</sub> = c<sup>3 </sup></p>
```

Komplexere Formeln lassen sich in HTML nicht mehr ausdrücken. Diese stellt man entweder als Grafiken ins Netz oder verwendet eine Technologie wie MathML, eine auf XML basierende Sprache für mathematische Formeln. Letztere benötigt allerdings ein eigenes
Browser-Plug-In.
Info Der Befehl `font` fasst `font-style`, `font-variant`, `font-weight`, `font-size/line-height` und `font-family` zusammen. Unbedingt vorhanden sein müssen `font-size`und `font-family`, und zwar immer und in genau in ihrer Position am Ende der Aufzählung, die anderen drei sind optional und können in ihrer Reihenfolge vertauscht werden. Separat festlegen müssen Sie `color`.

Überschriften

HTML unterscheidet sechs **Überschriftebenen**. In der Praxis werden allerdings selten mehr als vier Ebenen benötigt und die Browser bieten auch meist nur für die ersten vier Ebenen sinnvolle Default-Formatierungen an.

Die Überschriftenebene ist an der Ziffer im einleitenden und abschließenden Überschriften-Tag erkennbar:

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- Beispiel für Überschriften in HTML -->
<html>
<head>
<title>Überschrift</title>
</head>
<body>
<h1>Überschrift 1. Ordnung</h1>
<h2>Überschrift 2. Ordnung</h2>
<h3>Überschrift 3. Ordnung</h3>
<h4>Überschrift 4. Ordnung</h4>
<h5>Überschrift 5. Ordnung</h5>
<h6>Überschrift 6. Ordnung</h6>
</body>
</html>
```

Alle Überschriftenelemente beginnen mit h (heading zu Deutsch: Überschrift). Die Ziffer dahinter bezeichnet die Überschriftenebene. Achten Sie darauf, dass Sie beim Abschluss-Tag einer Überschrift stets die gleiche Ebenenziffer
angeben wie beim Start-Tag. Webbrowser verwenden für die Darstellung von Überschriften Default-Werte. Zweifellos entspricht diese kaum je den Wünschen, die Sie an die Darstellung haben: Ausrichtung, Schriftart, Schriftgröße, Farbe, Abstände, eventuell
auch Schmuckrahmen, eigene Hintergrundfarbe usw. All das können Sie später mithilfe von CSS bewerkstelligen. An dieser Stelle wollen wir uns mit der Default-Darstellung begnügen. HTML kennt für Text noch weitere **Blockelemente** außer dem
normalen Absatz: die Überschriften. Sie reichen von <h1> bis <h6>. <h1> ist dabei die Überschrift der obersten Ebene. Überschriften werden vom Browser selbstständig mit anderer Schriftgröße und Hervorhebung formatiert.
Info In der Praxis macht es allerdings mehr Sinn, die Formatierung von Überschriften per CSS selbst zu übernehmen und die Überschriften-Tags nur als logische Auszeichnung zu verwenden. Logisch heißt, dass Sie für eine oberste Überschriftenebene
auch nur <h1> verwenden und darunter direkt <h2> und nicht wegen kleinerer Schriftgröße <h3> oder <h4>. Ein strukturierter Text zeigt, wie die Überschriften funktionieren. Statt
realen Inhalten finden Sie hier kurze Platzhalter:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- ueberschriften.html -->
<html>
<head>
<title>Überschriften</title>
</head>
<body>
<h1>Thema des Kapitels</h1>
<p>Vortext.</p>
<h2>Inhalt 1</h2>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<p>Beschreibung.</p>
<h3>Unterpunkt 1.1</h3>
<p>Erläuterung.</p>
<h2>Inhalt 2</h2>
<p>Beschreibung.</p>
</body>
</html>
```

Mit Hilfe von CSS können Sie dies nun hübscher formatieren. Sehr praktisch sind dazu die Tag-Selektoren (CSS-Einführung). So geht es Schritt für Schritt:

- 1. Erstellen Sie einen `<style>`-Block im Kopf der HTML-Seite.
- 2. Zuerst legen Sie eine Schriftart für die gesamte Seite fest. Dies geschieht am besten für das `<body>`-Tag, da die Schriftart an alle untergeordneten Elemente vererbt wird.

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

Info Der Netscape Navigator 4.x vererbt nicht korrekt. Wenn Sie eine **Schriftart** auch in Tabellen verwenden möchten, müssen Sie die Tags für Tabellenzellen mit einbeziehen:

```
body, td, th {
font-family: Arial, Geneva, Helvetica, sans-serif;
}
```

- 3. Anschließend formatieren Sie die normalen Absätze. Diese Reihenfolge ist zwar nicht nötig, eine logische Anordnung hilft aber, den Code auch im Nachhinein noch verstehen und warten zu können.

