

Was Sie mindestens über HTML und das Web wissen müssen



In diesem Kapitel

- ▶ HTML in Textdateien erstellen
- ▶ Webseiten anbieten und anzeigen
- ▶ Links und URLs verstehen
- ▶ Die grundlegende HTML-Syntax verstehen

Willkommen in der wundervollen Welt von HTML und des Web. Wenn Sie über einige Grundkenntnisse verfügen, ein wenig geübt haben und – natürlich – etwas zu sagen haben, können auch Sie sich Ihren eigenen kleinen Bereich im Cyberspace schaffen oder auf Ihrer bereits geleisteten Arbeit aufbauen.

Dieses Buch gibt Ihnen eine praktische Anleitung, wie Sie Ihre erste Webseite erstellen, wie Sie eine vorhandene Webseite aufpeppen oder wie Sie komplexe Websites entwerfen können, auf denen Text, Multimedia, Skripts und andere Komponenten integriert sind, die den Benutzern fesselnde Internet-Erlebnisse bieten.

Am besten lernen Sie HTML, wenn Sie von Anfang an praktisch damit arbeiten. Deshalb erhalten Sie in diesem Kapitel einen schnellen Überblick über ihre Grundlagen. Sie erfahren, wie HTML hinter den Kulissen von Webseiten funktioniert, und lernen ihre Bausteine kennen. Wenn Sie einen Eindruck davon haben, wie HTML arbeitet, können Sie auch schon anfangen, eigene Webseiten zu erstellen.

Webseiten in ihrem natürlichen Umfeld

Webseiten können verschiedene Arten von Inhalten enthalten: *Text, Grafiken, Formulare, Audio- und Video-Dateien* sowie *Interaktive Spiele*.

Wenn Sie eine Zeit lang im Web surfen, begegnen Ihnen vielfältige Inhalte, die in verschiedenen Formen dargestellt werden. Doch obwohl alle Websites verschieden sind, haben sie eins gemeinsam: die *Hypertext Markup Language* HTML.

Unabhängig von den enthaltenen Informationen werden alle Webseiten mit HTML (oder der eng verwandten XHTML) erstellt. HTML ist der Zement, der die Struktur einer Webseite zusammenhält, und die Inhalte, die Grafiken und die anderen Informationen entsprechen den Ziegelsteinen.



HTML-Dateien, mit denen Webseiten erstellt werden, sind einfach Textdokumente. Deshalb funktioniert das Web so gut. Text ist die universelle Sprache der Computer. Jede Textdatei, die Sie auf einem Windows-Computer erstellen, einschließlich einer HTML-Datei, funktioniert gleichermaßen gut auf jedem anderen Betriebssystem.

Doch Webseiten sind keine *reinen* Textdokumente. Sie enthalten spezielle Textkomponenten, die so genannten *HTML*-Auszeichnungen, mit denen der Inhalt aufbereitet, gegliedert und gestaltet wird. HTML ist eine Sammlung von Anweisungen, die in die Inhalte einer einfachen Textdatei eingebettet werden und die das Aussehen und Verhalten einzelner Textkomponenten festlegen.

Doch was genau ist die Hypertext Markup Language, und wie funktioniert sie? Diese Fragen wollen wir jetzt beantworten!

Hypertext

Mit speziellen, eingebetteten Anweisungen können HTML-Dokumente auf etwas anderes verweisen. Diese Zeiger werden als *Hyperlinks*, *Links* oder *Verknüpfungen* bezeichnet. Hyperlinks sind der Leim, der das World Wide Web zusammenhält. In einem Webbrowser werden Hyperlinks normalerweise blau und unterstrichen dargestellt. Wenn Sie auf einen Hyperlink klicken, bringt Sie der Browser an eine andere Stelle.



Egal, ob Hypertext oder nicht – eine Webseite ist eine Textdatei. Sie können eine Webseite mit jeder Anwendung erstellen und editieren, die reine Textdateien speichern kann (wie beispielsweise der Windows Editor). Wenn Sie anfangen, HTML zu lernen, ist ein Texteditor das beste Werkzeug. Sie starten den Windows Editor und können loslegen. Einige Software-Werkzeuge verfügen über ausgefeilte Funktionen; und es gibt spezielle Anwendungen (siehe Kapitel 20), die Ihnen helfen, Webseiten zu erstellen, aber sie erzeugen dieselben Textdateien wie einfache Texteditoren.

Das World Wide Web wird seinem Namen gerecht. Es ist buchstäblich ein Netz von Webseiten, die auf Webservern auf der ganzen Welt gespeichert und auf Millionen von Arten miteinander verbunden sind. Diese Verbindungen werden mit Hyperlinks hergestellt, die Seiten miteinander verknüpfen. Ohne diese Links wäre das Web kein Netz von verknüpften Seiten, sondern eine Gruppe eigenständiger, isolierter Seiten.

Der Nutzen des Web liegt zu einem großen Teil tatsächlich in seiner Möglichkeit, Seiten und andere Ressourcen (wie Bilder, herunterladbare Dateien und Medienpräsentationen) auf derselben oder auf einer anderen Website miteinander zu verknüpfen. Beispielsweise ist die REGIERUNGonline (www.bundesregierung.de) eine so genannte *Portalseite*, deren einzige Funktion darin besteht, einen Zugang zu anderen Websites und Webseiten zu schaffen. Wenn Sie nicht wissen, welche Behörde für eine bestimmte Frage zuständig ist, oder wenn Sie sich über ein Gesetz informieren wollen, können Sie diese Site (siehe Abbildung 1.1) besuchen.



Abbildung 1.1: REGIERUNGonline ist ein Portal, das Hyperlinks verwendet, um Besuchern bei der Suche nach Informationen über die Bundesregierung zu helfen.

Auszeichnungen

Webbrowser wurden speziell zu dem Zweck entwickelt, HTML-Anweisungen (so genannte *Auszeichnungen*) zu lesen und die Seiten entsprechend anzuzeigen.

Die Auszeichnungen sind in eine Textdatei (den Inhalt) eingebettet, um dem Browser Anweisungen zu geben.

Beispielsweise enthält die Webseite aus Abbildung 1.2 verschiedene Komponenten. Ein Blick auf die zugehörigen HTML-Auszeichnungen im Text zeigt Ihnen, wie diese Seite aufgebaut und formatiert wird.

