

100%
Markt+Technik

TEIL
1
2. Auflage

little **boxes**

Webseiten gestalten mit HTML & CSS
Grundlagen

PETER MÜLLER



Mit der beiliegenden CD können Sie sofort loslegen.
Hier finden Sie alle im Buch gezeigten Beispiele,
nützliche Editoren und andere Tools sowie
Ausschnitte aus Peter Müllers Video-Trainings
»Little Boxes, Teil 1« und »Little Boxes, Teil 2«.

Kapitel 3



Wichtige HTML-Elemente (1)

Worin Sie die wichtigsten HTML-Elemente kennenlernen. Sie teilen die Webseite zunächst in Bereiche ein und beginnen dann mit der Erstellung eines sinnvoll strukturierten HTML-Dokuments.

Die Themen im Überblick:

- Die Seite in Bereiche einteilen: div, Seite 58
- Überschriften: h1 bis h6, Seite 61
- Fließtext: Absätze und Hervorhebungen, Seite 62
- Über Block- und Inline-Elemente, Seite 64
- HTML-Elemente verschachteln, Seite 66
- Jeder Browser hat ein eingebautes Stylesheet, Seite 66
- Listen: Aufzählungen und Nummerierungen, Seite 67
- Definitionlisten: dl, dt und dd, Seite 71
- Auf einen Blick, Seite 72

HTML ist im Vergleich zu Englisch, Deutsch oder Swahili eine recht einfache Sprache. Es gibt nur etwa 85 verschiedene Elemente, und in diesem und dem nächsten Kapitel lernen Sie die wichtigsten davon kennen. Korrektes und sinnvolles HTML ist das Fundament einer jeden Webseite.

3.1 Die Seite in Bereiche einteilen: div

`div` ist ein Block-Element, mit dem man andere Elemente wie Absätze, Listen und Grafiken zu einem gemeinsamen Bereich (`div = division = Bereich`) zusammenfassen kann.

Anders ausgedrückt: `div` ist ein Container, eine Kiste, eine Box, in der mehrere andere Kisten aufbewahrt werden können, um sie zum Beispiel später per CSS gemeinsam gestalten zu können.

Vier Bereiche: Kopf, Navigation, Text und Fußzeile

Fast alle Webseiten haben einen Kopfbereich mit einem Logo, einen Bereich für die Navigationselemente, einen Bereich für den Inhalt und eine Fußzeile. Die folgende Minimalausstattung entspricht dem und hat sich als Grundlage für einfache Layouts bewährt:

- *Kopfbereich* mit Logo und Slogan
- *Navigationsbereich* mit Navigationselementen
- *Textbereich* mit Text und Grafiken
- *Fußzeile* mit Platz für Adresse, Copyright, Impressum usw.

Um die Bereiche im Quelltext voneinander unterscheiden zu können, erhalten sie mit dem Attribut `id` einen eindeutigen Namen. In HTML sieht das so aus:

- `<div id="kopfbereich"> </div>`
- `<div id="navibereich"> </div>`
- `<div id="textbereich"> </div>`
- `<div id="fussbereich"> </div>`

Die Einteilung in diese Bereiche ist die Grundlage für einfache Layouts mit ein oder zwei Spalten. Bei komplexeren Layouts ist unter Umständen eine etwas andere Aufteilung sinnvoll, aber zunächst einmal leisten diese vier Bereiche gute Dienste.

Der Schutzumschlag: wrapper

Viele Bücher haben einen Schutzumschlag, einen zusätzlichen Umschlag um den festen Einband, der die Aufgabe hat, diesen zu schüt-

zen (und der sich einfach und billig bedrucken lässt). Einen solchen Schutzumschlag benötigt Ihre Webseite auch: einen Bereich, der alle anderen Bereiche umfasst.

Dieser Bereich hat inhaltlich keine Bedeutung und ist im Grunde eine Dopplung von `body`, erweist sich aber beim Layouten mit CSS als sehr praktisch. Er wird auf vielen Webseiten *wrapper* (»räpper« gesprochen) genannt, was auf Deutsch »Schutzumschlag« heißt. Mit vollem Namen heißt er `<div id="wrapper">`.

Die Bereiche erstellen mit `div`

Die folgenden Bereiche erstellen Sie gleich zu Anfang im `body` der Webseite.