```
p {
font-size: 10pt;
}
```

- 4. Als Nächstes ist Überschrift 1 dran. Sie wird vom Browser automatisch sehr groß und fett formatiert. Die fette Formatierung ist in Ordnung, aber die Größe schrauben wir herunter.

```
h1 {
font-size: 14pt;
font-weight: bold;
}
```

Eigentlich müssten Sie `font-weight` hier nicht angeben. Da einige Nutzer aber auch eigene Stilangaben

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

treffen, wenn Ihre Stilangaben fehlen, haben wir uns dies angewöhnt.

- 5. Überschrift 2 formatieren Sie ähnlich wie Überschrift 1, nur ein wenig kleiner:

```
h2 {
font-size: 12pt;
font-weight: bold;
}
```

- 6. Bei Überschrift 3 entfernen Sie die fette Formatierung und verwenden stattdessen kursiv:

```
h3 {
font-size: 10pt;
font-weight: normal;
font-style: italic;
}
```

Achten Sie immer auf die Standardwerte. Überschrift 3 ist standardmäßig fett formatiert. Das heißt, Sie müssen in CSS `font-weight` auf normal setzen, um den Fettdruck auszuschalten.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<!-- ueberschriften_format.html -->
<html>
<head>
<title>Überschriften</title>
<style type="text/css"><!--
body {
font-family: Arial, Geneva, Helvetica, sans-serif;
}
p {
font-size: 10pt;
}
h1 {
font-size: 14pt;
font-weight: bold;
}
h2 {
font-size: 12pt;
font-weight: bold;
}
h3 {
font-size: 10pt;
font-weight: normal;
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
font-style: italic;
}
--></style>
</head>
<body>
<h1>Thema des Kapitels</h1>
<p>Vortext.</p>
<h2>Inhalt 1</h2>
<p>Beschreibung.</p>
<h3>Unterpunkt 1.1</h3>
<p>Erläuterung.</p>
<h2>Inhalt 2</h2>
<p>Beschreibung.</p>
</body>
</html>
```

Zeilenumbruch erzwingen

In dem Artikel Absätze konnten Sie eine Besonderheit feststellen: Der Text, der im zweiten Absatz im HTML-Dokument über drei Zeilen verteilt war, wurde im Browser in einer Zeile dargestellt. Das liegt daran, dass HTML **Zeilenumbrüche** oder mehrere Leerzeichen hintereinander in einem p-Element als ein einzelnes Leerzeichen darstellt.

Um trotzdem einen Zeilenumbruch erzwingen zu können, wird das `br`-Element (br = line break, = Zeilenumbruch) verwendet. Dieses Element, wie auch viele weitere, die Sie kennenlernen werden, besitzt eine Besonderheit: Es gibt kein
Ende-Tag zum `br`-Element, da es einen Text nicht direkt formatiert und keine weiteren Elemente enthalten darf – es gibt also keinen Gültigkeitsbereich.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<!-- Beispiel für Zeilenumbrüche im p-Element -->
<html>
<head>
<title>Listing 2.2</title>
</head>
<body>
<p>Dieser Absatz enthält sehr viel Text
<br>,
der im      HTML-Dokument aber ein vollkommen anderes Erscheinungsbild
<br>
besitzt,         als die        Ausgabe lässt.
</p>
</body>
</html>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Info

 (br = break = Umbruch) fugt an der gewünschten Stelle einen Zeilenumbruch ein. Dabei ist es egal, ob das allein stehende Tag am Ende der vorherigen Zeile steht (wie im Beispiel) oder in einer eigenen Zeile, oder am Anfang
der folgenden Zeile. Wenn Sie XHTML-konform arbeiten, müssen Sie das br-Element als Inhaltsleer kennzeichnen. Dazu notieren Sie das allein stehende Tag in der Form <br />.

