

Einleitung

In dieser Einleitung:

| | |
|------------------------------------------|----|
| Danksagung | 14 |
| Einführung | 14 |
| Was ist HTML? | 14 |
| Der Einsatz der Übungsdateien | 20 |
| Weitere Hilfen | 22 |
| Konventionen und Features in diesem Buch | 22 |

Danksagung

Ich möchte mich ganz herzlich bei der wunderbaren Redaktion von O'Reilly Media bedanken, die dieses Buch problemlos und glatt durch den Prozess der Bearbeitung und Produktion begleitet hat. Dies ist mein erstes Buch für O'Reilly, und ich hoffe definitiv, dass es nicht das letzte ist. Es war wirklich eine Freude, mit Ihnen allen zu arbeiten.

Einführung

Hypertext Markup Language (HTML) ist die zugrunde liegende Auszeichnungssprache des World Wide Web. Sie ist der Grundstoff, aus dem praktisch jede Site gewebt ist – egal ob es sich dabei um solche Riesenprojekte wie die Firmen-Homepage von Microsoft handelt oder um die Projektarbeit an einer Grundschule, die nur aus einer Seite besteht.

Lassen Sie sich von dem Begriff *Auszeichnungssprache (markup language)* nicht einschüchtern. Mit einer solchen Sprache wird einfacher Text kommentiert oder *ausgezeichnet*, damit ein Browser weiß, wie er ihn formatieren soll, damit er auf einer Webseite gut aussieht. Es ist ganz einfach, hier den Einstieg zu finden. Tatsächlich können Sie eine einfache Webseite in nur wenigen Minuten erstellen. Es gibt zwar komplett ausgestattete WYSIWYG-Tools (das steht für »What You See Is What You Get«, also so wie die Seite auf dem Bildschirm dargestellt wird, ist sie hinterher im Internet zu sehen), mit denen man den Vorgang beschleunigen kann, Webseiten zu schreiben. Doch im Grunde brauchen Sie nur ein gewöhnliches Programm zur Textbearbeitung wie den Editor von Microsoft und keine spezielle Software oder umfassende Schulung.

In dieser Einführung lernen Sie einige Grundlagen über HTML. Sie lernen, wie man damit reinen Text auf ansprechende Weise formatieren kann, wie man Grafiken und Hyperlinks einbindet und Webinhalte in praktisch jedem Programm erstellen kann, das Text bearbeiten kann. Diese Einführung erläutert, was Cascading Style Sheets (CSS) sind und wie damit für eine einheitliche Formatierung auf großen Websites gesorgt wird. Sie lernen außerdem die Unterschiede zwischen HTML4, XHTML und HTML5 kennen, damit Sie die wichtige Entscheidung treffen können, zu welcher HTML-Version Ihr Code konform sein soll. Schließlich bekommen Sie Infos über die in diesem Buch verwendeten Konventionen, z.B. spezielle Hilfen wie Anmerkungen, Tipps, Warnhinweise und Referenzen auf die Datendateien.

Was ist HTML?

Einfach ausgedrückt ist eine *Webseite* (oder ein *HTML-Dokument*) eine einfache Textdatei, die anhand der *Hypertext Markup Language* (HTML) kodiert wurde, damit sie in einem Browser ansprechend formatiert dargestellt werden kann. Eine *Website* hingegen ist die Gesamtheit aller Webseiten, die unter einer Internetadresse zu finden sind. Der Begriff HTML bedeutet Wort für Wort aufgeschlüsselt Folgendes:

- **Hypertext** Auf diesen Text kann man klicken, um von einem Dokument zum nächsten zu springen. Das bezieht sich auf die Möglichkeit, Webseiten miteinander zu verknüpfen.
- **Markup** Mit den so genannten *Tags* (engl. für Etikett oder Aufkleber) werden Layout- und Formatierungsangaben auf den Text angewendet. Wörtlich genommen wird der Text mit den Tags »ausgezeichnet« (*marked up*).
- **Language** Eine Anspielung auf die Tatsache, dass HTML als Programmiersprache betrachtet wird