ToDo: Die Webseite mit `div` in Bereiche einteilen

1. Öffnen Sie die Startseite `index.html`.
2. Fügen Sie den fett gedruckten Quelltext hinzu:

```
<body id="startseite">
<div id="wrapper">

<div id="kopfbereich">

</div> <!-- Ende kopfbereich -->

<div id="navibereich">

</div> <!-- Ende navibereich -->

<div id="textbereich">

</div> <!-- Ende textbereich -->

<div id="fussbereich">

</div> <!-- Ende fussbereich -->

</div> <!-- Ende wrapper -->
</body>
```

3. Speichern Sie die Seite.

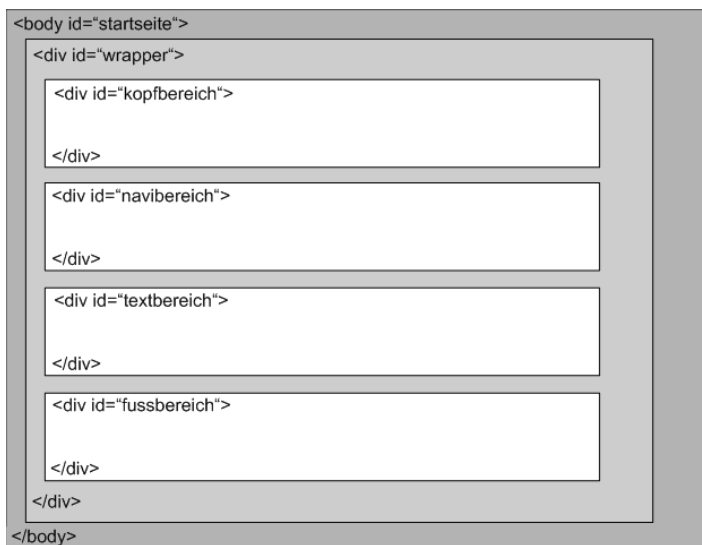
Bei der Darstellung im Browserfenster hat sich nichts geändert, aber bei der Gestaltung per CSS werden die Bereiche und die mit `id` vergebenen Namen wichtig.

Die Kommentare nach jedem `</div>` wirken momentan vielleicht etwas übertrieben, helfen aber, die Übersicht zu behalten, auch wenn Anfangs- und Ende-Tag später etwas weiter auseinanderstehen.

Visuell: schematische Darstellung der `div`-Bereiche

Mit den `div`-Elementen erzeugen Sie rechteckige Kästchen auf der Webseite, in denen andere HTML-Elemente wie Überschriften und Absätze später aufbewahrt werden. Schematisch dargestellt sieht das so aus:

Abbildung 3.1:
Schematische
Darstellung der
mit `div` erzeugten
Bereiche



In dieser Abbildung ist auch deutlich zu sehen, dass der Wrapper im Grunde eine Dopplung von `body` ist.

Tipp Quelltext visualisieren mit **View Source Chart**

Beim Lernen von HTML hilft es vielen Einsteigern, sich den Quelltext wie in der obigen Abbildung mit Kästchen zu visualisieren. Genau das macht das Firefox-Add-on *View Source Chart* direkt im Browser, dessen Installation im Kapitel über nützliche Werkzeuge ab Seite 369 beschrieben wird.

3.2 Überschriften: h1 bis h6

Webseiten werden nach der Anzeige im Browserfenster meist nicht konzentriert Wort für Wort gelesen, sondern auf der Suche nach Relevantem flüchtig überflogen.

Durch den Einsatz von Überschriften, Absätzen und Listen lockern Sie die Webseite optisch auf und bieten dem Auge des Betrachters – bildlich gesprochen – beim Überfliegen des Textes Landeplätze an.

Benutzen Sie Überschriften

Auch wenn Sie im Deutschunterricht das Wort »Gliederung« nicht sonderlich anziehend fanden, sollten Sie auf Ihren Webseiten Überschriften einsetzen. Überschriften geben einer Webseite Struktur und lockern den Text optisch auf.

Überschriften sind im Idealfall so geschrieben, dass der Leser den Inhalt der Webseite durch das Springen von Überschrift zu Überschrift erschließen kann. Ein Hörer kann seinem Screenreader zum Beispiel sagen, dass er nur die Überschriften einer Webseite vorlesen soll.