Automatischer Zeilenumbruch

Sie können einen **Textbereich** bestimmen, in dem kein automatischer Zeilenumbruch erfolgt. Alles, was innerhalb dieses Bereichs steht, wird in einer langen Zeile atgezeigt. Der Anwender kann dann mit der horizontalen Scroll-Leiste die überlangen
Textzeile anzeiget Diese Möglichkeit gehört jedoch nicht zum offiziellen HTML-Sprachstandard. Sie sollten sie daher vermeiden.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01/EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>ARD und ZDF</h1>
<nobr>Die vom ZDF sagen die von ARD senden ständig Wiederholungen, und die vom
ARD sagen die von ZDF senden ständig Wiederholungen, und so wiederholen sich ARD
und ZDF ständig ohne überhaupt etwas zu senden.</nobr>
</body>
</html>
```

Info Erläuterung: <nobr> bewirkt, dass der auf das Tag folgende Text nicht umbrochen wird (nobr = no break = kein umbruch). Am Ende des Textabschnitts, der nicht umbrochen werden soll, notieren Sie das abschließende
Tag </nobr>. Wenn Sie Textzeilen unabhängig vom Anzeigefenster des Anwenders genau kontrollieren und nach HTML-Standard arbeiten wollen, können Sie präformatierten Text einsetzen.

Geschützte Leerzeichen

Wie Sie verhindern können, dass bei einem **Leerzeichen** ein automatischer Zeilenumbruch erfolgen darf, zeigt Ihnen folgendes Beispiel.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
</head>
<body>
<h1>HTML 2.0 und HTML 4.0</h1>
<p>Es gibt vieles, worin sich HTML 2.0 und HTML 4.0 unterscheiden.</p>
</body>
</html>
```

Info Die Zeichenfolge ` ` erzeugt ein geschütztes Leerzeichen (nbsp = nonbreaking space = nicht umbrechbares Leerzeichen). Es wird ein normales Leerzeichen atgezeigt, doch an dieser Stelle kann kein Zeilenumbruch
erfolgen. Notieren Sie die Zeichenfolge inklusive kaufrnännisches Und am Beginn und Strichpunkt am Ende. Die gleiche Wirkung erzielen Sie durch Notieren der Zeichenfolge `&#11160;`. Zu dieser Art von Zeichennotation siehe auch benannte
Zeichen für den Zeichensatz ISO 8859-1 im Anhang. Durch Notieren mehrerer solcher benannter Zeichen hintereinander können Sie auch mehrere Leerzeichen in Folge erzwingen.

Zeilenumbruch erlauben

Webbrowser umbrechen Text normalerweise nur bei Leerzeichen, weil durch Leerzeichen Wörter voneinander abgegrenzt werden. Sie können explizit weitere Stellen markieren, an denen der Browser den Text umbrechen darf. Dies gilt für alle Absatzarten in **HTML**.
Diese Möglichkeit gehört jedoch nicht zum offiziellen HTML-Sprachstandard. Sie sollten sie daher vermeiden.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01/EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Langes Wort</h1>
<p>Donaudampfschiffahrt-<wbr>Kapitänsmütze Donaudampfschiffahrt-
<wbr>Kapitänsmütze ... </p>
</body>
</html>
```

Info Mit `<wbr>` markieren Sie eine Stelle, an der getrennt werden darf, falls diese Stelle bei der Bildschirmanzeige am Ende der Zeile steht (wbr = word break = Umbruch innerhalb eines Wortes). Sinnvoll ist dies bei
langen Wörtern oder aus Bindestrichen bestehenden Ausdrücken. Innerhalb von Abschnitten mit verhindertem Zeilenumbruch bewirkt `<wbr>`, dass an der betreffenden Stelle trotzdem ein Umbruch erfolgen darf.

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Zitate definieren

Sie können **Zitate** von Fremdautoren in einem eigenen, anders formatierten (zumeist eingerückten) Absatz hervorheben. Es handelt sich dabei jedoch um eine logische, inhaltliche Auszeichnung. Wie diese Absätze genau formatiert werden, bestimmt
letztlich der Webbrowser. Die Vorgabe ist jedoch, solche Absätze auffällig und vom übrigen Text unterscheidbar anzuzeigen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Franz Kafka</h1>
<p>Über die Krähen und den Himmel schreibt Franz Kafka:</p>
<blockqoute>
Die Krähen behaupten, eine einzige Krähe könne den Himmel zerstören; das ist
zweifellos, beweist aber nichts gegen den Himmel, denn Himmel bedeutet eben:
Unmöglichkeit von Krähen.
</blockqoute>
</body>
</html>
```

Info Typischerweise stellen Browser ein Zitat mit eingerückten Rändern dar.