TIPP Wenn von Computer programmieren die Rede ist, denkt man gewöhnlich daran, dass ein kompiliertes Programm geschrieben wird. Bei einer kompilierten Programmiersprache wird der von Menschen lesbare Programmiercode von einer Software in eine ausführbare Datei konvertiert (die dann meist die Endung `.exe` oder `.com` bekommt) und an den Endbenutzer weitergegeben. Im Kontrast dazu ist HTML eine interpretierte Programmiersprache. Das bedeutet, dass das Programm in einer für Menschen lesbaren Form an die Endbenutzer weitergegeben wird, und das Programm, in dem dieser Code dann geöffnet wird, kümmert sich darum, wie er auszuführen ist. Der HTML-Code für Webseiten befindet sich in Dateien. Immer wenn Ihr Browser eine Webseite öffnet, verarbeitet er den in dieser Datei befindlichen HTML-Code.

Die Bedeutung von HTML-Tags

Der Code in einer HTML-Datei besteht aus Text, der von *Tags* eingefasst wird. Diese Tags geben an, wo die Formatierung erfolgen soll, wie das Layout erscheinen soll, welche Bilder an welcher Stelle eingesetzt werden sollen usw.

Sie wollen beispielsweise ein bestimmtes Wort kursiv setzen:

Alles muss raus.

In HTML gibt es nicht wie in einem Textverarbeitungsprogramm eine Schaltfläche für Kursivschrift. Also müssen Sie das Wort, das kursiv erscheinen soll, »taggen«, d.h. kennzeichnen. Der Code, um die Kursivschrift einzuschalten, ist `<i>`, und zum Ausschalten schreiben Sie `</i>`. Der HTML-Code sieht dann also wie folgt aus:

```
<i>Alles</i> muss raus.
```

Dies ist ein Beispiel für ein *zweiteiliges* oder auch *umschließendes Tag*, das Text in öffnende und schließende Tags einfasst, in diesem Fall `<i>` und `</i>`. Beachten Sie den Schrägstrich (Slash) im schließenden Tag (`</i>`). Mit diesem Slash wird ein öffnendes Tag von einem schließenden Tag unterschieden. Bei einem zweiteiligen Tag gibt es für jedes öffnende auch ein dazugehöriges schließendes Tag.

Zu diesem System mit den Tags ist es gekommen, weil damals, als das Internet noch in den Kinderschuhen steckte, praktisch jeder, der ins Internet ging, die Verbindung mit einem Einwahlmodem aufbaute und dafür Geschwindigkeiten nutzte, die zwischen 2400 bps und 28.8 Kbps lagen. Das ist *wirklich* sehr langsam. Textdateien werden viel schneller übertragen als binäre Dateien, also musste jegliche Art eines Systems der Informationsweitergabe, das populär werden wollte, auf Text basieren. Anderenfalls wären in der Zeit, die eine Seite zum Laden braucht, die Leute eingeschlafen.

Wer Webseiten gestaltete, wollte außerdem, dass seine Seiten ansprechend und attraktiv sind. Man konnte nicht einfach die Seiten in einer Textverarbeitung formatieren, weil jedes Programm mit der Formatierung unterschiedlich umging. Außerdem konnte man unmöglich wissen, welches Textverarbeitungsprogramm jemand verwendet, der eine Site besucht. Die Dateien aus einem Textverarbeitungsprogramm sind außerdem viel größer als reine Textdateien.

Die Schöpfer des Internet schufen eine elegante Lösung. Anstatt die formatierten Seiten übers Internet zu senden, schufen sie eine Applikation, den so genannten Webbrowser oder einfach Browser, der reinen Textcode (HTML-Tags) als Formatierungsanweisungen interpretieren konnte. Der Text konnte nun schnell und effizient als Text gesendet werden. Auf dem lokalen PC wird er dann verarbeitet und ansprechend dargestellt.

HTML funktionierte für sich genommen bei allen Arten der Textformatierung hervorragend, doch einige Webdesigner wünschten sich auf ihren Seiten Bilder und Grafiken. Um dem zu entsprechen, wurde das ``-Tag geschaffen, mit dem Designer auf eine auf dem Server gespeicherte Grafik verweisen konnten. Wenn der Browser zu diesem Tag kommt, fordert er den Download dieser Bilddatei vom Server an und stellt sie auf der Seite dar. Wie man Bilder einfügt, erfahren Sie in Kapitel 9, »Grafiken darstellen«.