HTML kennt sechs verschiedene Elemente für Überschriften, von h1 bis h6. Das *h* steht für *heading* (»Überschrift«), die Zahl dahinter für die Gliederungsebene. *h2* bedeutet nicht »die zweite Überschrift im Text«, sondern »eine Überschrift der zweiten Gliederungsebene«. Es kann auf einer Webseite also durchaus mehrere h2-Überschriften geben.

Überschriftenebene nicht aufgrund der Schriftgröße wählen

In den meisten Browsern werden die unterschiedlichen Überschriftenebenen standardmäßig durch verschieden große Schriftarten dargestellt. Widerstehen Sie der Versuchung, eine bestimmte Überschriftenebene aufgrund der Schriftgröße zu wählen:

- Das Aussehen einer Überschrift und damit auch deren Schriftgröße gestalten Sie später mit CSS.
- In HTML sagen Sie momentan nur, *dass* der markierte Text eine Überschrift ist und welche Gliederungsebene sie hat.

Im Klartext: Beginnen Sie Ihre Webseite nicht z. B. mit `<h3>`, nur weil Ihnen `<h1>` zu groß erscheint.

Alle Überschriften sind übrigens sogenannte *Block-Elemente* (siehe Seite 64): Sie werden automatisch so breit, wie es geht, und haben einen integrierten Zeilenumbruch.

ToDo: Eine Überschrift für die Webseite erstellen

1. Öffnen Sie ggf. die Startseite *index.html* im Editor.
2. Setzen Sie den Cursor in den Kopfbereich.
3. Ergänzen Sie den fett hervorgehobenen Quelltext:

```
<div id="kopfbereich">  
<h1>Little Boxes</h1>  
</div>
```

4. Speichern Sie die Seite und betrachten Sie sie in einem Browser.

Dick und fett prangt der Text im Browserfenster. Nicht hübsch, aber Überschrift.

3.3 Fließtext: Absätze und Hervorhebungen

Zwischen den Überschriften steht der Fließtext, und der besteht zu einem großen Teil aus Absätzen, Hervorhebungen und Listen.

Absätze: p wie paragraph

Das wahrscheinlich am häufigsten verwendete HTML-Element ist *p*, kurz für *paragraph*, auf Deutsch »Absatz«. Jeder normale Fließtextabsatz auf einer Webseite wird mit `<p>` eingeleitet und mit `</p>` beendet.

ToDo: Einen normalen Fließtextabsatz erstellen

1. Ergänzen Sie auf der Beispielseite den fett gedruckten Quelltext:

```
<div id="kopfbereich">  
<h1>Little Boxes</h1>  
<p>Webseiten gestalten mit HTML und CSS. Grundlagen.</p>  
</div>
```

2. Speichern Sie die Seite und betrachten Sie sie in einem Browser.

p ist ein Block-Element (siehe Seite 64) und darf *keine* anderen Block-Elemente enthalten, auch kein weiteres p.

Der Abstand zwischen den Überschriften und Absätzen

Die meisten Browser fügen vor und nach Überschriften und Absätzen einen Abstand hinzu. Versuchen Sie momentan gar nicht erst, diesen zu entfernen. In HTML geht das nicht, später mit CSS ist es einfach.

Tipp

Text hervorheben: strong und em

Die Elemente `strong` und `em` sind zur Hervorhebung von Text gedacht. Beide sind *Inline-Elemente* (siehe Seite 64), werden also nur so breit wie der enthaltene Text und erzeugen keinen eigenen Zeilenumbruch:

- *strong* bedeutet *stark hervorheben* und wird in visuellen Browsern meist **fett** gedruckt.
- *em* hingegen steht für *emphasize*, auf Deutsch *betonen*. *em* wird meist *kursiv* dargestellt.

Als Faustregel benutzen Sie `strong`, um den Text bereits *vor* dem Lesen hervorzuheben, und `em`, wenn er erst *während* des Lesens auffallen soll.

ToDo: Text mit strong und em hervorheben

1. Ergänzen Sie den Textbereich auf der Beispielseite wie folgt:

```
<div id="textbereich">
<h2>Startseite</h2>
```

```
<p>Webseiten bestehen aus <strong>rechteckigen Kästchen</strong>,
die über-, neben- und ineinandergestapelt werden. Alles Runde ist
entweder Trick, Grafik oder beides.</p>
```

```
<p>Die folgenden Schritte zeigen den Weg vom Zusammenbauen der
<em>Kisten</em> bis zur fertig gestalteten <em>Webseite</em>:</p>
```

```
</div> <!-- Ende textbereich -->
```

2. Speichern Sie die Seite und betrachten Sie sie im Browser.

3.4 Über Block- und Inline-Elemente

Wie Sie bereits gesehen haben, gibt es verschiedene Arten von HTML-Elementen, die sich unterschiedlich verhalten. Die beiden wichtigsten sind Block- und Inline-Elemente.