Diese Seite enthält die folgenden Komponenten: eine Überschrift, die die Informationen auf dieser Seite beschreibt, ein Bild, einen Absatz mit Text über Rotweine und eine Liste verschiedener Rotweinsorten.

Diese Komponenten sind unterschiedlich formatiert.

- ✓ Die Überschrift am oberen Rand der Seite ist größer als der Text in dem Absatz.
- ✓ Die Listenelemente werden durch Aufzählungszeichen (große Punkte) gekennzeichnet.

Aufgrund des HTML-Codes weiß der Browser, wie diese verschiedenen Komponenten auf der Seite angezeigt werden sollen (siehe Listing 1.1).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <title>Weinsorten</title>
  </head>

  <body>
    <h1>
      Rotweinsorten verstehen
    </h1>
    <p>Obwohl Weine im Allgemeinen entweder in Rotweine
      oder Weißweine eingeteilt werden, gibt es tatsächlich
      eine ganze Reihe von Weinsorten mit eigenen spezifischen
      Merkmalen. So umfasst die Kategorie der Rotweine
      eine ansehnliche Sammlung von über 20 Sorten.
      Dazu zählen:
    </p>
    <ul>
      <li>Barbera</li>
      <li>Brunello</li>
      <li>Cabernet Franc</li>
      <li>Cabernet Sauvignon</li>
      <li>Carignan</li>
      <li>Carmenere</li>
      <li>Charbono</li>
      <li>Dolcetto</li>
      <li>Gamay</li>
      <li>Grenache</li>
      <li>Malbrec</li>
      <li>Merlot</li>
      <li>Mourvedre</li>
      <li>Neebiolo</li>
      <li>Petite Sirah</li>
      <li>Pinot Noir</li>
      <li>Sangiovese</li>
      <li>Syrah</li>
      <li>Tempranillo</li>
      <li>Zinfandel</li>
    </ul>
  </body>
</html>
```

Listing 1.1: Beispiel für HTML-Auszeichnungen

Der Text, der in die Kleiner-als- und Größer-als-Zeichen (< >) eingeschlossen ist, ist HTML-Code. Dieser Code wird häufig auch als *Tag* oder *Auszeichnung* (engl. *markup*) bezeichnet.

Beispielsweise kennzeichnet die Auszeichnung `<p>...</p>` den Text über die Rotweinsorten als Absatz, und die Auszeichnung `...` kennzeichnet die Elemente in der Liste der Rotweinsorten. Und das ist auch schon alles. Sie betten die Auszeichnung zusammen mit Ihrem Text in eine Textdatei ein, um dem Browser mitzuteilen, wie Ihre Webseite angezeigt werden soll.



Tags und Inhalte zwischen und innerhalb von Tags werden zusammenfassend als *Elemente* bezeichnet.



Abbildung 1.2: Diese Webseite enthält verschiedene Komponenten und Auszeichnungen.

Browser

Das letzte Stück des Web-Puzzles ist ein Webbrowser. Webbrowser interpretieren Anweisungen, die in (X)HTML geschrieben sind, und verwenden diese Anweisungen, um die Inhalte einer Webseite auf Ihrem Bildschirm anzuzeigen.

Browser im Überfluss

Im Internet gibt es zahlreiche Browser, die unterschiedlich groß sind und über verschiedene Funktionen verfügen. Viele Browser existieren in mehreren Versionen. Zu den gebräuchlichsten Browsern zählen der Microsoft Internet Explorer, Mozilla Firefox, Opera und der Netscape Navigator. Als HTML-Entwickler müssen Sie über die Anforderungen Ihres eigenen Browsers hinausdenken. Jeder Benutzer hat andere Browser-Präferenzen und arbeitet mit anderen Browser-Einstellungen.

Jeder Browser stellt HTML-Code etwas unterschiedlich dar. Außerdem geht jeder Browser mit JavaScript, Multimedia, Style Sheets und anderen HTML-Add-ins etwas anders um. Wenn Sie dann noch die verschiedenen Betriebssysteme (Mac, Windows, Linux usw.) einbeziehen, bekommen Sie einen Eindruck davon, wie kompliziert es werden kann.

Meistens können Sie die Unterschiede vernachlässigen. Aber manche Kombinationen von HTML, Text und Multimedia-Dateien zwingen bestimmte Browser in die Knie.

Wenn Sie HTML-Seiten erstellen, sollten Sie Ihre Seiten mit so vielen Browsern wie möglich testen. Sie sollten auf Ihrem System zu Testzwecken wenigstens drei verschiedene Browser installieren. Wir empfehlen Ihnen die neuesten Versionen des Internet Explorer, von Firefox und von Opera.

Eine ziemlich komplette Liste der verfügbaren Browser finden Sie in der Webbrowser-Kategorie von Yahoo! Besuchen Sie:

http://dir.yahoo.com/Computers_and_Internet/Software/Internet/World_Wide_Web/Browsers/



Wenn Sie HTML-Code schreiben, sollten Sie immer daran denken, dass die Leute Ihre Inhalte mit verschiedenen Browsern betrachten. Denken Sie einfach daran, dass es mehr als eine Art von Browser gibt und von jedem mehrere Versionen im Einsatz sind.



Die meisten Benutzer surfen mit einem grafischen Browser (beispielsweise mit Netscape oder mit dem Internet Explorer) im Web, der Bilder, Textformate, komplexe Layouts und mehr anzeigt. Einige Benutzer bedienen sich jedoch lieber eines textbasierten Browsers (wie Lynx), weil sie sehbehindert sind und die Vorteile einer grafischen Darstellung nicht nutzen können oder weil sie einen schlanken, schnellen Webserver vorziehen, der nur die (Text-)Inhalte darstellt. Selbst wenn Sie mit einem grafischen Browser im Web surfen, sollten Sie immer daran denken, dass einige Besucher Ihre Seite nur in Textform sehen werden. In Kapitel 18 wird näher beschrieben, wie Sie eine Webseite so gestalten können, dass sie unabhängig vom Browser für jeden Benutzer zugänglich ist.

