



Webpublishing mit HTML & CSS

➤ Aktuell zu HTML 5 und CSS 3

Mit einem Vorwort von **Stefan Münz**

3

Einführung in HTML und XHTML



Nachdem wir die Erörterung über das World Wide Web beendet und uns sortiert sowie jede Menge Text gelesen und Konzepte verdaut haben, fragen Sie sich vielleicht, wann Sie denn nun tatsächlich eine Webseite schreiben werden. Schließlich haben Sie deswegen dieses Buch gekauft. Warten Sie nicht mehr länger! In diesem Kapitel erstellen Sie Ihre erste (wenn auch kurze) Webseite und lernen etwas über HTML (die Sprache zum Schreiben von Webseiten). Sie erfahren außerdem das Folgende:

- » Was HTML ist und warum Sie es verwenden müssen
- » Was Sie beim Erstellen von HTML-Seiten tun können und was nicht
- » Was HTML-Tags sind und wie Sie sie verwenden
- » Wie Sie Seiten verfassen, die dem XHTML-Standard entsprechen
- » Wie Sie CSS-Formatierungen verwenden können, um das Aussehen und die Handhabung Ihrer Seiten zu beeinflussen

3.1 Was ist HTML (und was ist es nicht)?

Beachten Sie einen weiteren Punkt, bevor Sie tatsächlich mit dem Schreiben von Webseiten beginnen. Sie sollten wissen, was HTML ist, was es kann und, das ist das Wichtigste, was es nicht kann.

HTML steht für *Hypertext Markup Language*. HTML basiert auf SGML, der *Standard Generalized Markup Language*, einem umfangreicheren, viel komplizierteren Dokumentverarbeitungssystem. Um HTML-Seiten zu schreiben, benötigen Sie nicht viele Kenntnisse über SGML. Es hilft jedoch, wenn Sie eine der Haupteigenschaften von SGML kennen, nämlich dass sie die allgemeine Struktur des Inhalts innerhalb von Dokumenten beschreibt – nicht das tatsächliche Aussehen auf der Seite oder auf dem Bildschirm. Dieses Konzept mag Ihnen ein wenig fremd vorkommen, wenn Sie es gewohnt sind, mit WYSIWYG-Editoren (What You See Is What You Get) wie Dreamweaver von Adobe oder Microsoft FrontPage bzw. Expression Web zu arbeiten. Wir werden uns den Informationen also behutsam nähern.

3.1.1 HTML beschreibt die Struktur einer Seite

HTML ist aufgrund seiner Abstammung von SGML eine Sprache zur Beschreibung der Struktur eines Dokuments, nicht seiner tatsächlichen Darstellung. Die Idee dahinter ist, dass die meisten Dokumente gemeinsame Elemente haben – beispielsweise Titel, Absätze und Listen. Bevor Sie mit dem Schreiben beginnen, können Sie daher die Menge der Elemente in diesem Dokument festlegen, definieren und geeignet benennen (siehe Abbildung 3.1).

Wenn Sie mit Textverarbeitungsprogrammen, die Formatierungen verwenden (z.B. Microsoft Word), oder mit Absatzkatalogen (wie FrameMaker) gearbeitet haben, haben Sie etwas Ähnliches gemacht: Jeder Textabschnitt genügt einer Formatierung aus einer Menge von Formatierungen, die bereits vor dem Beginn der Arbeit definiert sind.

HTML definiert eine Menge verbreiteter Formatierungen für Webseiten: Überschriften, Absätze, Listen und Tabellen. Es definiert auch Zeichenformate wie Fettschrift oder Schreibmaschinenschrift für Codebeispiele. Diese Formatierungen werden innerhalb von HTML-Dokumenten mit Hilfe von *Tags* angezeigt. Jedes Tag hat einen bestimmten Namen und ist vom Inhalt des Dokuments mit einer Notation abgesetzt, die ich etwas später behandle.



Abbildung 3.1: Elemente eines Dokuments

3.1.2 HTML beschreibt nicht das Seitenlayout

Wenn Sie mit einer Textverarbeitung oder einem Seitenlayoutprogramm arbeiten, sind die Formatierungen nicht nur benannte Elemente einer Seite, sondern sie enthalten auch Informationen wie die Schriftgröße und -formatierung, Einzug, Unterstreichungen usw. Wenn Sie also einen Text schreiben, der als Überschrift gedacht ist, können Sie die Formatierung für Überschriften auf ihn anwenden und das Programm formatiert diesen Absatz automatisch mit dem richtigen Format.

So weit geht HTML nicht. Zum größten Teil sagt die HTML-Spezifikation nichts darüber aus, wie die Seite aussieht, wenn sie angeschaut wird. HTML-Tags geben lediglich an, dass ein Element eine Überschrift oder eine Liste ist. Sie sagen nichts dazu, wie diese Überschrift oder Liste formatiert werden soll. Betrachten Sie als Beispiel ein Magazin und einen Layouter, der Ihren Artikel formatiert. Die Aufgabe des Layouters besteht darin, zu entscheiden, wie groß die Überschrift ist und in welcher Schriftart sie stehen sollte. Sie müssen sich nur darum kümmern, Vermerke anzubringen, welcher Abschnitt eine Überschrift sein soll.

HINWEIS

Auch wenn HTML nicht viel darüber sagt, wie eine Seite bei der Ansicht aussieht, können Sie mit CSS fortgeschrittene Formatierungen auf HTML-Tags anwenden. HTML hat sich dahingehend entwickelt, dass Webautoren CSS für Formatierungsanweisungen anstreben. Ich behandle CSS weiter hinten in diesem Kapitel.



Webbrowser übernehmen neben den Netzwerkfunktionen zum Abholen von Seiten aus dem Web auch die Aufgabe der Formatierung von HTML. Wenn Sie eine HTML-Seite in einen Browser wie Firefox, Safari oder Internet Explorer laden, interpretiert oder analysiert (*parst*) der Browser die HTML-Tags und formatiert den Text und die Bilder auf dem Bildschirm. Der Browser enthält Zuordnungen zwischen den Namen der Seitenelemente und den tatsächlichen Formatierungen auf dem Bildschirm, so können z.B. Überschriften in einer größeren Schriftart dargestellt werden als der übrige Text auf der Seite. Er bricht auch den gesamten Text um, so dass er sich der aktuellen Fensterbreite anpasst.