Block-Elemente werden so breit, wie es geht

Block-Elemente ähneln Absatzformaten in Word:

- Der Kasten eines Block-Elements wird automatisch so breit, wie es geht.
- Nachfolgende Elemente stehen unterhalb des Kastens in der nächsten Zeile.

Block-Elemente enthalten normalen Text, Inline-Elemente und manchmal auch andere Block-Elemente. Beispiele für Block-Elemente sind die Überschriften `h1` bis `h6`, Fließtextabsätze mit `p` sowie die Listenelemente `ul`, `ol` und `li`, die Sie ab Seite 67 kennenlernen.

Inline-Elemente werden nur so breit wie ihr Inhalt

Inline-Elemente ähneln den Zeichenformaten aus Word:

- Der Kasten eines Inline-Elements wird nur so breit wie sein Inhalt.
- Nachfolgender Text fließt direkt nach dem Element weiter.

Inline-Elemente erzeugen keine neue Zeile und sind den Block-Elementen untergeordnet. Sie dürfen normalen Text und andere Inline-Elemente enthalten, aber keine Block-Elemente. Beispiele für Inline-Elemente sind `strong` und `em` sowie Hyperlinks (`a`) und Grafiken (`img`).

Block- und Inline-Elemente sichtbar machen

In Abbildung 3.2 sind die auf der Beispielseite bisher vorhandenen Block- und Inline-Elemente sichtbar gemacht worden und gekennzeichnet:

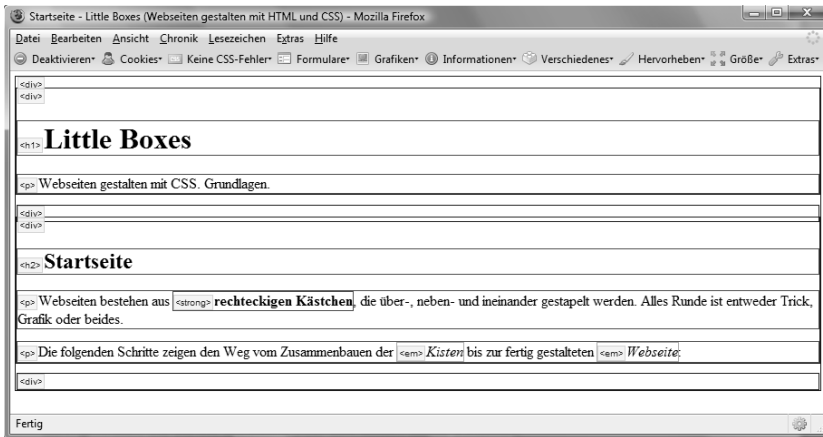


Abbildung 3.2:
Block- und Inline-
Elemente sicht-
bar gemacht

In der Abbildung sehen Sie sehr deutlich, dass Block-Elemente so breit werden, wie es geht, Inline-Elemente nur so breit wie ihr Inhalt.

Falls Sie Abbildung 3.2 bei sich im Browser nachbauen möchten, benötigen Sie zunächst das Firefox-Add-on *Web Developer Toolbar* (siehe Seite 370). Nach der Installation führen Sie dann die folgenden Schritte aus:

1. Aktivieren Sie in der Web Developer Toolbar im Menü HERVORHEBEN ganz unten die Option ELEMENTNAMEN BEIM HERVORHEBEN EINBLENDEN.
2. Um Block-Elemente hervorzuheben, wählen Sie im selben Menü die Option BLOCK-LEVEL-ELEMENTE HERVORHEBEN.
3. Zur Hervorhebung der Inline-Elemente rufen Sie die Option BENUTZERDEFINIERT ELEMENTE HERVORHEBEN ... auf und tragen dort die Namen der gewünschten Elemente (z. B. strong oder em) ein.

Block- oder Inline-Element?