Sie können zu diesem Zweck mehrere Webbrowser verwenden, doch wir raten Ihnen, mit dem Browser anzufangen, mit dem Sie gegenwärtig im Web surfen. Auf diese Weise müssen Sie sich nicht gleichzeitig an einen neuen Browser gewöhnen *und* (X)HTML lernen.

Üblicherweise fordern Webbrowser Webseiten über das Internet von einem Webserver an. Sie können auch HTML-Seiten anzeigen, die Sie auf Ihrem Computer gespeichert haben, bevor Sie sie auf einem Webserver im Internet veröffentlichen. Wenn Sie eigene HTML-Seiten entwickeln, begutachten Sie diese so genannten *lokalen* Webseiten in Ihrem Browser. Diese lokalen Seiten vermitteln Ihnen einen guten Eindruck davon, was die Benutzer sehen werden, wenn die Seite im Internet veröffentlicht wird.



Jeder Browser interpretiert HTML-Code auf seine Weise. Deshalb sieht derselbe HTML-Code in verschiedenen Browsern etwas anders aus. Wenn Sie mit einfachen HTML-Auszeichnungen arbeiten, sind die Unterschiede unbedeutend, aber wenn Sie andere Elemente wie beispielsweise JavaScript oder Multimedia ins Spiel bringen, kann die Wahl der passenden Auszeichnungen problematisch werden.

In Kapitel 2 erfahren Sie, wie Sie mit einem Webbrowser eine lokale Kopie Ihrer ersten Webseite begutachten können.



Einige Leute verwenden reine Textbrowser wie beispielsweise Lynx, weil sie

- ✓ sehbehindert sind und keine grafische Anzeige verwenden können,
- ✓ schlanke, schnelle Browser bevorzugen, die nur Text anzeigen.

Webserver

Ihre HTML-Seiten haben nur dann einen größeren Nutzen, wenn Sie sie dem Rest der Welt zur Verfügung stellen, und Webserver machen dies möglich. Ein *Webserver* ist ein Computer,

- ✓ der mit dem Internet verbunden ist
- ✓ auf dem Webserver-Software installiert ist
- ✓ der die Anforderungen eines Webbrowsers erfüllen kann, der spezielle Seiten abruft

Fast jeder normale Computer kann die Funktionen eines Webserver übernehmen. Doch im Allgemeinen sind Webserver Computer, die speziell für diese Aufgabe zur Verfügung gestellt werden. Sie müssen kein Internet- oder Computer-Experte sein, um Ihre Webseiten öffentlich zugänglich zu machen, aber Sie müssen einen Webserver für Ihre Webseiten finden.

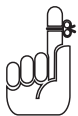
- ✓ Wenn Sie Seiten für die Website eines Unternehmens erstellen, existiert möglicherweise bereits ein Webserver, auf dem Sie die Seiten veröffentlichen können; fragen Sie einfach den zuständigen IT-Abteilungsleiter.
- ✓ Wenn Sie eine neue Site erstellen wollen, müssen Sie einen Host (dt. wörtlich *Gastgeber*) für Ihre Seiten finden.



Webhosting ist heutzutage ein umfangreicher Geschäftszweig und somit ist es leicht, einen preiswerten Host zu finden. In Kapitel 3 beschreiben wir ausführlich, wie Sie Ihre Hosting-Anforderungen ermitteln und einen geeigneten Hosting-Provider finden können.

Anatomie eines URL

Das Web besteht aus Millionen von Ressourcen, die miteinander verknüpfbar sind. Um einen Hyperlink erstellen zu können, müssen Sie die genaue Position einer Ressource kennen, denn ohne die genaue Adresse (den so genannten *Uniform Resource Locator* oder URL) können Sie die entsprechende Webseite nicht direkt von der Adressleiste Ihres Webbrowsers aus aufrufen.



URLs sind das Standardadressierungssystem für Ressourcen im Web. Jede Ressource (sei es eine komplette Website, eine Webseite oder eine einzelne Datei) hat einen eindeutigen URL. URLs funktionieren ähnlich wie Ihre Postanschrift. In Abbildung 1.3 werden die Komponenten eines URL gezeigt.



Abbildung 1.3: Die Komponenten eines URL identifizieren die genaue Position einer einzelnen Datei im Web.

Jede Komponente eines URL spielt eine besondere Rolle, wenn die Position einer Webseite oder eine andere Webressource definiert wird:

- ✓ **Protokoll:** Dieser Teil des Links gibt das Protokoll an, das der Browser verwenden soll, wenn er die Datei anfordert.

Das Protokoll für Webseiten ist `http://` (die vertraute Einleitung der meisten Web-URLs).

- ✓ **Domäne:** Dieser Teil des Links zeigt auf die allgemeine Website (z.B. `www.sun.com`), auf der die Datei gespeichert ist. Eine Domäne kann (wie bei einer privaten Website) einige wenige Dateien oder (wie bei einer Unternehmenssite wie `www.sun.com`) Millionen von Dateien enthalten.
- ✓ **Pfad:** Dieser Teil des Links nennt die Folge der Ordner, durch die Sie navigieren müssen, um zu einer bestimmten Datei zu gelangen.

Um beispielsweise zu einer Datei in dem Unterordner `search` in dem Ordner `training` zu gelangen, müssen Sie den Pfad `/training/search/` angeben.

- ✓ **Dateiname:** Der Name der Datei gibt genau an, auf welche Datei des Verzeichnispfads der Browser zugreifen soll.

Der URL aus Abbildung 1.3 zeigt auf die Domäne von Sun und enthält einen Pfad, der zu der Datei `index.html` führt:

`http://www.sun.com/training/search/index.html`