Das ``-Tag unterscheidet sich in verschiedener Hinsicht vom `<i>`-Tag. Es ist *freistehend*, was bedeutet, dass es kein schließendes Tag hat, und es akzeptiert Attribute. Ein Attribut ist Text innerhalb des Tags, in dem Informationen enthalten sind, wie sich das Tag verhalten soll. Für ein ``-Tag müssen Sie beispielsweise eine Quelle (*source*) angeben, was `src` abgekürzt wird. Hier folgt ein Beispiel:

```

```

Dieses ``-Tag verwendet das Attribut `src=` und gibt an, dass die Datei *baum.gif* dargestellt werden soll. Viele Tags akzeptieren Attribute, die entweder optional oder erforderlich sind. In den Übungen dieses Buchs werden Sie viele Beispiele dafür finden.

Mit HTML können Sie auch *Hyperlinks* von einer Seite zu einer anderen erstellen. Wenn ein Besucher auf einer Website auf einen Hyperlink klickt, lädt der Browser die Seite, auf die verwiesen wird, oder springt zu einem gekennzeichneten Abschnitt (ein »Lesezeichen«) auf der gleichen Seite. Wie man Hyperlinks erstellt, erfahren Sie in Kapitel 5, »Erstellen von Hyperlinks und Anker«.

Das Tag für einen Hyperlink ist `<a>`, ein zweiteiliges Tag, doch die meisten Leute würden es ohne das Attribut, das die Datei oder den Standort, wohin gesprungen werden soll beschreibt, nicht erkennen. Um beispielsweise einen Hyperlink mit den Worten *Hier klicken* zu erstellen, durch den beim Anklicken zu der Datei `index.htm` gesprungen wird, nimmt man folgende Codierung:

```
<a href="index.htm">Hier klicken</a>
```

Natürlich gibt es noch viel mehr HTML, doch dies ist die grundlegende Funktionsweise. Reiner Text wird mit Tags ausgezeichnet, die kennzeichnen, wo solche Elemente wie Formatierungen, Hyperlinks und Grafiken eingesetzt werden sollen, und ein Browser interpretiert diese Tags und stellt die Seite in formatiertem Zustand dar. Der Trick besteht natürlich darin zu wissen, mit welchen Tags man was erreicht und wo sie angemessen sind und welche Attribute sie brauchen. Genau das ist das Thema dieses Buchs.

Der Umgang mit Cascading Style Sheets

Wer als Webdesigner mit früheren Versionen von HTML gearbeitet hat, um große Websites zu erstellen, war häufig davon frustriert, wie oft er immer wieder die gleichen Sachen eingeben musste. Nehmen wir an, dass eine Website aus 200 Seiten besteht, die alle das gleiche Grundlayout und Design aufweisen. Um eine Änderung des Designs bei der gesamten Site vorzunehmen, müsste ein Designer sich alle 200 Seiten vornehmen und jede einzeln manuell bearbeiten.

Spätere Versionen von HTML haben sich dieser Problematik angenommen und unterstützen nun Cascading Style Sheets. Basierend auf dem gleichen Prinzip, nach dem Formatvorlagen in einem Textverarbeitungs- oder Layoutprogramm arbeiten, können Webdesigner nun anhand von Cascading Style Sheets die Formatierung für einen bestimmten Tag-Typ festlegen. Dafür wird üblicherweise ein separates Stylesheet-Dokument erstellt und dieses Stylesheet dann auf mehrere Seiten angewendet. Müssen Sie am Stil der Seite etwas ändern? Bearbeiten Sie einfach das Stylesheet, und automatisch werden alle Seiten ebenfalls geändert.

Obwohl Sie Dokumente auch weiterhin anhand der älteren Methoden formatieren können (das lernen Sie auch ein wenig in diesem Buch), verlassen sich die meisten Webdesigner beim Formatieren heutzutage fast ausschließlich auf Cascading Style Sheets, und bei XHTML ist es sogar Pflicht, dass Sie damit arbeiten.