Unterschiedliche Browser auf unterschiedlichen Plattformen können die Formatierungselemente unterschiedlich anzeigen. Einige Browser können abweichende Schriftformatierungen als andere verwenden. Beispielsweise kann ein Browser auf einem Desktoprechner Kursivschrift kursiv anzeigen, während ein Handheld oder Mobiltelefon invertierten oder unterstrichenen Text auf Systemen verwendet, die keine Kursivschrift enthalten. Oder sie geben eine Überschrift in Großbuchstaben anstelle einer größeren Schriftart aus.

Für Sie als Gestalter von Webseiten bedeutet das, dass die Seiten, die Sie mit HTML erstellen, je nach System und Browser unterschiedlich aussehen können. Die tatsächlichen Informationen und Links auf diesen Seiten sind weiterhin vorhanden, aber die Ansicht auf dem Bildschirm verändert sich. Sie können eine Webseite gestalten, die auf dem einen Computersystem perfekt aussieht, aber auf einem anderen völlig anders (und möglicherweise sogar unlesbar).

EXKURS

Die Entwicklung des optischen Erscheinungsbilds für Tags

In der Praxis werden die meisten HTML-Tags auf eine recht einheitliche Art und Weise formatiert, zumindest auf Desktopcomputern. Als die ersten Browser entwickelt wurden, entschied irgendjemand, dass Links unterstrichen und blau sein sollen, besuchte Links violett und hervorgehobener Text in Kursivschrift. Es gab ähnliche Annahmen für alle weiteren Tags. Seitdem befolgt nahezu jeder Browserhersteller diese Vereinbarungen mehr oder weniger genau. Diese Konventionen verwischten die Trennung zwischen Struktur und Darstellung, aber sie existiert immer noch, auch wenn es nicht offensichtlich ist.

3.1.3 Warum ist das so?

Wenn Sie es gewohnt sind, Dokumente zu erstellen, die als gedrucktes Papier enden, mag Ihnen dieses Konzept fast pervers vorkommen. Keine Kontrolle über das Seitenlayout? Die ganze Gestaltung kann davon abhängen, wo die Seite angeschaut wird? Das ist schrecklich! Warum um Himmels Willen soll ein System so funktionieren?

Erinnern Sie sich noch an Kapitel 1, als ich erwähnte, dass eines der tollen Dinge im Web die Plattformunabhängigkeit ist und dass Webseiten auf allen Computersystemen, allen Bildschirmgrößen und allen Anzeigegeräten angeschaut werden können? Wenn das höchste Ziel der Webveröffentlichung darin besteht, dass Ihre Seiten für jeden auf der Welt lesbar sind, können Sie nicht davon ausgehen, dass alle diese Leser dasselbe Computersystem, dieselbe Bildschirmgröße, dieselbe Anzahl an Farben oder dieselben Schriftarten wie Sie haben. Das Web berücksichtigt all diese Unterschiede und bietet allen Browsern und allen Computersystemen eine gemeinsame Basis.

Das Web ist als Gestaltungsmedium keine neue Papiersorte. Es ist ein vollkommen anderes Medium, mit seinen eigenen Einschränkungen und Zielen, die sich von der Arbeit mit Papier unterscheiden. Die wichtigsten Regeln für die Gestaltung von Webseiten sind im Folgenden genannt und ich werde im gesamten Buch auf ihnen herumreiten:

Was Sie tun sollten	Was Sie nicht tun sollten
Gestalten Sie Ihre Seiten so, dass Sie in den meisten Browsern funktionieren.	Gestalten Sie Ihre Seiten nicht auf der Basis des Aussehens auf Ihrem Computersystem und in Ihrem Browser.
Konzentrieren Sie sich auf einen klaren, gut strukturierten Inhalt, der einfach zu lesen und zu verstehen ist.	

Im gesamten Buch zeige ich Ihnen Beispiele von HTML-Code und wie er bei der Anzeige aussieht.

3.1.4 Markup – was heißt das?

HTML ist eine *Markupsprache*. Das Schreiben in einer Markupsprache bedeutet, dass Sie mit dem Text der Seite beginnen und um Wörter und Absätze spezielle Tags hinzufügen. Die Tags zeigen die verschiedenen Teile der Seite an und erzielen unterschiedliche Effekte im Browser. Im nächsten Abschnitt erfahren Sie mehr über Tags und ihre Verwendung.

HTML enthält eine definierte Menge an Tags, die Sie verwenden können. Sie können keine neuen Tags erstellen, um neue Formatierungen oder Funktionen zu erstellen. Und nur um sicherzustellen, dass die Dinge verwirrend sind, unterstützen verschiedene Browser unterschiedliche Arten von Tags. Um das genauer zu verstehen, schauen wir uns kurz die Geschichte von HTML an.

3.1.5 Eine kurze Geschichte der HTML-Tags

HTML 2.0 war der ursprüngliche Standard für HTML (eine schriftliche Spezifikation dafür wird vom W3C entwickelt und gepflegt) und die Menge der Tags, die alle Browser unterstützen müssen. Die meisten Tags dieser ursprünglichen Spezifikation werden weiterhin unterstützt. In den nächsten Abschnitten lernen Sie hauptsächlich die Tags kennen, die zuerst mit HTML 2.0 eingeführt wurden.

Die Spezifikation HTML 3.2 wurde Anfang 1996 entwickelt. Verschiedene Softwarehersteller, darunter IBM, Microsoft, Netscape Communications Corporation, Novell, SoftQuad, Spyglass und Sun Microsystems erarbeiteten diese Spezifikation zusammen mit dem W3C. Einige der grundlegenden Neuerungen in HTML 3.2 waren Funktionen wie Tabellen, Applets und Textfluss um Bilder herum.



HINWEIS

Die in HTML 3.2 eingeführten Verbesserungen werden weiter hinten in diesem Buch behandelt. Über Tabellen lernen Sie mehr in Kapitel 10. Das Kapitel 12 sagt Ihnen, wie Sie Java-Applets verwenden.

Das 1997 eingeführte HTML 4.0 enthielt viele neue Funktionen, die den Entwicklern eine stärkere Kontrolle über das Seitenlayout als HTML 2.0 und 3.2 geben. Wie bei HTML 2.0 und 3.2 ist das W3C für diesen Standard zuständig.