In Kapitel 6 wird ausführlich und abschließend beschrieben, wie Sie mit HTML und URLs Hyperlinks in Webseiten einfügen können. In Kapitel 3 erfahren Sie, wie Sie einen URL für Ihre eigene Website bekommen, wenn Sie sie auf einem Webserver speichern wollen.



Eine Einführung in Internet-Protokolle

Die Interaktionen zwischen Browsern und Servern werden durch einen Satz von Befehlen zur Kommunikation zwischen Computern gesteuert, der als *Hypertext Transfer Protocol* (HTTP) bezeichnet wird. Dieses Protokoll definiert die Regeln, wie Browser Webseiten anfordern sollen und wie Webserver auf diese Anforderungen reagieren sollen.

HTTP ist nicht das einzige Protokoll, das im Internet verwendet wird. Das *Simple Mail Transfer Protocol* (SMTP) und das *Post Office Protocol* (POP) ermöglichen den Austausch von E-Mails, und das *File Transfer Protocol* (FTP) ermöglicht es, Dateien und Ordner über das Internet hinauf- oder herunterzuladen, zu kopieren und zu löschen. Glücklicherweise erledigen die Webbrowser und Server alle Aufgaben, die mit HTTP verbunden sind, sodass Sie Ihre Seiten nur auf einem Server speichern bzw. eine Webadresse in einen Browser eingeben müssen, um dieses Protokoll nutzen zu können.

(X)HTML-Komponenten

Die folgenden Abschnitte lüften das Geheimnis des X. Dieser Abschnitt zeigt

- ✓ die Unterschiede zwischen HTML und XHTML
- ✓ die Syntax von HTML
- ✓ die Regeln für die Anwendung von HTML
- ✓ Namen für wichtige Komponenten von (X)HTML-Auszeichnungen
- ✓ wie Sie die Möglichkeiten von HTML richtig und am besten anwenden



HTML- und XHTML-Spezifikationen

Die formalen Dokumente, die HTML and XHTML beschreiben, stehen auf der Website des W3C (www.w3.org).

Auszeichnungssprachen werden nicht nur durch ihren Namen, sondern normalerweise zusätzlich durch eine Versionsnummer identifiziert. Die aktuelle Version von HTML ist 4.01 vom 24. Dezember 1999 (www.w3.org/TR/html4). HTML 5 existiert als »Working Draft« vom 22. Januar 2008.

XHTML hat zwei Hauptentwürfe, 1.0 und 1.1, durchlaufen. Die Version 1.1 ist fortgeschrittener als 1.0, aber die meisten Entwickler von Webinhalten und Software-Tools halten sich an die Spezifikation 1.0. Eine Spezifikation für XHTML 2.0 hat den Status »Working Draft« erreicht. Die aktuelle Version ist vom 26. Juli 2006. (Working Drafts sind Entwürfe, deren Inhalt und Struktur noch nicht endgültig feststeht.) Wenn eine W3C-Spezifikation fertig gestellt wird, wird sie als *W3C Recommendation* (W3C-Empfehlung) bezeichnet.

Die Spezifikationen aller drei Versionen von XHTML finden Sie unter:

- ✓ XHTML 2.0 Working Draft (2006-07-26)
<http://www.w3.org/TR/xhtml2/>
- ✓ XHTML 1.1 Module-based XHTML Second Edition 2007-02-16
<http://www.w3.org/TR/xhtml11/>
- ✓ XHTML 1.0 Recommendation (Second Edition; 2002-08-01)
<http://www.w3.org/TR/xhtml1/>

W3C-Spezifikationen zu lesen, erfordert einen gewissen Lernaufwand. Mehrfaches Lesen hilft. Lassen Sie sich durch die formale Sprache und Notation dieser Dokumente nicht abschrecken: Wenn Sie begriffen haben, worum es geht, werden Sie die Präzision und Ausführlichkeit schätzen! Doch ob Sie diese Spezifikationen überhaupt lesen, liegt ganz bei Ihnen!

Die HTML-4.01-Spezifikation ist das Regelwerk der HTML; die XHTML-1.0-Spezifikation ist das Regelwerk der XHTML. Beide Spezifikationen beschreiben genau, welche Elemente Sie verwenden können, welche Attribute zu diesen Elementen gehören und wie Sie diese Elemente kombinieren können, um Seitenstrukturen wie Listen, Formulare, Tabellen und Frames zu erstellen. In diesem Buch verwenden wir die XHTML-1.0-Spezifikation als Basis.

HTML und XHTML: Worin unterscheiden sie sich?

HTML ist die Abkürzung für *Hypertext Markup Language*. Die Sprache wurde in den späten 1980er und frühen 1990er Jahren zur Beschreibung von Webseiten entwickelt und wird heute durch zahlreiche Standardbeschreibungen (so genannte *Spezifikationen*) des World Wide Web Consortium (W3C) definiert. Die letzte HTML-Spezifikation wurde 1999 endgültig verabschiedet.

Wenn Sie vor *HTML* ein *X* setzen, erhalten Sie *XHTML*. Dies ist die Abkürzung für den Namen einer neuen, verbesserten Version von HTML, die auf der *eXtensible Markup Language (XML)* basiert. XML wurde konzipiert, um den Datenaustausch zwischen Computern, Software und dem Internet zu erleichtern.

Die ursprüngliche Formulierung der HTML enthielt einige Unregelmäßigkeiten, die es Software schwierig machte, HTML-Dokumente korrekt zu lesen. Dagegen verwendet XHTML eine außerordentlich regelmäßige und vorhersagbare Syntax, die mit Software viel leichter verarbeitet werden kann. Eines Tages wird XHTML HTML ersetzen, aber HTML bleibt im Einsatz. In diesem Buch werden beide Varianten behandelt; und Sie erfahren, wie Sie das *X* vor Ihre HTML-Dokumente setzen und sie in XHTML-Dokumente umwandeln können.

- ✓ Die meisten HTML- und XHTML-Auszeichnungen sind identisch.
- ✓ In einigen Fällen sehen HTML- und XHTML-Auszeichnungen etwas anders aus.
- ✓ In einigen Fällen müssen HTML- und XHTML-Auszeichnungen unterschiedlich angewendet werden.