Zuerst scheuen Sie vielleicht den Extraaufwand, doch wenn Sie eine Site mit vielen einzelnen Seiten erstellen, wird es sich um ein Vielfaches auszahlen, wenn Sie dafür mit einem Cascading Style Sheet arbeiten.

Warum soll man HTML in Editor lernen?

In diesem Buch wird die HTML-Programmierung auf Einsteigerebene gelehrt, dies aber auf recht fundamentalistische Weise: indem in Editor reine Textdateien erstellt werden. Auf dem Markt gibt es heutzutage so viele gute Programme zur Erstellung von Webseiten, dass Sie sich wohl fragen, warum wir hier so vorgehen.

Einfach ausgedrückt liegt das daran, dass es die beste Art und Weise ist, HTML zu lernen, wenn Sie Ihre Codezeilen selbst schreiben. In diesem Buch erfahren Sie, wie Sie eine Website von Grund auf erstellen und dabei jede einzelne Codezeile selbst schreiben. Es ist langsamer und vielleicht nicht so lustig wie bei den schicken grafischen Programmen, aber ein hervorragendes Training.

Im letzten Kapitel dieses Buchs wird gezeigt, wie man Webinhalte mit Microsoft Expression Web erstellt, und vielleicht entscheiden Sie sich später für ein solches Programm. Allerdings werden Sie ein deutlich besserer Webdesigner und können deutlich besser nachvollziehen, was in solchen Designprogrammen abläuft, wenn Sie anfangs die Arbeit mit Editor durchstehen.

Die Wahl der HTML-Version

Bei den verschiedenen HTML-Versionen werden für manche Arten von Inhalten unterschiedliche Tags verwendet, obwohl sie insgesamt eher ähnlich als unterschiedlich sind, vor allem auf der Einsteigerebene, die in diesem Buch abgedeckt wird. Im Folgenden werden die verschiedenen HTML-Versionen vorgestellt, auf die Sie treffen können:

- **HTML4** Ein sehr stabiler, allgemein akzeptierter Codesatz, der bei kleineren Coding-Fehlern recht nachsichtig ist. Die Verwendung von HTML4 ist wünschenswert, wenn es vor allem auf die Kompatibilität mit allen Browsern ankommt.
- **XHTML** Eine strikte, auf Standards basierende Implementierung von HTML4, die mit XML (eXtensible Markup Language) erstellt wurde. Beim XHTML-Coding werden die gleichen Codes wie in HTML4 verwendet, und somit ist es auch kompatibel mit den gleichen Browsern wie HTML4 (mehr darüber erfahren Sie im Kasten über XML).
- **HTML5** Ein überarbeiteter Codesatz, der auf HTML4 aufbaut, um neue Fähigkeiten zu ergänzen

HTML5 bietet viele dramatische Verbesserungen in den Bereichen Anwendungsverarbeitung und Multimedia, doch eine Menge dieser Eigenschaften liegen außerhalb des Spektrums dieses Buchs. Was die grundlegende Codierung angeht (und das ist Thema dieses Lehrbuchs), ist der größte Unterschied, dass es neue, spezifische Codes für unterschiedliche Inhaltsarten gibt, die früher mit eher allgemeineren Codes abgewickelt wurden. Beispielsweise besitzt HTML5 die Tags `<audio>` und `<video>` um Multimedia-Inhalte einzubauen, während HTML4 alle Arten von Multimedia-Inhalten über ein generisches `<embed>`-Tag einfügt.

Weil sich dieses Buch mit HTML5 beschäftigt, könnte man sich naheliegenderweise entscheiden, die Codierungen generell mit HTML5-Tags vorzunehmen, doch in der realen Welt läuft es nicht ganz so einfach.

Ein guter Browser sollte idealerweise jedes Tag und jede Version von HTML unterstützen, weil die Unterschiede der verschiedenen HTML-Versionen für den Besucher der Website völlig unsichtbar sein sollten. Allerdings ist HTML5 so neu, dass nicht alle Browser hier auf dem neuesten Stand sind, und wer einen älteren Computer nutzt, hat vielleicht noch nicht die aktuelle Version eines Browsers, auch wenn eine zu HTML5 kompatible Version bereits erhältlich ist.