Framesets (ursprünglich eingeführt in Netscape 2.0) und fließende Frames (ursprünglich eingeführt in Internet Explorer 3.0) wurden offizieller Teil der HTML 4.0-Spezifikation. Framesets werden ausführlicher in Kapitel 17 behandelt. Wir sehen auch zusätzliche Verbesserungen bei der Tabellenformatierung und dem Rendering. Die bei weitem wichtigste Änderung in HTML 4.0 war jedoch die verbesserte Integration von Formatierungen.



HINWEIS

Wenn Sie daran interessiert sind, wie die HTML-Entwicklung funktioniert und was genau im W3C los ist, besuchen Sie auf der Website des Consortiums die Seiten für HTML unter <http://www.w3.org/pub/WWW/MarkUp/>.

Es gab eine Zeit, in der Microsoft und Netscape häufig neue Versionen ihrer Browser herausgaben, um im Wettstreit zu sehen, wer die meisten unwiderstehlichen neuen Funktionen zu HTML hinzugefügt hatte, ohne darauf zu warten, dass der Standardisierungsprozess mithält. Heute wachsen die Versionen im Browsermarkt. Microsoft Internet Explorer und Mozilla Firefox sind verbreitet, dazu kommen andere Browser wie Apple Safari, Google Chrome und Opera. Obwohl häufig neue Versionen erscheinen, konzentrieren sich alle darauf, Webstandards einzurichten, anstatt auf eigene Faust nicht standardisierte Funktionen einzuführen. Die wichtigste unter den jüngeren Entwicklungen

war jedoch die Ausbreitung des Webs auf mobile Geräte. Handys und andere Geräte werden leistungsfähiger und verbreiten sich immer mehr, und es wird für Webentwickler wichtiger, diese Plattformen beim Entwurf ihrer Browser in Betracht zu ziehen.

Die zusätzliche Arbeit, die mit den Abweichungen zwischen den verschiedenen Browsern und Plattformen verbunden ist, bereitet den Webentwicklern seit langer Zeit Kopfschmerzen. Es kann verwirrend sein, alle diese Informationen im Auge zu behalten. In diesem Buch erläutere ich bei der Einführung eines Tags alle browserspezifischen Probleme, auf die Sie stoßen können.

3.2 Der aktuelle Standard: XHTML 1.1

XHTML 1.1 ist in *XML (Extensible Markup Language)* geschrieben und stellt den aktuellen Standard dar, an den sich die meisten Webentwickler halten. Das *X* steht für XML, einem weiteren von SGML abgeleiteten Standard. Der Hauptunterschied zu HTML besteht darin, dass XML strengere Regeln für die Dokumentenstruktur enthält. Während HTML 4 nicht geschlossene Elemente verzieht, muss in einem gültigen XML-Dokument jedes Tag geschlossen sein, jedes Attribut einen Wert haben usw. XHTML 1.1 fordert, dass HTML-Dokumente auch aus gültigem XML bestehen.

Technisch gesehen sind XHTML 1.1 und HTML 4.01 *sehr* ähnlich. Die Tags und Attribute sind fast dieselben, aber es müssen einige einfache Regeln befolgt werden, um sicherzustellen, dass ein Dokument die XHTML 1.1-Spezifikation erfüllt. In diesem Buch erläutere ich, wie die verschiedenen HTML-Tags zu behandeln sind, um sicherzustellen, dass Ihre Seiten lesbar sind und weiterhin in allen Browsern gut aussehen.

3.3 Der zukünftige Standard: HTML5

Die HTML-Arbeitsgruppe des W3C ist damit beschäftigt, einen neuen Standard für HTML zu erstellen: HTML5. Das Ziel von HTML5 besteht darin, neue Elemente einzuführen, die den aktuellen Zustand des Webs besser widerspiegeln. Zu diesen Elementen zählen Dinge wie `<header>` und `<footer>` für die Kopf- bzw. Fußzeile der Seite.

HTML5 fordert nicht, dass Webseiten aus gültigem XML bestehen, und schwächt damit einige der Regeln ab, die XHTML 1.1 auferlegte. Das heutige HTML oder XHTML bleibt aber in HTML5 gültig, wenn es vollständig verabschiedet sein wird.

Die meisten Browser unterstützen bereits die neuen Funktionen von HTML5, auch wenn die Spezifizierungen noch nicht abgeschlossen sind. Ich werde Ihnen erläutern, was Sie jetzt schon mit HTML5 machen können und was Sie tun müssen, wenn die Browserunterstützung für HTML5 erweitert wird.

Einer der wichtigsten Unterschiede zwischen HTML5 und älteren HTML-Spezifikationen besteht darin, dass HTML5 in Zusammenarbeit mit den Browserherstellern entworfen wurde. In der Vergangenheit war die Unterstützung für die HTML-Spezifikationen zwischen den Browseranbietern bestenfalls uneinheitlich. HTML5 wird nur Funktionen enthalten, auf deren Unterstützung sich die Browserhersteller festgelegt haben. Wenn über eine bestimmte Funktion keine Einigkeit erzielt werden kann, wird sie nicht in eine Spezifikation aufgenommen. Es besteht die Hoffnung, dass am Ende des HTML5-Prozesses Webentwicklern eine Spezifikation zur Verfügung steht, auf die sie bei ihrer Arbeit zählen können.

Schließlich verschwinden mit HTML5 viele Elemente, die in früheren Standards eingeführt, aber mittlerweile durch CSS-Formatierungen verdrängt wurden. Einige dieser Tags wurden in früheren Standards als veraltet bezeichnet – HTML5 entfernt sie gänzlich.

3.4 Das Aussehen von HTML-Dateien

In HTML geschriebene Seiten sind reine Textdateien (ASCII), was bedeutet, dass sie keine plattform- oder programmspezifischen Informationen enthalten. Jeder Editor, der Text unterstützt, kann ihn lesen. (Das sollte jeder Editor können – mehr zu diesem Thema finden Sie im Abschnitt »HTML-Editoren« ab Seite 97) HTML-Seiten enthalten das Folgende:

- » Den eigentlichen Text der Seite
- » HTML-Tags, die Seitenelemente, Strukturen, Formatierungen und Hypertextlinks auf andere Seiten oder enthaltene Medien anzeigen

Die meisten HTML-Tags sehen ungefähr wie folgt aus:

```
<tagname>Betroffener Text</tagname>
```

Der Tagname selbst (hier `tagname`) ist in spitze Klammern eingeschlossen (`<` `>`). HTML-Tags haben in der Regel ein Anfangs- und ein Endtag und sie umschließen den betroffenen Text. Der Anfangstag schaltet eine Funktion an (wie eine Überschrift, Fettschrift usw.), während der Endtag sie wieder abschaltet. In Endtags steht ein Schrägstrich (`/`) vor dem Tagnamen. Das Anfangstag (beispielsweise `<p>` für einen Absatz, engl. `paragraph`) und das Endtag (beispielsweise `</p>` für einen Absatz) bilden zusammen das, was offiziell *HTML-Element* genannt wird.