In diesem Buch wird gezeigt, wie Sie Code erstellen können, der sowohl unter HTML als auch XHTML funktioniert.

Syntax und Regeln

HTML ist eine sehr geradlinige Sprache, um die Inhalte einer Webseite zu beschreiben. XHTML ist noch anspruchsloser. Ihre Komponenten sind leicht benutzbar – und wenn Sie erst einmal wissen, wie einige HTML-Elemente verwendet werden, folgt der Rest fast automatisch. Die Komponenten von HTML und XHTML lassen sich in drei Hauptkategorien einordnen:

- ✓ **Elemente** identifizieren die verschiedenen Komponenten einer HTML-Seite mit Tags.
- ✓ **Attribute** legen zusätzliche Eigenschaften eines Elements fest.
- ✓ **Entities** repräsentieren Nicht-ASCII-Textzeichen wie beispielsweise das Copyright-Symbol (©) oder Buchstaben mit Akzenten (É).

Alle HTML- oder XHTML-Auszeichnungen, mit denen Sie die Inhalte einer Webseite beschreiben, bestehen aus einer Kombination von Elementen, Attributen und Entities.



In diesem Kapitel werden die grundlegende Form und die Syntax von Elementen, Attributen und Entities abgedeckt. In Teil II und III dieses Buches wird näher beschrieben, wie Sie mit den Elementen und Attributen die folgenden Aufgaben lösen können:

- ✓ Bestimmte Komponenten von Text (z.B. Absätze oder Tabellen) beschreiben
- ✓ Einen speziellen Effekt (beispielsweise eine Änderung der Schriftart) auf einer Seite erzeugen
- ✓ Bilder und Links in eine Seite einfügen

Die (X)HTML-Typen

Die HTML- und XHTML-Spezifikationen verwenden *Document Type Definitions* (DTDs), die in der Standard Generalized Markup Language (SGML) – der Stammutter aller Auszeichnungssprachen – geschrieben sind, um die Einzelheiten von (X)HTML zu definieren.

In den früheren Versionen verwendete HTML Elemente zur Formatierung; im Laufe der Zeit erkannten die Entwickler jedoch,

- ✓ dass für die Formatierung eine eigene Sprache benötigt wurde (die heute als *Cascading Style Sheets*, kurz *CSS*, bezeichnet wird),
- ✓ und dass HTML-Elemente nur die Struktur einer Seite beschreiben sollten.

Auf diese Weise entstanden die drei Varianten von HTML und XHTML. Dies sind die so genannten XHTML-DTDs:

- ✓ **XHTML Transitional:** Verwendet HTML-Elemente, um Schriften und Seitenfarben zu beschreiben. *XHTML Transitional* ist eine Version, die Formatierungselemente von älteren Versionen von XHTML berücksichtigt. Formatierungselemente in XHTML Transitional werden als *ausgemustert* oder *veraltet* (engl. *deprecated*) gekennzeichnet, weil das W3C die HTML-Entwickler dazu veranlassen möchte, auf diese Elemente zu verzichten und stattdessen eine Kombination von *XHTML Strict* und *CSS* zu verwenden. Wir verwenden für die Auszeichnungen in diesem Buch die XHTML Transitional DTD.
- ✓ **XHTML Strict:** Eine Version von (X)HTML, die keine Formatierungselemente enthält. Diese Version soll grundsätzlich mit *CSS* zusammenarbeiten, um die Seiten zu formatieren.

Dieser Ansatz, der XHTML Strict mit *CSS* kombiniert, ist eine ehrgeizige Methode, um Webseiten zu erstellen, aber in der Praxis hat sie ihre Vor- und Nachteile. *CSS* bietet zwar eine größere Kontrolle über die Seitenformatierung, aber *Style Sheets* zu erstellen, die in allen Browsern gut funktionieren, kann schwierig sein. In Kapitel 9 werden *Style Sheets* und die damit verbundenen Probleme ausführlicher beschrieben.

- ✓ **XHTML Frameset:** Eine Version von XHTML, die *Frames* umfasst, das heißt Auszeichnungen, die es Ihnen ermöglichen, in demselben Browserfenster mehr als eine Webseite oder Ressource gleichzeitig anzuzeigen. *Frames* werden auf einigen Websites immer noch verwendet, sind heute aber weniger populär als in den späten 1990er Jahren. Wir raten Ihnen, *Frames* nur zu verwenden, wenn Sie Informationen aus mehreren HTML-Dokumenten gleichzeitig in einem einzigen Browserfenster anzeigen *müssen*.

Alle Webbrowser unterstützen alle Elemente von *XHTML Transitional* (und von XHTML 1.0 Transitional, wenn die Tags korrekt formatiert werden). Deshalb können Sie Elemente dieser Version verwenden oder sich stattdessen auf *XHTML Strict* beschränken. Wenn

Sie Frames verwenden, arbeiten Sie technisch mit *XHTML Frameset*, aber alle Elemente funktionieren auf dieselbe Art.

In diesem Buch behandeln wir alle HTML-Tags aller Versionen unter einer einzigen Kategorie (die wir passenderweise als *XHTML* bezeichnen), weil alle gebräuchlichen Webbrowser diese drei Varianten unterstützen.

Elemente

Elemente bilden den Kern von (X)HTML. Sie dienen dazu, alle Textkomponenten auf einer Seite zu beschreiben, und bestehen aus Tags und den Inhalten zwischen diesen Tags. Es gibt zwei Haupttypen von Elementen:

- ✓ **Elemente, die Inhalte beschreiben, bestehen aus einem Tag-Paar, das Inhalt einschließt.**
- ✓ **Elemente, die etwas in eine Seite einfügen, verwenden ein einziges Tag.**

Tag-Paare

Elemente, die Inhalte beschreiben, verwenden ein *Tag-Paar*, das heißt zwei Tags, die den Anfang bzw. das Ende eines Elements markieren. Sie sehen folgendermaßen aus:

```
<tag>...</tag>
```



Inhalte – wie beispielsweise Absätze, Überschriften, Tabellen und Listen – verwenden immer ein *Tag-Paar*:

- ✓ Ein Start-Tag (`<tag>`), das dem Browser sagt, wo ein Absatz beginnt.
- ✓ Ein End-Tag (`</tag>`), das dem Browser sagt, wo ein Absatz endet.

Der Inhalt selbst steht zwischen dem Start- und dem End-Tag. Beispielsweise verwendet die Seite mit den Rotweinsorten aus Listing 1.1 das Absatz-Element (`<p>`), um einen Absatz zu definieren:

```
<p>Obwohl Weine im Allgemeinen entweder in Rotweine  
oder Weißweine eingeteilt werden, gibt es tatsächlich  
eine ganze Reihe von Weinsorten mit eigenen spezifischen  
Merkmale. So umfasst die Kategorie der Rotweine  
eine ansehnliche Sammlung von über 20 Sorten.  
Dazu zählen:  
</p>
```

Einzelne Tags

Elemente, die etwas in eine Seite einfügen, werden als *leere Elemente* bezeichnet (weil sie keine Inhalte einschließen). Sie verwenden nur ein einziges Tag:

```
<tag />
```



Images (Bilder) und Zeilenumbrüche fügen etwas in die HTML-Datei ein, sodass sie ein einziges Tag verwenden.

Ein Hauptunterschied zwischen XHTML und HTML besteht darin, dass in XHTML alle leeren Elemente mit einem Schrägstrich vor dem schließenden Größer-als-Symbol geschlossen werden müssen, weil XHTML auf XML basiert und XML bei leeren Elementen einen schließenden Schrägstrich verlangt:

```
<tag/>
```

Doch damit diese Art von Auszeichnungen in älteren Browsern lesbar ist, müssen Sie vor dem schließenden Schrägstrich ein Leerzeichen einfügen:

```
<tag />
```

Dieses Leerzeichen ermöglicht es älteren Browsern, den schließenden Schrägstrich zu ignorieren (da sie nichts über XHTML wissen). Neuere Browser, die XHTML verstehen, ignorieren das Leerzeichen und interpretieren das Tag genau wie beabsichtigt, nämlich wie `<tag/>` (gemäß der XML-Regeln).



HTML erfordert bei leeren Elementen keinen Schrägstrich, aber diese Auszeichnung ist ausgemustert. Ein leeres HTML-Element sieht folgendermaßen aus:

```
<tag>
```

In Listing 1.1 wird auch ein `image-Element` (``) verwendet, um ein Bild in die Seite einzufügen:

```

```

Das ``-Element referenziert ein Bild. Wenn der Browser die Seite anzeigt, ersetzt er das ``-Element durch die Datei, auf die das Element verweist. (Das Element verwendet zu diesem Zweck ein Attribut; siehe unten.) Gemäß der weiter oben eingeführten XHTML-Regel wird die HTML-Auszeichnung `` in XHTML als `` dargestellt. (Dies gilt für alle Elemente mit einem einzigen Tag.)



Sie können keine eigenen HTML- oder XHTML-Elemente *definieren*. Es gibt nur eine begrenzte Menge an gültigen Elementen und Elemente, die nicht zu dieser Menge gehören, werden von einem Browser ignoriert. Die Menge der Elemente, die Sie tatsächlich verwenden *dürfen*, wird durch die HTML-4.01- bzw. die XHTML-1.0-Spezifikation definiert.

Verschachtelung

Viele Seitenstrukturen kombinieren verschachtelte Elemente. Wenn Sie Ihre Elemente mit Koffern vergleichen, die sauber ineinandergepackt sind, können Sie nichts falsch machen.

Beispielsweise verwendet eine Aufzählungsliste zwei Arten von Elementen:

- ✓ Das ``-Element legt fest, dass die Liste ungeordnet (mit Aufzählungszeichen) sein soll.
- ✓ Das ``-Element markiert jeden Eintrag der Liste.

Wenn Sie Elemente durch diese Methode einbetten, müssen Sie darauf achten, dass ein inneres Element komplett geschlossen wird, bevor Sie das äußere Element schließen:

```
<ul>
  <li>Barbera</li>
  <li>Brunello</li>
</ul>
```

Attribute

Mit Attributen können Sie modifizieren, wie ein Element Inhalte beschreibt oder auf einer Seite funktioniert. Attribute können als Erweiterungen eines Elements betrachtet werden, die es ermöglichen, ein Element an verschiedene Umstände anzupassen. Beispielsweise verwendet das ``-Element das `src`-Attribut, um den Speicherort des Bildes anzugeben, das Sie an einer bestimmten Stelle in eine Seite einfügen wollen:

```

```

In diesem HTML-Codeausschnitt ist das ``-Element selbst ein allgemeines Flag, das dem Browser sagt, dass Sie ein Bild einfügen wollen; das `src`-Attribut gibt dann die konkreten Details über das Bild an, das Sie einfügen wollen, in diesem Fall `red_grapes.jpg`. Andere Attribute (wie `width`, `height`, `align` und `hspace`) legen fest, wie das Bild angezeigt werden soll, und das `alt`-Attribut gibt eine Textalternative an, falls der Browser das Bild nicht anzeigen kann.



In Kapitel 7 wird das ``-Element mit seinen Attributen ausführlich beschrieben.

Die Attribute eines Elements werden in dem Start-Tag des Elements nach seinem Namen, doch vor dem schließenden Größer-als-Zeichen angegeben:

```
<tag attribut="wert" attribut="wert">
```



Die XML-Syntax schreibt vor, die Werte von Attributen immer in Anführungszeichen einzuschließen. Die Reihenfolge der Attribute und ihrer Werte *innerhalb des Start-Tags* ist beliebig.

Jedes (X)HTML-Element verfügt über eine Reihe von Attributen, die mit ihm verwendet werden können, und Sie dürfen Attribute und Elemente nicht beliebig kombinieren. Einige Attribute können einen beliebigen Text als Wert haben (beispielsweise den Speicherort eines Bildes oder eine Seite, zu der Sie eine Verknüpfung herstellen wollen). Andere Attribute können nur einen Wert aus einer vordefinierten Liste annehmen (wie beispielsweise die Optionen zur Ausrichtung von Text in der Zelle einer Tabelle).

Die HTML-4.01- und die XHTML-1.0-Spezifikation definieren genau, welche Attribute Sie mit jedem Element verwenden können und welche Werte (falls explizit definiert) jedes Attribut annehmen kann.



In allen Kapiteln in den Teilen II und III dieses Buches wird beschrieben, welche Attribute Sie mit den behandelten (X)HTML-Elementen verwenden können.

Entities

Text macht das Web möglich, aber er hat seine Grenzen. *Entities* sind spezielle Zeichen, die auf einer Webseite angezeigt werden können.

Nicht-ASCII-Zeichen

Der American Standard Code for Information Interchange (ASCII) für Text definiert eine ziemlich kleine Anzahl von Zeichen. Es gibt Zeichen, die in dem einfachen ASCII-Zeichensatz nicht enthalten sind – etwa das *Handelsmarkensymbol*, *Brüche* oder *Zeichen mit Akzenten*.

Beispielsweise enthält die Liste der Weißweinsorten aus Abbildung 1.4 zwei *e*-Zeichen (é) mit einem Akzent und zwei deutsche Umlaute (ü).

Weil der ASCII-Zeichensatz weder das akzentuierte *é* noch den deutschen Umlaut *ü* enthält, verwendet HTML stattdessen so genannte *Entities*, um diese Zeichen zu repräsentieren. Wenn der Browser auf ein Entity stößt, ersetzt er es durch das referenzierte Zeichen. Entities beginnen mit einem kaufmännischen Und-Zeichen (&) und enden mit einem Semikolon (;). In dem folgenden HTML-Code werden die Entities fett dargestellt:



Abbildung 1.4: Der ASCII-Zeichensatz kann nicht alle Textzeichen repräsentieren, sodass Sie stattdessen HTML-Entitäten benutzen müssen.