In der HTML-Elemente-Referenz von SelfHTML können Sie sich darüber informieren, ob ein bestimmtes Element *Block* oder *Inline* ist:

■ <http://de.selfhtml.org/html/referenz/elemente.htm>

Tip

3.5 HTML-Elemente verschachteln: zuerst geöffnet, zuletzt geschlossen

Beim Verschachteln von HTML-Elementen müssen Sie eine wichtige Grundregel beachten:

Das zuerst geöffnete Element wird zuletzt geschlossen.

Hier ein Beispiel:

```
<p>Normal, <strong>fett, <em>fett und kursiv</em></strong>. Wieder normal.</p>
```

Die Elemente `p`, `strong` und `em` werden geöffnet und in *umgekehrter* Reihenfolge geschlossen. Die von den Elementen erzeugten Kästchen werden also wie russische Matroschka-Puppen ineinander verschachtelt: `em` steckt in `strong`, das wiederum in der `p`-Kiste sitzt. Grafisch dargestellt sieht das so aus:

Abbildung 3.3:
Eine Kiste in einer
Kiste in einer Kiste



3.6 Jeder Browser hat ein eingebautes Stylesheet

Die Startseite sieht bis jetzt ungefähr so aus:

Abbildung 3.4:
Die Übungsseite
mit Überschriften,
Absätzen und
Hervorhebungen



In Abbildung 3.4 sehen Sie, dass die Elemente im Browserfenster schon Formatierungen haben:

- Überschriften sind fett und groß.
- Zwischen Absätzen sind Leerzeilen.
- `strong` wird fett und `em` kursiv.

Im Quelltext steht nichts dergleichen. Woher kommen also die Formatierungen?

Des Rätsels Lösung ist, dass jeder Browser fest eingebaute Gestaltungsanweisungen hat. Wenn er eine `h1`-Überschrift sieht, denkt er: »Mmmh. `h1` ist eine wichtige Überschrift. Hier steht nicht, wie sie aussehen soll, also mach ich den Text mal groß und fett.«

Solche gespeicherte Gestaltungsanweisungen heißen in vielen Programmen *Formatvorlagen*, oder auf Englisch *style*. Eine Sammlung von Styles heißt *Stylesheet*, und jeder Browser hat ein solches Stylesheet fest eingebaut.

Mit den *Cascading Style Sheets* erstellen Sie eigene Formatvorlagen, die die eingebauten Styles der Browser überschreiben. Alles, was Sie nicht definieren, wird jedoch weiterhin von den Stylesheets der Browser formatiert.

3.7 Listen: Aufzählungen und Nummerierungen

Geordnete und ungeordnete Listen sind ein wichtiges Stilmittel zur Strukturierung von Text auf Webseiten.

Aufzählungen: ungeordnete Listen mit `ul` und `li`

Eine Aufzählung besteht in HTML aus *zwei* Elementen:

- `` und `` kennzeichnen Beginn und Ende der Aufzählung.
- `` und `` markieren jedes Listenelement innerhalb der Aufzählung.

`ul` steht übrigens für »unordered list«, zu Deutsch »ungeordnete Liste«.

ToDo: Eine Aufzählung (ungeordnete Liste) erstellen

1. Erstellen Sie folgende ungeordnete Liste *unterhalb* der auf Seite 63 eingefügten zwei Absätze und *innerhalb* von #textbereich:

```
<ul>  
  <li>So funktioniert HTML</li>  
  <li>Wichtige HTML-Elemente</li>  
  <li>Gestalten mit CSS</li>  
</ul>
```

2. Speichern Sie die Seite und betrachten Sie sie in einem Browser.

`` sagt dem Browser: »Hier beginnt eine ungeordnete Liste.« Zwischen `` und `` stehen dann die einzelnen Listenelemente, bevor die Liste mit `` beendet wird.

Und so sieht die Aufzählung im Browserfenster aus:

Abbildung 3.5:
Ungeordnete Liste im Browserfenster

- So funktioniert HTML
- Wichtige HTML-Elemente
- Gestalten mit CSS

Beachten Sie, dass im HTML nichts über das zu verwendende Aufzählungszeichen steht. Der Quelltext sagt, dass der Text eine Aufzählung *ist*, aber nicht, wie sie *aussieht*. Das wird momentan dem eingebauten Stylesheet des Browsers überlassen.

Tipp Ungeordnete Listen als Grundlage für die Navigation

Ungeordnete Listen bilden oft die Grundlage für die Navigation. Auf Seite 77 erstellen Sie eine HTML-Navigationsliste für die Beispielseiten, die Sie per CSS zuerst *horizontal* (Seite 175) und dann später *vertikal* (ab Seite 302) gestalten.