ACHTUNG

Beachten Sie den Unterschied zwischen dem im Zusammenhang mit Tags erwähnten Schrägstrich (`/`) und dem umgekehrten Schrägstrich (`\`), der von DOS und Windows in Dateiverweisen auf Festplatten (wie in `C:\Windows` oder anderen Verzeichnispfaden) verwendet wird. Wenn Sie aus Versehen in HTML den umgekehrten anstelle des normalen Schrägstrichs verwenden, erkennt der Browser die Endtags nicht.

Nicht alle HTML-Tags haben ein öffnendes Anfangs- und ein schließendes Endtag. Einige Tags sind nur einseitig und wiederum andere sind Behälter, die zusätzliche Informationen und Text innerhalb der Klammern enthalten. XHTML 1.1 fordert jedoch, dass *alle* Tags geschlossen sind. Sie lernen die richtige Methode zum Öffnen und Schließen von Tags im Laufe dieses Buchs kennen.

Ein weiterer Unterschied zwischen HTML 4.0 und XHTML 1.1 betrifft die Verwendung von Tags und Attributen in Kleinbuchstaben. HTML-Tags unterscheiden nicht zwischen Groß- und Kleinschreibung, d.h., Sie können sie in Großbuchstaben, Kleinbuchstaben oder einer beliebigen Kombination angeben. `<HTML>` ist also dasselbe wie `<html>` und wiederum dasselbe wie `<HtMl>`. Das gilt nicht für XHTML 1.1, wo alle Tag- und Attributnamen in Kleinbuchstaben geschrieben werden müssen. Damit Sie sich an diese Denkweise gewöhnen, enthalten die Beispiele in diesem Buch die Tag- und Attributnamen in fettgedruckten Kleinbuchstaben.

Übungsaufgabe 3.1: Erstellen Sie Ihre erste HTML-Seite



Nun, da Sie gesehen haben, wie HTML aussieht, sind Sie an der Reihe und sollen Ihre erste Webseite erstellen. Starten Sie mit einem einfachen Beispiel, damit Sie ein grundlegendes Gefühl für HTML erhalten.

Um mit dem Schreiben von HTML anzufangen, benötigen Sie weder einen Webserver noch einen Webhost, ja nicht einmal eine Verbindung zum Internet. Alles, was Sie brauchen, ist ein Programm, in dem Sie Ihre HTML-Dateien erstellen können, und mindestens ein Browser, um sie anzuschauen. Sie können schreiben, verlinken und ganze Gruppen von Webseiten testen, ohne überhaupt ein Netzwerk zur Verfügung zu haben. So werden Sie tatsächlich im Großteil dieses Buchs arbeiten. Später beschreibe ich die Veröffentlichung des Ganzen im Web, damit auch andere Personen Ihre Arbeit sehen können.

Als Erstes benötigen Sie einen *Texteditor*. Das ist ein Programm, das Dateien im ASCII-Format speichert. Das ASCII-Format ist nur reiner Text, ohne Zeichenformatierungen oder Sonderzeichen. Für Windows stellen Notepad und Microsoft WordPad gute, einfache Texteditoren dar. (Sie sind standardmäßig installiert.) Es gibt für verschiedene Betriebssysteme auch Shareware-Editoren, z.B. für Windows, Mac OS X und Linux. Wenn Sie in Ihren Browser die Adresse <http://www.download.com> eingeben und als Suchbegriff »Texteditor« eingeben, finden Sie viele zum Download verfügbare Programme. Wenn Sie Windows nutzen, sollten Sie insbesondere HTML-Kit ausprobieren. Das ist ein freier Texteditor, der auf das Bearbeiten von HTML-Dateien spezialisiert ist. Sie können ihn auf <http://www.chami.com/html-kit/> herunterladen. Ebenso sollten Mac-Benutzer sich TextWrangler anschauen, den es auf <http://www.barebones.com> gibt. Wenn Sie lieber in einem Textverarbeitungsprogramm wie Microsoft Word arbeiten, brauchen Sie keine Angst zu haben. Sie können in diesen Programmen Seiten genauso wie in Texteditoren schreiben, es ist nur etwas komplizierter. Wenn Sie den Befehl SPEICHERN oder

SPEICHERN UNTER verwenden, sehen Sie ein Menü mit Formaten, die Sie zum Speichern der Datei verwenden können. Einer der Einträge davon sollte »Nur Text«, »Nur Text mit Zeilenumbrüchen« oder »DOS-Text« lauten. Alle diese Optionen speichern die Datei als reinen ASCII-Text, so als würden Sie einen Texteditor verwenden. Wenn Sie für HTML-Dateien die Auswahl zwischen »DOS-Text« und lediglich »Text« haben, nehmen Sie »DOS-Text« (womit dann allerdings die eventuell im Text vorhandenen Umlaute verloren gehen) und verwenden Sie die Option mit Zeilenumbrüchen, sofern vorhanden.

ACHTUNG

Viele Textverarbeitungsprogramme enthalten einen HTML-Modus oder Verfahren zum Erstellen von HTML- bzw. XML-Code. Diese Funktion kann zu ungewöhnlichen Ergebnissen oder Dateien führen, die sich einfach nicht so verhalten, wie Sie erwarten würden. Es ist keine gute Idee, ein Textverarbeitungsprogramm einzusetzen, um HTML zu erstellen, wenn Sie planen, die Dateien später zu bearbeiten. Wenn Sie an den Beispielen in diesem Buch arbeiten, sollten Sie einen regulären Texteditor verwenden.

Was ist mit dem Überangebot an freien und kommerziellen HTML-Editoren, die versprechen, mit ihrer Hilfe HTML viel einfacher erstellen zu können? Einige von ihnen sind Texteditoren, die bestimmte mit HTML-Code verbundene Aufgaben vereinfachen. Wenn Sie einen dieser Editoren haben, sollten Sie ihn verwenden. Wenn Sie einen ausgefallenen Editor haben, der behauptet, das ganze HTML vor Ihnen zu verbergen, legen Sie ihn für die nächsten Kapitel beiseite und benutzen Sie wenigstens für eine gewisse Zeit einen reinen Texteditor. Der Anhang A enthält viele URLs, unter denen Sie freie und kommerzielle HTML-Editoren für verschiedene Plattformen herunterladen können. Sie finden sie im dortigen Abschnitt »HTML-Editoren und -Konverter«.