TIPP Auf dieser Site wird aufgelistet, welche HTML5-Features von jeder Version aller bekannten Browser unterstützt werden: <http://caniuse.com>.

Der Code, den Sie beim Durcharbeiten der Übungen in diesem Buch erstellen werden, basiert auf HTML5, aber ich werde Ihnen auch einige Problemumgehungen (Workarounds) zeigen, wenn HTML5-Code in bestimmten Browsern Probleme verursacht. Sie werden beide Wege kennenlernen, wie ein bestimmter Effekt erstellt wird, damit Sie selbst entscheiden können, welche Codes Sie im Bedarfsfall in welchen Situationen einsetzen.

Was ist XML und XHTML?

Es gibt eine mit HTML zusammenhängende Sprache namens eXtensible Markup Language (XML), mit der Programmierer ihre eigenen Tags erstellen können. Sie wird allgemein beispielsweise für Webdatenbanken eingesetzt, weil damit die Tags für jedes Datenfeld definiert werden können. Weil man XML so umfassend an die eigenen Zwecke anpassen kann, können Programmierer damit praktisch jede andere Auszeichnungssprache schaffen, einfach indem all die offiziell akzeptierten Tags dieser Sprache neu erstellt werden. Das W3C hat genau das gemacht: Dort wurde die gesamte HTML-Sprache in XML neu erstellt und als Extensible HTML (XHTML) bezeichnet. Die Version 1.0 erschien 2001, und die aktuelle Version XHTML 2.0 kam 2004 heraus.

XHTML ist also ein HTML, das innerhalb der umfassenderen Sprache XML geschrieben wurde. Weil sie in ihrer Funktionalität praktisch identisch ist mit HTML, ist der Basissatz der Tags der gleiche, und Sie können HTML und XHTML gleichzeitig lernen. Mit XHTML können Sie auch neue Tags und Erweiterungen schaffen – für erfahrene Webentwickler ein wertvolles Feature.

Es gibt nur eines, worauf man bei XHTML achten muss: Es verzeiht keine Fehler. HTML erwartet beispielsweise technisch gesehen, dass Sie jeden Absatz mit `<p>` beginnen und ihn mit `</p>` abschließen. Aber bei HTML können Sie das schließende `</p>`-Tag auch weglassen, wenn Sie wollen (oder es vergessen haben). Bei XHTML funktioniert das aber nicht. Es gibt noch viele andere kleine Aspekte, wo XHTML derart pingelig ist.

Von XHTML hat man mal gedacht, dass es HTML4 als Nachfolger ersetzen würde, doch aufgrund von Interoperabilitätsproblemen ist dies noch nicht geschehen. Stattdessen setzt HTML5 an, HTML4 zu beerben. In diesem Buch wird XHTML nicht explizit ausgeführt, doch das Meiste, was Sie hier lernen, kann auch auf das Coding mit XHTML angewendet werden.

Warum in HTML5 codieren?

Die kurze Antwort lautet: Sie sollten Ihren Code in HTML5 schreiben, weil es eine Investition in die Zukunft ist. In nur wenigen Jahren wird es der Standard sein, auf dem praktisch alle Websites basieren.

Eine etwas längere Antwort wäre, dass damit sauberer, leichter zu schreibender Code möglich ist. Die Technologie von Webseiten ist sprunghaft angewachsen, was größtenteils an den deutlich schnelleren Internetverbindungen liegt, die allgemein zur Verfügung stehen, aber auch daran, dass Benutzer, Designer und Programmierer von ihren Webseiten immer mehr Funktionalität verlangen, z.B. die präzisere Steuerung von Fonts und Layout, die bessere Darstellung (Rendering) auf Geräten in einem wilden Spektrum von Handys bis zu Desktop-Monitoren, besseren Bildern, mehr Interaktivität, Video, Audio, Animationen und bessere Unterstützung für verschiedene Bild- und Dateiformate. Weil schnelle Internetverbindungen weithin verfügbar sind, wollen die Leute nicht mehr lange darauf warten, bis die Seiten mit großen Audio- und Videodateien geladen sind, und das bedeutet, dass es auch immer mehr Sites gibt, auf denen Audio- und Videoinhalte integriert sind.