Nummerierungen: geordnete Listen mit ol und li

Geordnete Listen (»Ordered lists«) sind besser bekannt als *Nummerierungen*. Der Aufbau ist identisch mit ungeordneten Listen, lediglich das `ul` wird durch `ol` ersetzt:

```
<ol>
  <li>So funktioniert HTML</li>
  <li>Wichtige HTML-Elemente</li>
  <li>Gestalten mit CSS</li>
</ol>
```

Zwei Buchstaben geändert, und schon ist eine Aufzählung eine Nummerierung:

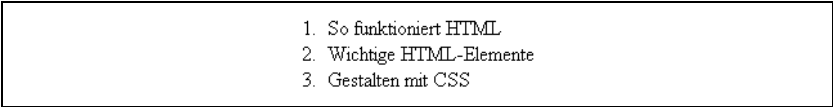
- 
1. So funktioniert HTML
 2. Wichtige HTML-Elemente
 3. Gestalten mit CSS

Abbildung 3.6:
Nummerierung
im Browser-
fenster

Verschachtelte Listen

Listen können ineinander verschachtelt werden. Achten Sie im folgenden ToDo darauf, dass das `` am Anfang des zweiten Listenelements, »Wichtige HTML-Elemente«, erst einige Zeilen tiefer *nach* Beendigung der verschachtelten Liste geschlossen wird.

ToDo: Erstellen Sie eine verschachtelte Liste

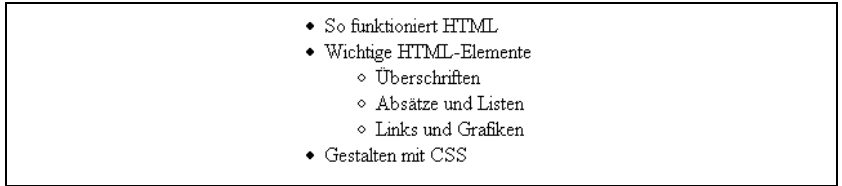
1. Ergänzen Sie die Liste auf der Beispielseite um den fett gedruckten Quelltext:

```
<ul>
  <li>So funktioniert HTML</li>
  <li>Wichtige HTML-Elemente
    <ul>
      <li>Überschriften</li>
      <li>Absätze und Listen</li>
      <li>Links und Grafiken</li>
    </ul>
  </li><!-- beginnt VOR "Wichtige HTML-Elemente"-->
  <li>Gestalten mit CSS</li>
</ul>
```

2. Speichern Sie die Webseite und betrachten Sie sie im Browser.

Die verschachtelte Liste sieht im Browser so aus:

Abbildung 3.7:
Eine verschachtelte ungeordnete Liste im Browserfenster

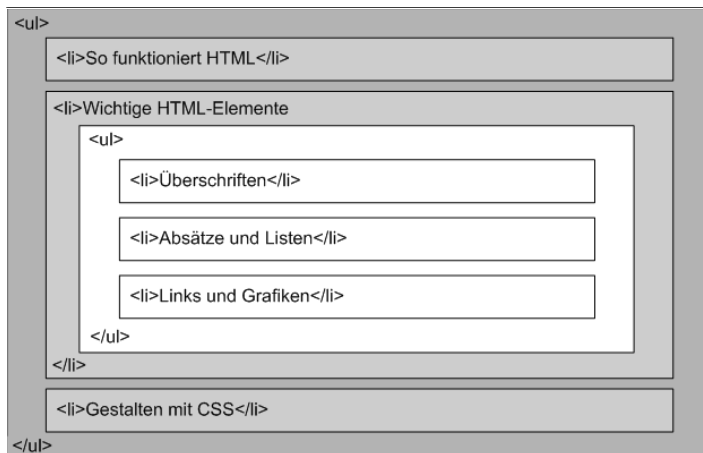


Bei der Verschachtelung von Listen können Sie ungeordnete und geordnete Listen natürlich auch mischen.

Schematische Darstellung der verschachtelten Liste

Zum besseren Verständnis hier ein Schema der Listenkisten aus dem Beispiel:

Abbildung 3.8:
Listenkisten –
schematische
Darstellung der
verschachtelten
Liste



Bei der Gestaltung mit CSS ist die korrekte Verschachtelung der Kisten im HTML-Quelltext wichtig, um später einzelne Listenkisten gezielt gestalten zu können.