Öffnen Sie Ihren Texteditor und geben Sie den nachfolgenden Text ein. Sie brauchen momentan noch nicht zu verstehen, was er bedeutet. In diesem und dem nächsten Kapitel werden Sie viel darüber lernen. Das einfache Beispiel soll nur dazu dienen, dass Sie endlich loslegen können:

```
<!DOCTYPE html><html>
<head>
<title>Meine Beispiel-HTML-Seite</title>
</head>
<body>
<h1>Dies ist eine HTML-Seite</h1>
</body>
</html>
```

HINWEIS

Beachten Sie, dass das Tag `<!DOCTYPE>` im voranstehenden Beispiel nicht wie die übrigen Tags in Kleinbuchstaben erscheint. Es ist eine Ausnahme zu der XHTML-Regel und sollte in Großbuchstaben erscheinen. Der Zweck dieses Tags besteht darin, den Validatoren und Browsern mitzuteilen, gemäß welcher Spezifikation Ihre Seite geschrieben wurde, in diesem Fall HTML5. In Kapitel 19 erwähne ich weitere dieser Dokumenttypen, die Ihnen begegnen werden, und erläutere, wie Sie einen für Ihre Seiten auswählen.



Nachdem Sie Ihre HTML-Datei erstellt haben, speichern Sie sie auf der Festplatte. Denken Sie daran, falls Sie ein Textverarbeitungsprogramm wie Microsoft Word verwenden, dass Sie den Befehl **SPEICHERN UNTER** auswählen und die Datei als »Nur Text« speichern. Wenn Sie einen Namen für sie auswählen, sollten Sie die beiden folgenden Regeln befolgen:

- » Der Dateiname sollte die Erweiterung *.html* haben (*.htm* ist in Ordnung, aber nicht vorzuziehen) – beispielsweise *myfile.html*, *text.html* oder *index.htm*. Die meiste Web-Software fordert diese Endungen, daher gewöhnen Sie es sich am besten gleich jetzt an. (Wenn Sie Windows verwenden, stellen Sie sicher, dass Ihr Computer so eingerichtet ist, dass er Dateitypen anzeigt. Ist das nicht der Fall, erstellen Sie möglicherweise versehentlich Dateinamen wie *myfile.html.txt*, die Ihr Browser nicht als HTML-Dateien erkennt.)
- » Verwenden Sie kurze, einfache Namen ohne Leer- oder Sonderzeichen (Umlaute, ß, Akzente) – lediglich die 26 Buchstaben des Alphabets sowie Zahlen sind geeignet. Stellen Sie sicher, dass Sie nur beschreibende, lesbare Dateinamen wählen. Sie helfen Ihnen dabei, die Übersicht darüber zu behalten, wofür sie verwendet wurden, und können Suchmaschinen das Verarbeiten Ihrer Website erleichtern.

Übungsaufgabe 3.2: Anschauen der Ergebnisse



Nachdem Sie nun eine HTML-Datei vorliegen haben, sollten Sie Ihren Webbrowser starten. Sie müssen nicht mit dem Internet verbunden sein, da Sie keine Seiten auf anderen Websites öffnen werden. Ihr Browser oder die Netzwerkverbindungssoftware beschwert sich möglicherweise über die fehlende Netzwerkverbindung, aber Sie können offline arbeiten.

Nachdem Ihr Browser gestartet ist, suchen Sie nach einem mit **ÖFFNEN**, **DATEI ÖFFNEN** oder vielleicht **SEITE ÖFFNEN** beschrifteten Menüeintrag oder einer entsprechenden Schaltfläche. Damit können Sie die lokale Festplatte durchsuchen. Der Befehl **ÖFFNEN** (oder sein Pendant) öffnet ein lokales Dokument, analysiert es und zeigt es an. Mit Hilfe Ihres Browsers und dieses Befehls können Sie HTML-Dateien auf Ihrem Computer zu Hause in aller Ruhe schreiben und testen. (In den meisten Betriebssystemen können Sie einfach das Symbol der HTML-Datei in ein geöffnetes Browserfenster ziehen, falls Sie das bevorzugen.)

Wenn Ihr Browser nicht eine ähnliche Ansicht wie in Abbildung 3.2 zeigt (wenn z.B. Teile fehlen oder alles wie eine Überschrift aussieht), gehen Sie zurück in Ihren Texteditor und vergleichen Sie Ihre Datei noch einmal mit dem Beispiel. Vergewissern Sie sich, dass alle Tags auch schließende Tags haben und dass jeder öffnenden spitzen Klammer (<) auch eine schließende (>) entspricht. Sie müssen den Browser dazu nicht schließen, korrigieren Sie nur die Datei und speichern Sie sie erneut unter demselben Namen.



Abbildung 3.2: Die Beispiel-HTML-Datei

Gehen Sie dann zum Browser zurück. Suchen Sie einen Menüeintrag oder eine Schaltfläche namens AKTUALISIEREN oder NEU LADEN und betätigen Sie diese. Der Browser liest die neue Version der Datei, und voilà, Sie können die Datei immer wieder editieren und im Browser prüfen, bis alles richtig ist.

Wenn Sie statt einer Ansicht wie in Abbildung 3.2 den tatsächlichen HTML-Text in Ihrem Browser sehen, sollten Sie überprüfen, ob Ihre Datei die Erweiterung *.html* oder *.htm* hat. Diese Erweiterung sagt dem Browser, dass es sich um eine HTML-Datei handelt. Sie ist wichtig.

Wenn es schief geht – Sie erhalten z.B. einen schwarzen Bildschirm oder einige seltsame Zeichen –, stimmt irgendetwas mit Ihrer ursprünglichen Datei nicht. Wenn Sie sie mit einem Textverarbeitungsprogramm erstellt haben, versuchen Sie, sie mit einem reinen Texteditor zu öffnen. (Auch hier reicht Notepad völlig aus.) Wenn der Texteditor die Datei nicht lesen kann oder das Ergebnis entstellt aussieht, haben Sie die Datei nicht im richtigen Format gespeichert. Gehen Sie in Ihren ursprünglichen Editor zurück und versuchen Sie, die Datei noch einmal nur als Text abzuspeichern. Versuchen Sie dann erneut, sie im Browser anzuschauen, bis alles funktioniert.