HTML wurde ursprünglich nicht für die Umständlichkeiten bei der Übertragung von Multimedia-Inhalten geschaffen, und darum wechseln immer mehr High-End-Profiseiten zu anderen Sprachen und Technologien, die auf HTML aufsetzen, um diese Inhalte zu übermitteln, z.B. JavaScript, Java und Active Server Pages (ASP).

In HTML5 gibt es einige wichtige neue Tags, damit Audio, Video und Anwendungen reibungsloser und verlässlicher integriert werden können. Über viele dieser neuen Tags, z.B. `<audio>`, `<video>` und `<canvas>`, erfahren Sie mehr in den Kapiteln 15 und 16.

In HTML5 werden einige ältere Tags nicht mehr unterstützt. Beispielsweise konnte man in den Zeiten vor HTML4 eine Schrift über das Tag `` angeben. Heutzutage wird das meist über Cascading Style Sheets erledigt, von daher wurde dieses Tag schon längere Zeit von Webdesignern sowieso nicht mehr eingesetzt. Bei HTML5 ist dieses Tag formell aus der Sprache entfernt worden.

Eines der wesentlichsten Dinge, die in HTML5 nun fehlen, ist die Fähigkeit, Websites anhand der Befehle `<frame>` und `<frameset>` mit mehreren Rahmen (Frames) zu erstellen. Sie können weiterhin Websites mit verschiedenen Abschnitten bauen, doch das löst man nun kompetenter anhand von Tabellen oder Abschnitten (*divisions*). Kapitel 11 behandelt diese Abschnitte, und zwar auf diese neuere Weise, die von den meisten Webdesignern bevorzugt wird. In den Kapiteln 12 und 13 geht es um Tabellen – ein immer noch akzeptabler Weg, der von vielen Gelegenheitswebdesignern eingeschlagen wird, die aus Programmen wie Microsoft Word solche Tabellen kennen.

Minimale Systemanforderungen

Für die Entwicklung mit HTML gibt es keine minimalen Systemanforderungen. Sie können dafür jedes Textverarbeitungsprogramm und jede Art von Computer und Betriebssystem nehmen. Das ist ja das Schöne an HTML! In diesem Buch arbeiten wir mit Editor, doch Sie können auch jedes andere Editorprogramm nehmen.

Um Ihre Arbeiten zu testen, brauchen Sie einen HTML5-konformen Browser, zum Beispiel die neuesten Versionen von Internet Explorer 9, Google Chrome und Firefox.

Der Einsatz der Übungsdateien

Bei jeder Übung wird in den Lektionen in einem Abschnitt erklärt, welche Dateien für diese Übung benötigt werden, und welche Vorbereitungen erforderlich sind, bevor Sie sich an die jeweilige Übung machen. Die Übungsdateien finden Sie unter <http://www.microsoft-press.de/support> auf Deutsch (hier tragen Sie die ISBN dieses Buchs ein, die unten rechts auf der Rückseite des Buchs steht). Die englischen Originaldateien sind unter <http://oreilly.com/catalog/0790145302083/> erhältlich. Wenn Sie die heruntergeladene Datei entpacken, werden für jedes Kapitel separate Ordner erstellt und darin dann wieder getrennte Ordner für die jeweiligen Übungen.