```
<html>
<head>
<title>Weinsorten</title>
</head>

<body bgcolor="#FFFFFF">
  <h2>Weißweinsorten</h2>
  <ul>
    <li>Chardonnay</li>
    <li>Chenin Blanc</li>
    <li>Fum&eacute; Blanc</li>
    <li>Gew&uuml;rtraminer</li>
    <li>Gr&uuml;ner Veltliner</li>
    <li>Marsanne</li>
    <li>Muscat</li>
    <li>Pinot Blanc</li>
    <li>Pinot Gris</li>
    <li>Reising</li>
    <li>Sauvignon Blanc</li>
    <li>S&eacute;millon</li>
    <li>Trebbiano</li>
    <li>Viognie</li>
  </ul>
</body>
</(X)HTML>
```

Das akzentuierte é wird durch das Entity é und der deutsche Umlaut ü durch das Entity ü repräsentiert.

(Anmerkung des Übersetzers: Die aktuellen Browser-Versionen sind sehr wohl in der Lage, die deutschen Umlaute *ä, ö, ü, Ä, Ö, Ü* sowie das *ß* direkt, das heißt: ohne Entity, darzustellen. Deswegen werden die Umlaute in übersetzten Textbeispielen auch direkt dargestellt.)

(X) HTML-Zeichencodes

Die Verschlüsselungen für den ISO-Latin-1-Zeichensatz werden standardmäßig zur Verfügung gestellt und die zugehörigen Entities können ohne spezielle Anstrengungen aufgerufen werden (Tabelle 1.1 enthält einen Verweis auf eine komplette Tabelle). Aber bei den anderen Verschlüsselungen aus Tabelle 1.1 sind spezielle Auszeichnungen erforderlich, um den Browser anzuweisen, Unicode-Zeichencodes zu interpretieren. Unicode ist ein internationaler Standard (ISO Standard 10646), der genügend Zeichencodes enthält, um die meisten Alphabete der Erde sowie zahlreiche andere Symbole und nicht-alphabetische Zeichen zu verschlüsseln. Diese speziellen Auszeichnungen haben folgende Form:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF 8">
```

Wenn UTF-8 als Wert von `charset` angegeben wird, können Sie die gebräuchlichen Unicode-Codetabellen aus Tabelle 1.1 verwenden.

Name	URL
Unicode Code Charts	www.unicode.org/charts/
ISO-Latin-1-Zeichensatz	www.htmlhelp.com/reference/charset/
Griechische Zeichen	www.unicode.org/charts/PDF/U0370.pdf
Währungssymbole	www.unicode.org/charts/PDF/U20A0.pdf
Diverse Symbole	www.unicode.org/charts/PDF/U2600.pdf
Pfeile	www.unicode.org/charts/PDF/U27F0.pdf www.unicode.org/charts/PDF/U2900.pdf
Mathematische Zeichen (es gibt sechs relevante Codetabellen)	Suchen Sie auf www.unicode.org/charts/ nach <code>math</code>
Allgemeine Satzzeichen	www.unicode.org/charts/PDF/U2000.pdf

Tabelle 1.1: Online-Verweise auf (X)HTML-Zeichencodes

Tag-Zeichen

HTML-fähige Software geht davon aus, dass einige HTML-Zeichen wie beispielsweise das Größer-als- und das Kleiner-als-Zeichen auf der fertigen Webseite nicht angezeigt, sondern verborgen werden sollten. Die folgenden Entities zeigen Zeichen an, die normalerweise als Teil von HTML-Auszeichnungen verborgen werden:

- ✓ Kleiner-als-Zeichen (<): `<`
- ✓ Größer-als-Zeichen (>): `>`
- ✓ Kaufmännisches Und-Zeichen (&): `&`



Die Zeichen `<` und `>` werden als Teil von Auszeichnungen verwendet, sind aber *Anweisungen an den Browser* und werden selbst nicht auf der Seite angezeigt. Wenn Sie diese Symbole auf einer Webseite anzeigen wollen, müssen Sie die entsprechenden Entities in die Seite einfügen:

```
<p>Das Absatz-Element markiert Textabsätze:</p>  
<p>&lt;p>&gt;Dies ist ein Absatz.&lt;p>/p>&gt;</p>
```

In der ersten Zeile dieses HTML-Codes werden Tags verwendet, um einen einfachen Absatz zu definieren. In der zweiten Zeile werden Entities verwendet, um die Symbole `<` und `>` darzustellen.

Abbildung 1.5 zeigt, wie der Browser diese Entities in Zeichen umwandelt und diesen HTML-Code in seinem Fenster darstellt.



Abbildung 1.5: Um ein *Kleiner-als-Zeichen*, ein *Größer-als-Zeichen* oder ein *kaufmännisches Und-Zeichen* in dem Browserfenster darzustellen, müssen Sie Entities verwenden.

Die Komponenten einer Webseite

Kommentare schließen Text in (X)HTML-Dateien ein, der auf der endgültigen Seite nicht angezeigt wird. Ein Kommentar wird durch die folgenden Auszeichnungen gekennzeichnet:

- ✓ Der Kommentar beginnt mit dem String `<!--`.
- ✓ Der Kommentar endet mit dem String `-->`.

In dem folgenden Code erklären Kommentare, wie die einzelnen Auszeichnungselemente funktionieren und wo sie in der HTML-Auszeichnungshierarchie angesiedelt sind.