3.5 Textformatierung und HTML

Wenn ein Browser eine HTML-Datei analysiert, werden alle Formatierungen, die Sie von Hand hinzugefügt haben, ignoriert – also zusätzliche Leerzeichen, Tabulatoren, Zeilenumbrüche usw. Das Einzige, das die Formatierung in einer HTML-Seite angibt, ist ein HTML-Tag. Wenn Sie Stunden damit verbringen, eine reine Textdatei sorgfältig zu editieren, um hübsch formatierte Absätze und Spalten mit Zahlen zu erhalten, aber keinerlei Tags verwenden, fließt, wenn ein Webbrowser diese Datei lädt, der gesamte Text in einen Absatz. All Ihre Arbeit war umsonst.

HINWEIS

Es gibt zwei Ausnahmen von dieser Regel, ein Tag namens `<pre>` und eine CSS-Eigenschaft. Sie erfahren davon in Kapitel 7.



Dass sämtliche Leerräume (Leerzeichen, Tabulatoren, Zeilenumbrüche) ignoriert werden können, hat den Vorteil, dass Sie die Tags an eine beliebige Stelle setzen können. Die folgenden Beispiele ergeben alle dieselbe Ausgabe. Probieren Sie sie aus!

```
<h1> Wenn Musik die Nahrung der Liebe ist, so spielt fort.</h1>  
<h1>  
Wenn Musik die Nahrung der Liebe ist, so spielt fort.  
</h1>
```

```
<h1>  
Wenn Musik die Nahrung der Liebe ist, so spielt fort. </h1>
```

```
<h1> Wenn Musik die Nahrung der Liebe ist,  
so spielt fort. </h1 >
```

3.6 CSS-Formatierung

Weiter vorne in diesem Kapitel habe ich die CSS-Formatierung als eine Möglichkeit vorgestellt, das Aussehen und die Handhabung von Webseiten zu steuern. Formatierungen beeinflussen die Art und Weise, wie der Browser HTML-Tags wiedergibt (bzw. Elemente, wie sie in Standarddokumenten genannt werden). In diesem Kapitel habe ich z.B. einige Male das Tag `<h1>` verwendet. Viele Browser geben Text, der in ein `<h1>`-Tag eingeschlossen ist, in großer Fettschrift aus und lassen nach der Überschrift etwas Leerraum, bevor sie Weiteres drucken. Mit Hilfe von CSS-Formatierungen können Sie den Browser anweisen, das `<h1>`-Tag anders als normal auszugeben. CSS-Formatierungen bieten eine große Flexibilität, wie Sie das Aussehen eines beliebigen Elementtyps ändern können, und es gibt eine Vielzahl an Möglichkeiten, diese Formatierungen anzuwenden.

Die Vorteile von CSS bestehen darin, dass sie in unterschiedlichen Ausprägungen verwendet werden können. Beispielsweise können Sie alle Formatierungen in einer gesonderten Datei speichern, auf die Sie in Ihrer Webseite verweisen. Wenn Sie dann das Erscheinungsbild Ihrer Website ändern wollen, brauchen Sie nur die CSS-Datei zu editieren und führen damit Änderungen durch, die für alle Seiten gelten, die diese Formatierung nutzen. Wenn Sie es vorziehen, können Sie auch Formatierungen am Anfang einer Seite angeben, so dass sie nur für diese Seite gelten. Sie können mit Hilfe des Attributs `style` auch Formatierungen innerhalb der Tags verwenden.

Ebenso ist es möglich, die Ausprägung der erstellten Formatierungen in Abhängigkeit von ihrer Definition zu steuern. Beispielsweise können Sie Regeln aufstellen, die für alle Tags eines bestimmten Typs gelten, wie alle `<h1>`-Elemente. Oder Sie teilen die Elemente in Klassen ein und definieren Regeln, die nur für die Mitglieder einer Klasse gelten. Klassen sind Kategorien oder Bezeichnungen, die Tags über das Attribut `class` zugewiesen werden. Sie können z.B. eine Klasse namens `headline` erzeugen und dann allen `<h1>`-Elementen in der Klasse `headline` die Farbe Rot zuweisen. Ferner können Sie Regeln schreiben, die für

einzelne Elemente gelten, indem Sie ihnen mit dem Attribut `id` bestimmte Bezeichner zuweisen und Regeln verfassen, die auf diesen Bezeichner anzuwenden sind. Nachfolgend sehen Sie ein Beispiel für ein `<h1>`-Tag, das sowohl eine Klasse als auch eine ID enthält:

```
<h1 class="headline" id="leadstoryheadline">Überschrift des Aufmachers</h1>
```

Im Verlauf dieses Buchs werden Sie feststellen, dass CSS-Formatierungen als Ersatz für viele verbreitete Tags dienen können. Bei der Beschreibung verschiedener Tags werde ich erläutern, wie Sie dieselben Effekte mit CSS-Formatierungen erreichen können. Allgemein bedeutet die Flexibilität der CSS, dass Sie HTML verwenden sollten, um die Struktur von Seiten zu beschreiben, und CSS, um ihr Erscheinungsbild zu definieren. Das anspruchsvollste Kapitel zu diesem Thema ist das Kapitel 13. Es erläutert, wie Sie CSS verwenden, um das gesamte Layout einer Seite oder sogar einer ganzen Website zu verwalten.

3.6.1 Formate in Tags einschließen

Sie haben bereits gesehen, wie HTML-Seiten unter Verwendung von Tags erstellt werden. Ich möchte hier eine kurze Pause einlegen und auch Attribute behandeln. Ein Attribut ist eine zusätzliche Information, die in irgendeiner Weise das Verhalten eines Tags verändert. Sie werden im Anfangstag eines Paares angegeben. Ein Beispiel:

```
<tag attribute="Wert">
```

Einige Attribute können mit fast jedem Tag verwendet werden, andere sind sehr spezifisch. Eins der fast überall möglichen Attribute ist `style`. Wenn Sie es in ein Tag einfügen, können Sie eine oder mehrere Formatierungsregeln angeben. Nachfolgend sehen Sie ein Beispiel mit dem `<h1>`-Tag, das ich weiter oben eingeführt habe:

```
<h1 style="font-family: Verdana, sans-serif;">Überschrift</h1>
```

Das `style`-Attribut des `<h1>`-Tags enthält eine Formatierungsangabe. Alle Formatierungsangaben folgen diesem grundlegenden Muster, in dem die Eigenschaft auf der linken und der zugehörige Wert auf der rechten Seite angegeben sind. Die Regel endet auf ein Semikolon und Sie können in einem `style`-Attribut mehrere durch Kommata getrennte Regeln angeben. Bei nur einer Regel ist das Semikolon optional, ich empfehle aber, es trotzdem zu verwenden. Im obigen Beispiel lauten die Eigenschaft `font-family` und der Wert `Verdana, sans-serif`. Dieses Attribut modifiziert das standardmäßige `<h1>`-Tag, indem es den Schrifttyp auf Verdana setzt, und wenn dieser Font nicht auf dem System installiert ist, verwendet der Browser irgendeine andere serifenlose Schrift. (*Serifen* sind die kleinen Linien am Ende der Buchstaben.)