Verschachtelte Listen und visuelle Editoren

Falls Sie einen visuellen Editor wie *Dreamweaver* benutzen, sollten Sie nach der Erstellung einer verschachtelten Liste den Quelltext überprüfen, denn viele visuelle Editoren schreiben bei verschachtelten Listen falschen Quelltext.

Im Browser sieht zunächst alles okay aus, aber beim Gestalten mit CSS kann es wichtig werden, dass die Listen korrekt verschachtelt sind.

Tipp

Definitionslisten: dl, dt und dd

Das ursprüngliche Einsatzgebiet für Definitionslisten sind Glossare, Listen von Einträgen mit Begriffen und Erklärungen. Da nicht viele Leute regelmäßig Glossare bauen, sind Definitionslisten ziemlich unbekannt geblieben und werden hier auch nur der Vollständigkeit halber erwähnt.

Anders als geordnete und ungeordnete Listen haben Definitionslisten kein Aufzählungszeichen, bestehen dafür aber aus gleich drei Elementen:

```
<dl>
  <dt>Zu definierender Begriff (definition (list) term)</dt>
  <dd>Beschreibung oder Erklärung (definition (list) definition)</dd>
</dl>
```

Im Browser sieht das so aus:

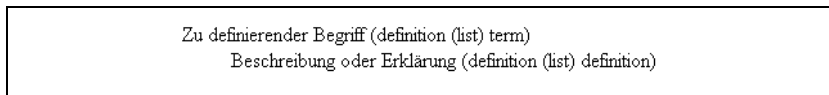


Abbildung 3.9:
Definitionsliste
im Browser-
fenster

Auf den ersten Blick nicht besonders hübsch, so eine Definitionsliste, aber das Aussehen kann mit CSS beliebig verändert werden.

Im CSS-Zeitalter erleben Definitionslisten so etwas wie eine Renaissance, denn `dt` und `dd` bieten in bestimmten Situationen nützliche Grundstrukturen, zum Beispiel bei Produkten und Produktbeschreibungen oder der Beschriftung von Bildern.

Tipp Mehr zu Definitionslisten

In »Little Boxes, Teil 2« wird beschrieben, wie man Definitionslisten zur Beschriftung von Bildern einsetzen kann. Online gibt es auf der australischen Site *MaxDesign* ein sehr gutes (englisches) Tutorial zu Definitionslisten:

■ <http://maxdesign.com.au/presentation/definition/>

3.8 Auf einen Blick

Hier noch einmal die wichtigsten Punkte dieses Kapitels im Überblick:

- Beim Gestalten mit CSS wird die Webseite mit `div` in verschiedene Bereiche eingeteilt.
- Text auf Webseiten wird meist überflogen. Bieten Sie dem Auge des Lesers Landeplätze an: Überschriften, kurze Absätze, Hervorhebungen und Listen.
- Benutzen Sie Überschriften (`h1` bis `h6`) und wählen Sie die Gliederungsebene nicht aufgrund der Schriftgröße.
- Normale Absätze im Fließtext stehen in einem `p`-Element, und Text kann mit `strong` und `em` hervorgehoben werden.
- Die beiden wichtigsten Arten von HTML-Elementen sind *Block* und *Inline*:
 - Block-Elemente erzeugen einen Zeilenumbruch und werden so breit, wie es geht (`div`, `h1` bis `h6`, `p`, `ul`, `ol`, `li` und andere).
 - Inline-Elemente erzeugen keinen Zeilenumbruch und werden nur so breit wie der Inhalt des Elements (`strong`, `em` und andere).
- Bei der Verschachtelung von HTML-Elementen dürfen sich die Elemente nicht überlappen: Das zuerst geöffnete Element wird zuletzt wieder geschlossen.
- Browser haben ein eingebautes Stylesheet, das durch eigene Styles außer Kraft gesetzt wird.

- Listen sind ein wichtiges Stilmittel zur Auflockerung des Textes:
 - Ungeordnete Listen (Aufzählungen) bestehen aus `ul` und `li`.
 - Geordnete Listen (Nummerierungen) bestehen aus `ol` und `li`.
 - Bei verschachtelten Listen wird das `li`-Element erst *nach* der eingeschobenen Liste beendet.
 - Definitionslisten bestehen aus `dl`, `dt` und `dd`.