Elemente sind in einer Struktur organisiert:

- ✓ Einige Elemente können nur innerhalb von anderen Elementen vorkommen.
- ✓ Einige Elemente müssen in allen wohlstrukturierten (X)HTML-Dokumenten vorkommen.

```
<html> <!-- Dieses Tag sollte immer am Anfang oder nahe am Anfang eines
wohlgeformten HTML-Dokuments stehen. -->
<head> <!-- Das head-Element liefert Informationen über das
HTML-Dokument insgesamt. -->
<title>Weinsorten</title> <!-- Der Text in dem title-Element wird in der
Titelleiste des Browserfensters angezeigt,
wenn die Seite dargestellt wird. -->
</head> <!-- Ende des head-Elements -->

<body bgcolor="#FFFFFF"> <!-- Der Inhalt einer Webseite wird
innerhalb des body-Elements festgelegt. -->
  <h2>Weißweine</h2> <!-- Überschriftenelemente beginnen mit dem
Buchstaben h, gefolgt von einer Zahl von 1 bis 6, die die
Hierarchie anzeigt.
Dies ist eine Überschrift der Stufe 2, h2. -->
  <ul> <!-- Dies ist das Element einer ungeordneten Liste, ul, das eine
Liste von Einträgen mit Aufzählungszeichen, li, erzeugt. -->
    <li>Chardonnay</li> <!-- Eine Reihe von Listeneinträgen -->
    <li>Chenin Blanc</li>
    <li>Fum&eacute; Blanc</li>
    <li>Gew&uuml;rztaminer</li>
    <li>Gr&uuml;ner Veltliner</li>
    <li>Marsanne</li>
    <li>Muscat</li>
    <li>Pinot Blanc</li>
    <li>Pinot Gris</li>
    <li>Riesling</li>
    <li>Sauvignon Blanc</li>
    <li>S&eacute;millon</li>
    <li>Trebiano</li>
    <li>Viognie</li>
  </ul> <!-- Ende der ungeordneten Liste -->
</body> <!-- Ende des body-Abschnitts -->
</html> <!-- Ende des HTML-Dokuments -->
```

Das vorstehende Dokument ist in ein `head`- und ein `body`-Element unterteilt. In jedem Abschnitt stehen bestimmte Arten von Elementen. Viele Kombinationen sind möglich und werden in diesem Buch immer wieder gezeigt!

HTML-Text gliedern

Über die Unterteilung in `head`- und `body`-Abschnitte hinaus kann Text in HTML-Dokumenten auf vielfältige Weise gegliedert werden.

Dokumentenkopf

In dem `head`-Abschnitt können Sie neben einem Titel diverse Labels und Informationen definieren, die hauptsächlich dazu dienen, das folgende Dokument zu beschreiben wie beispielsweise die verwendeten Zeichensätze, aufzurufende Skripts und Stilinformationen. Die eigentlichen Inhalte und die meisten (X)HTML-Elemente befinden sich in dem `body`-Abschnitt.

Überschriften

Überschriften (engl. *headings*) werden durch die Elemente `h1` bis `h6` gekennzeichnet. Sie haben nichts mit dem `head` des HTML-Dokuments zu tun. Einzelne Überschriften strukturieren den Inhalt, und der `head` identifiziert oder beschreibt das Dokument insgesamt.



In dem Beispiel der Weinsorten bildet das `h2`-Element den Titel einer Liste von Traubensorten.

Absätze und anderes

Wenn Sie auf einer Webseite Fließtext anzeigen wollen, können Sie ihn mit dem Absatzelement `p` (das aus den Tags `<p>` und `</p>` besteht), in Absätze unterteilen. Außerdem können Sie

- ✓ mit dem `break`-Element `
` Zeilenumbrüche erzwingen,
- ✓ mit den `<hr />`-Element horizontale Balken (Linien) erzeugen.

HTML verfügt außerdem über diverse Möglichkeiten, Text innerhalb von Absätzen hervorzuheben oder zu identifizieren (siehe die Teile II und III dieses Buches).

Listen

Mit HTML können Sie leicht ungeordnete Listen (Aufzählungslisten) erzeugen. Mit verschiedenen Mechanismen können Sie auch andere Arten von Listen einschließlich nummerierter Listen erstellen. Listen können in Listen eingebettet werden, um beliebig tiefe Gliederungen oder Inhaltsverzeichnisse komplexer Themen zu generieren (siehe Kapitel 5).

Tabellen

Mit HTML können Sie auch diverse Tabellen erstellen (siehe Kapitel 11) und strukturieren. Innerhalb einer Tabelle können Sie

- ✓ Spaltenüberschriften und Tabellendaten unterscheiden
- ✓ das Layout von Zeilen und Spalten an Ihre Anforderungen anpassen

Bilder in HTML-Dokumenten

Ein Bild in ein HTML-Dokument einzufügen, ist einfach. Durch passend gewählte Bilder können Sie Webseiten erheblich attraktiver gestalten. In Kapitel 7 erfahren Sie, wie Sie Bilder aus Dateien einbinden können. In Kapitel 9 wird gezeigt, mit welchen komplexen Auszeichnungen Sie den Textfluss um Grafiken herum festlegen können. Tipps zur Auswahl geeigneter Bilder runden das Thema ab.

Links und Navigationswerkzeuge

Die Struktur einer Webseite sollte den Besucher dabei unterstützen, sich schnell und leicht in einer Sammlung von Webseiten zurechtzufinden und gesuchte Inhalte zu finden. Links sind der Mechanismus, über den Besucher zu Ihren Webseiten finden. Deshalb werden in Kapitel 6 die folgenden Themen behandelt:

- ✓ Externe Komponenten oder Ressourcen referenzieren
- ✓ Von einer Seite zur nächsten springen
- ✓ Innerhalb einer Seite herumspringen
- ✓ Seiten organisieren und strukturieren

Struktur und Organisation werden umso wichtiger, je mehr Informationen Sie für Ihre Besucher bereithalten.

Navigationswerkzeuge etablieren Standardmethoden und Hilfsmittel zur Navigation innerhalb einer Website. Sie präsentieren den Besuchern Ihrer Website die Struktur der gesamten Site und stellen Mittel wie beispielsweise Menüs zur Verfügung, mit denen sich der Anwender schnell und leicht auf Ihrer Site bewegen kann.

Alles in allem sollte das Ergebnis eine wohlgeordnete Sammlung von Informationen und Bildern sein, die leicht zu verstehen, zu verwenden und zu erreichen sind.