In Formatdeklarationen können viele, viele Eigenschaften angegeben werden. Wie zuvor erwähnt, ist eine Deklaration in einem `style`-Attribut nur eine von mehreren Möglichkeiten, Formatierungen auf ein Dokument anzuwenden.

3.7 HTML-Editoren

Sie glauben vielleicht, dass all dieses Tag-Zeugs eine regelrechte Qual ist, insbesondere wenn Sie dieses kleine Beispiel beim ersten Mal nicht ganz verstanden haben. (Ärgern Sie sich nicht darüber, auch ich habe es beim ersten Mal nicht verstanden und ich habe es selbst erstellt.) Sie müssen alle diese Tags im Kopf behalten, sie korrekt eintippen und jedes einzelne schließen. Was für ein Theater!

Es gibt viele Freeware- und Shareware-Programme zum Editieren von HTML-Dateien. Die meisten von ihnen sind im Wesentlichen Texteditoren mit zusätzlichen Menüeinträgen oder Schaltflächen, die die geeigneten HTML-Tags in den Text einfügen. HTML-basierte Texteditoren sind insbesondere aus zwei Gründen nett: Sie müssen sich nicht alle Tags merken und auch nicht alle eingeben. Ich habe bereits HTML-Kit erwähnt, aber es gibt jede Menge andere. Viele Universaleditoren enthalten heutzutage auch spezielle Funktionen, die die Arbeit mit HTML-Dateien erleichtern.

Viele Editoren auf dem Markt behaupten, WYSIWYG zu unterstützen. Diese Programme tauschen die leichte Handhabbarkeit (Sie brauchen die Tags, die Sie verwenden, nicht im Kopf zu behalten) gegen die Kontrolle über das Markup ein. Der Einsatz eines Texteditors ist der beste Ansatz, um genau die Ergebnisse zu erzielen, die Sie wünschen.

Wenn Sie das im Hinterkopf haben, kann die Verwendung von WYSIWYG-Editoren eine schnelle Möglichkeit sein, einfache HTML-Seiten zu erstellen, solange Ihnen bewusst ist, dass die erzielten Ergebnisse variieren können. Für die professionelle Webentwicklung und den Einsatz vieler fortgeschrittener Funktionen können WYSIWYG-Editoren jedoch unzureichend sein und Sie müssen sowieso unter die Haube schauen, um mit dem HTML-Code zu spielen. Auch wenn Sie vorhaben, für die Masse Ihrer HTML-Arbeit einen WYSIWYG-Editor zu verwenden, sollten Sie für die nächsten Kapitel Geduld haben und die Beispiele in Texteditoren ausprobieren, damit Sie ein Gefühl dafür bekommen, was HTML ist, bevor Sie sich dazu entscheiden, einen Editor zu verwenden, der die Tags vor Ihnen verbirgt.

WYSIWYG-Editoren funktionieren in der Regel am besten mit Dateien, die Sie selbst erstellt haben. Wenn Sie bereits einige HTML-Dateien haben, die Sie bearbeiten müssen, kann es mehr schaden als nützen, sie in einem WYSIWYG-Editor zu öffnen, insbesondere wenn Sie in einem anderen WYSIWYG-Editor erstellt wurden.

ACHTUNG

Zusätzlich zu HTML- und WYSIWYG-Editoren können Sie Konverter verwenden, die Dateien aus vielen verbreiteten Textverarbeitungsprogrammen einlesen und in HTML umwandeln. Mit einer einfachen Menge an Vorlagen können Sie Ihre Seiten vollständig in Ihrem Lieblingstextverarbeitungsprogramm schreiben und wenn Sie fertig sind, können Sie das Ergebnis umwandeln.

In vielen Fällen können Konverter sinnvoll sein, insbesondere, um vorhandene Dokumente so schnell wie möglich ins Web zu stellen. Sie leiden jedoch unter denselben Problemen wie WYSIWYG-Editoren. Die Ergebnisse können von Browser zu Browser unterschiedlich sein und viele neuere oder fortgeschrittene Funktionen fehlen. Außerdem sind die meisten Konverter recht beschränkt, nicht unbedingt in ihren Funktionen, sondern hauptsächlich durch die Beschränkungen von HTML selbst. Mit keiner noch so originellen Konvertierung sind in HTML Dinge möglich, die es noch nicht gibt. Wenn eine bestimmte Fertigkeit nicht in HTML vorgesehen ist, kann der Konverter nichts zur Lösung dieses Problems beitragen. Tatsächlich könnte er am Ende seltsame Sachen mit Ihren HTML-Dateien anstellen und Ihnen mehr Arbeit bereiten, als wenn Sie alle Formatierungen selbst eingetragen hätten.

Wie zuvor erwähnt, finden Sie in Anhang A viele aktuell verfügbare Editoren für Webseiten. Wenn Sie einen einfachen HTML-Editor haben, können Sie diesen problemlos zunächst für die Beispiele in diesem Buch verwenden. Wenn Sie nur einen Texteditor haben – kein Problem, Sie müssen dann nur etwas mehr tippen.

3.8 Zusammenfassung

In diesem Kapitel lernten Sie einige grundlegende Dinge darüber, was HTML ist und wie Sie ein Textdokument als Webseite definieren. Sie erfuhren etwas über die Geschichte von HTML und die Gründe, warum die HTML-Spezifikation bereits mehrere Male geändert wurde. Sie haben auch gelernt, wie HTML mit Hilfe von CSS-Formatierungen erweitert werden kann. Sie erstellten Ihre erste Webseite mit einigen grundlegenden Tags. Das war gar nicht so schlecht, oder? Sie erfuhren auch einiges über die aktuellen Standardversionen von HTML und XHTML und wie Sie Formate mit CSS anwenden. In dem folgenden Kapitel erweitern Sie dieses Wissen und erfahren mehr darüber, wie Sie Überschriften, Text und Listen zu Ihren Seiten hinzufügen.