In der folgenden Tabelle werden die Ordner für die Übungsdateien nach Kapiteln und die dazugehörigen Unterordner aufgeführt. Für jedes Kapitel gibt es bei den Übungsdateien auch einen Unterordner namens *_Solutions* (Lösungen), in dem die fertigen Versionen der im jeweiligen Kapitel verwendeten Übungsdateien enthalten sind. Beim deutschen Quellcode tragen die Unterordner den englischen Namen, die hier in der Tabelle übersetzt sind.

| Kapitel | Ordner | Unterordner | Deutsch |
|-------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kapitel 1: Die Grundlagen von HTML und XHTML | 01Editing | Keine Unterordner | <i>Datenaufbereitung</i> |
| Kapitel 2: Die Dokumentstruktur einrichten | 02Structure | CreatingParagraphs PublishingFiles SpecifyingKeywords SpecifyingTitle | <i>Struktur</i> Absätze erstellen Dateien veröffentlichen Schlüsselwörter festlegen Titel festlegen |
| Kapitel 3: Text mit Tags formatieren | 03Format | ApplyingBold ApplyingSuperscript ConfiguringSettings CreatingHeadings FormattingQuotes UsingMonospace | <i>Format</i> Fettschrift anwenden Hochgestellt anwenden Einstellungen konfigurieren Überschriften erstellen Zitate formatieren Monospace einsetzen |
| Kapitel 4: Die Arbeit mit Listen und Hintergründen | 04Lists | ChoosingColors CreatingGlossary InsertingCharacters InsertingLines NestingLists SpecifyingImages | <i>Listen</i> Farben wählen Glossar erstellen Zeichen einfügen Zeilen einfügen Listen verschachteln Bilder einsetzen |
| Kapitel 5: Hyperlinks und Anker erstellen | 05Links | CreatingAnchors CreatingHyperlinks LinkingEmail LinkingOther | <i>Links</i> Anker erstellen Hyperlinks erstellen E-Mail-Links erstellen Andere Links |

| Kapitel | Ordner | Unterordner | Deutsch |
|----------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kapitel 6: Einführung in Stylesheets | 06Styles | ConstructingRules CreatingClasses CreatingExternal CreatingNested StylingHyperlinks | <i>Stylesheets</i> Regeln formulieren Klassen erstellen Externe Stylesheets erstellen Verschachtelte Stylesheets erstellen Hyperlinks stylen |
| Kapitel 7: Text mit Stylesheets formatieren | 07Text | AdjustingSpacing ApplyingBold ApplyingStrike CreatingSpan SelectingFont SelectingSize | <i>Text</i> Abstände anpassen Fettschrift anwenden Durchstreichungen anwenden Spans erstellen Schriftarten wählen Schriftgröße wählen |
| Kapitel 8: Absätze mit Stylesheets formatieren | 08Paragraphs | AddingBorders AdjustingHeight Indenting SettingAlignment | <i>Absätze</i> Rahmen einfügen Höhe anpassen Einrückungen Ausrichtungen einstellen |
| Kapitel 9: Bilder darstellen | 09Graphics | CaptioningFigures ClearingImages CreatingHyperlinks InsertingImages SizingImages UsingAlt UsingThumbnails | <i>Bilder</i> Bilder aufnehmen Bilderplatz freiräumen Hyperlinks erstellen Bilder einfügen Bildgröße ändern Alternativtext Thumbnails |
| Kapitel 10: Navigationshilfen erstellen | 10Navigation | CreatingGraphicBar CreatingImageMap CreatingTextBar Redirecting | <i>Navigation</i> Grafikleisten erstellen Imagemaps erstellen Textleisten erstellen Weiterleitungen |
| Kapitel 11: Auf Abschnitten basierende Layouts erstellen | 11Divisions | CreatingDivisions FormattingDivisions PositioningDivisions UsingSemantic | <i>Abschnitte</i> Abschnitte erstellen Abschnitte formatieren Abschnitte positionieren Semantische Tags |
| Kapitel 12: Tabellen erstellen | 12Tables | CreatingTable SettingWidth SpanningCells SpecifyingSize UsingTables | <i>Tabellen</i> Tabellen erstellen Spaltenbreite einstellen Zellen verbinden Größe angeben Arbeit mit Tabellen |
| Kapitel 13: Tabellen formatieren | 13FmtTables | ApplyingBackground ApplyingBorders ChangingPadding | <i>Tabellen formatieren</i> Hintergrund angeben Rahmen festlegen Innenabstand ändern |

| Kapitel | Ordner | Unterordner | Deutsch |
|---------------------------------------------------------|--------------|---------------------------------------------------|----------------------------------------------------------------------------------------|
| Kapitel 14: Benutzerformulare erstellen | 14Forms | CreatingButtons CreatingForms CreatingLists | <i>Formulare</i> Schaltflächen erstellen Formulare erstellen Listen erstellen |
| Kapitel 15: Audio und Video einbinden | 15AudioVideo | | <i>Audio und Video</i> |
| Kapitel 16: JavaScript und externe Inhalte einbinden | 16Canvas | | <i>Canvas</i> |
| Kapitel 17: HTML und Microsoft Expression Web | 17Expression | ViewingPage | <i>Expression</i> Seite betrachten |