`<blockqoute>` leitet einen eigenen Absatz für Zitate ein (blockqoute = geblocktes Zitat). `</blockqoute>` beendet den Absatz. Das `<blockqoute>`-Element wird gerne verwendet, um Einrückungen zu realisieren. Da es sich jedoch um eine logische **Textauszeichnung** handelt, die keine bestimmte Art
der Formatierung vorschreibt, ist der Einrück-Effekt nicht garantiert.

Zitate mit URI-Quellenangabe

Wenn ein Zitat im Internet verfügbar ist, können Sie den URI, also die Internet-Adresse der Quelle angeben. Ein Beispiel:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
```

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

```
<h1>Die Energie des Verstehens</h1>
<p>Geben Sie nicht auf, denn Sie haben sie:</p>
<blockqoute cite="http://www.html-info.eu/">
Die Energie des Verstehens
</blockqoute>
</body>
</html>
```

Info Mit dem Attribut `cite=` im einleitenden `<blockquote>`-Tag können Sie den URI der zitierten Quelle angeben (cite = Zitat). Dieses Attribut dient lediglich der Information über die Quelle des Zitats und die Spezifikation gibt auch keine Empfehlung zur Visualisierung (in diesem Beispiel werden Sie daher vermutlich keinen Effekt sehen).

Adresse definieren

Sie können Internet-Adressen von Personen oder Dateien in einem eigenen, anders formatierten (zumeist kursiv dargestellten, eingerückten) Absatz hervorheben. Auch dies ist eine logische Textauszeichnung, fur deren tatsächliche Formatierung bei der Ausgabe
es keine festen Vorschriften gibt.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Wenn Sie es genau wissen wollen</h1>
<p>Das HTML-Element für Adressen wird kaum verwendet. Wenn Sie es genau wissen
wollen, wenden Sie sich an:</p>
<adress>Dave Raggett, dsr@w3.org</adress>
<p>Dave Raggett gehört dem W3-Konsortium an und gibt viele Publikationen rund um
HTML auf dem W3-Server heraus.</p>
</body>
</html>
```

Info
`<address>` leitet einen eigenen Absatz für Internet-Adressen ein. `</address>` beendet den Absatz. Das `<address>`-Element darf keine anderen blockerzeugenden **Elemente** wie z.B. überschriften, Textabsätze, Listen, Zitate oder Adressen enthalten. Das `<address>`-Element ist auch in Verbindung mit E-Mail-Verweisen sinnvoll. Adressen formatieren Browser üblicherweise als kursiven Absatz.

Zitate und Adressen formatieren mit CSS

Wie Zitate und Adressen genau dargestellt werden, darauf haben Sie mit HTML keinen Einfluss. Die

Löwenstark Digital Group GmbH
Petzvalstr. 38
38104 Braunschweig
Tel.: 0531 21360 5500
Fax: 0531 21360 5795
kontakt@loewenstark.com

Browser benutzen Default-Formatierungen. Mit Stylesheets können Sie solche Elemente jedoch nach Wunsch formatieren. Bei Verwendung von Stylesheets lesen Sie
zunächst, wie man CSS-Formate definieren kann. Anschließend sind Sie in der Lage, CSS-Eigenschaften anzuwenden. Maßgeblich sind im hier beschriebenen Zusammenhang z.B. folgende CSS-Eigenschaften: Schriftformatierung, Außenrand und Abstand, Innenabstand,
Rahmen, Hintergrundfarben und -bilder.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Text des Titels</title>
</head>
<body>
<h1>Zitat und Adresse</h1>
<blockqoute style="padding:10px; border:thin solid blue;">
Klarheit ist wenn man so will nur die anschaulichere Variante der Wahrheit.
</blockqoute>
<p><adress style="color:blue;">
Brigitte Beispiel, beispiel@example.org
</adress></p>
</body>
</html>
```

Info Mit CSS lassen sich Zutate und Adressen vielfältig formatieren. Das `blockquote`-Element im Beispiel erhält einen dünnen durchgezogenen blauen Rahmen und der Text darin einen Innenabstand von 10 Pixeln zum Rahmen. Das `address`-Element
erhält eine blaue Schriftfarbe.