3.9 Hausaufgaben

Nachdem Sie jetzt eine Einführung in HTML und eine Kostprobe für das Erstellen einer ersten (einfachen) Webseite erhalten haben, erhalten Sie ein paar Hausaufgaben, die Sie weiter an das heranführen, was Sie lernen werden. Einer Reihe von Fragen und Antworten, die sich auf HTML-Formatierungen beziehen, folgen einige kurze Wiederholungsfragen zu HTML. Die Übungen fordern Sie dazu auf, den Code einer etwas umfangreicheren Seite in Ihrem Browser zu untersuchen.

3.9.1 Fragen und Antworten

F: Wie kann ich in HTML eine beliebige Textformatierung vornehmen?

A: Sie können einige Formatierungen an Zeichenketten vornehmen. CSS hat die meisten Tags zur Textformatierung überflüssig gemacht und fast alle wurden in HTML5 entfernt. Die Browser unterstützen jedoch noch die älteren Formatierungselemente und es verbleiben noch ein paar Formatierungstags. In Kapitel 7 lernen Sie einige Tricks zur Formatierung.

F: Ich nutze Windows. Mein Textverarbeitungsprogramm lässt mich Textdateien nur mit der Erweiterung .txt abspeichern. Wenn ich index.html eingebe, speichert das Programm die Datei als index.html.txt. Was kann ich tun?

A: Sie können die Dateien nach dem Speichern umbenennen, so dass diese die Erweiterung `.html` oder `.htm` aufweisen. Doch das kann für eine große Anzahl an Dateien sehr mühselig sein. Verwenden Sie für Ihre Webseiten besser einen Text- oder HTML-Editor.

3.9.2 Wiederholungsfragen

1. Wofür steht HTML? Was ist mit XHTML?
2. Was ist die Hauptfunktion von HTML?
3. Warum steuert HTML nicht das Layout einer Seite?
4. Was ist die Grundstruktur eines HTML-Tags?

3.9.3 Antworten

1. HTML steht für Hypertext Markup Language, XHTML für Extensible Hypertext Markup Language.
2. HTML ermöglicht es, die Struktur eines Dokuments zu beschreiben, so dass es formatiert werden kann, entweder mit HTML-Tags oder mit CSS.
3. HTML steuert nicht das Layout einer Seite, weil es für eine plattformunabhängige Darstellung entworfen wurde. Es berücksichtigt die Unterschiede vieler Plattformen und bietet allen Browsern und allen Computersystemen eine gemeinsame Basis.
4. Die meisten HTML-Elemente bestehen aus Anfangs- und Endtag, sie umschließen den Text, den sie betreffen. Die Tags sind in spitze Klammern eingeschlossen (`<>`). Das Anfangstag schaltet eine Funktion an, das Endtag, dem ein Schrägstrich (`/`) voransteht, schaltet sie wieder aus.

3.9.4 Übungen

1. Bevor Sie tatsächlich damit beginnen, eine inhaltsreichere HTML-Seite zu entwerfen, hilft es sicherlich, ein Gefühl dafür zu erhalten, wie eine HTML-Seite aussieht. Glücklicherweise gibt es viel Quellenmaterial, das Sie sich anschauen können. Jede Seite, die über die Internetverbindung zu Ihrem Browser gelangt, ist im HTML-Format (oder möglicherweise in XHTML) verfasst. (Sie sehen so gut wie nie den Code im Browser, sondern nur das Endergebnis.)

Die meisten Webbrowser enthalten eine Möglichkeit, sich die HTML-Quelle einer Webseite anzuschauen. Wenn Sie z.B. Internet Explorer verwenden, gehen Sie auf die Webseite, die Sie sich ansehen möchten. Wählen Sie ANSICHT/QUELLCODE aus, um den Quelltext in einem Textfenster zu betrachten. In Firefox wählen Sie ANSICHT/SEITENQUELLTEXT ANZEIGEN.

Gehen Sie auf eine typische Homepage und sehen Sie sich den Quelltext dazu an. Abbildung 3.3 zeigt beispielsweise die Homepage von Craigslist, einem freien Online-Anzeigensuchdienst unter <http://www.craigslist.org/>.



Abbildung 3.3: Die Homepage von Craigslist

Der HTML-Quellcode sieht wie in Abbildung 3.4 aus.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>craigslist: hamburg Kleinanzeigen für Jobs, Wohnungen, Persönliches, zum Verkauf, Dienstleistung
</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="description" content="craigslist bietet lokale Kleinanzeigen und Foren für Jobs, Wohnungen
<meta name="google-site-verification" content="Ie0Do80edE2EurJVj-bxqMdX7zkMjx_FnjtQp_XMFio">
<link type="text/css" rel="stylesheet" media="all" href="http://www.craigslist.org/styles/craigslist.c
</head>
<body class="hp">
  <table cellspacing="0" cellpadding="0" summary="page" id="container">
    <tr>
      <td id="leftbar">
        <div id="logo">
          <a href="http://www.craigslist.org/about/sites">craigslist</a> <sup><a href="h
        </div>
        <ul id="postlks">
          <li><a href="https://post.craigslist.org/ham">In 'Kleinanzeigen' veröffentlich
          <li><a href="https://accounts.craigslist.org">Mein Konto</a>
          <li><a href="http://www.craigslist.org/about/help/de">Hilfe/FAQ/Missbrauch/Rec
        </ul>
        <form id="search" action="/search/" method="GET">
          <div>craigslist durchsuchen</div>
          <input type="hidden" name="areaID" value="140">
          <input type="hidden" name="subareaID" value="">

```

Abbildung 3.4: Beispiel für einen HTML-Quelltext

Versuchen Sie den Quelltext Ihrer Lieblingswebseiten anzuschauen. Sie sollten einige Gemeinsamkeiten darin sehen, wie Webseiten organisiert sind, und ein Gefühl für die Arten von Tags bekommen, die HTML verwendet. Sie können viel über HTML lernen, wenn Sie den Onlinetext mit dem zugehörigen Quelltext vergleichen.

Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: info@pearson.de

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.**

Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

<http://ebooks.pearson.de>