Weitere Hilfen

Damit dieses Buch mit allen Inhalten korrekt erscheint, haben wir keine Mühen gescheut. Falls doch einmal Probleme auftauchen, nehmen Sie bitte Kontakt zu den im Folgenden aufgeführten Stellen auf.

Hilfe zu diesem Buch

Wenn es Fragen zum Inhalt dieses Buchs oder den Übungsdateien gibt, konsultieren Sie zuerst die Errata-Seite dieses Buchs, die Sie unter folgender Adresse finden:

<http://www.microsoft-press.de/support>

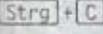
Auf dieser Seite werden Informationen über bekannte Fehler und Korrekturen für dieses Buch veröffentlicht. Wenn Sie hier nicht fündig werden, senden Sie Ihre Frage oder Ihren Kommentar an folgende E-Mail-Adresse:

mSPRESSDE@oreilly.de

Konventionen und Features in diesem Buch

Sie sparen Zeit in der Arbeit mit diesem Buch, wenn Sie verstehen, wie die *Schritt für Schritt*-Buchreihe besondere Anweisungen, zu drückende Tasten, zu klickende Schaltflächen etc. angibt.

| Konvention | Bedeutung |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vorbereitung | Dieses Wort findet sich vor jedem Absatz, in dem Übungen Schritt für Schritt vorgestellt werden. Darin werden Dinge erwähnt, die Sie prüfen sollten, oder vorbereitende Aktionen vor Beginn einer Übung. |
| Aufräumen | Dieses Wort findet sich nach jedem Absatz, in dem Übungen Schritt für Schritt vorgestellt werden. Darin stehen Anweisungen, welche geöffneten Dateien oder Programme geschlossen werden sollen, bevor Sie sich das nächste Thema vornehmen. |
| 1. | Anhand dieser nummerierten Schritte werden Sie durch die praktischen Übungen eines jeden Themas geleitet  |

| Konvention | Bedeutung |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Problemlösung | In diesen Absätzen finden Sie Hinweise, wie man häufig vorkommende Probleme löst, die vielleicht das Vorankommen bei einer Übung behindern |
| Tipp | Hier finden Sie hilfreiche Hinweise oder Abkürzungen, mit denen die Arbeit an einer Aufgabe erleichtert wird |
| Wichtig | In diesen Zeilen werden Informationen zusammengestellt, die Sie zum Abschließen einer Prozedur wissen müssen |
| Anmerkung | Diese Absätze bieten ergänzende oder mit dem Thema zusammenhängende Informationen |
|  | Das Pluszeichen zwischen zwei Tasten bedeutet, dass Sie die erste Taste gedrückt halten müssen, während Sie die zweite drücken. »  +  drücken« bedeutet also: »Halten Sie die Taste  gedrückt, während Sie  drücken.« |
| Benutzeroberflächenelemente | In den Übungen sind damit die Namen von Programmelementen wie Schaltflächen, Befehlen und Dialogfelder gemeint |
| Benutzereingabe | Das meint alles, was Sie eintippen sollen |
| Glossarbegriffe | Diese Begriffe werden am Ende des Buchs im Glossar erläutert |

Wie geht's weiter?

Als Einstieg gehen Sie zu Kapitel 1 und beginnen mit dem Lesen und dem Durcharbeiten der Übungen. Die Lektionen sind so gestaltet, dass sie in der Reihenfolge durchgegangen werden sollten, wie sie im Buch erscheinen, aber Sie können auch frei im Buch umherspringen, wenn Sie nur ein paar Lücken in Ihren HTML-Kenntnissen füllen wollen.