

Kapitel 1

HTML5-Dokumente

Inhalt

- Übergeordnete HTML5-Elemente
- Elemente zum Gliedern des Dokuments
- Alte und neue Elemente

1.1 Grundbestandteile

HTML5 bietet im Grunde drei Funktionen: Struktur, Styling und Funktionalität. Es wurde zwar nie offiziell verkündet, aber HTML5 gilt als das Produkt, das HTML, CSS und JavaScript kombiniert, auch wenn manche APIs und die gesamte CSS3-Spezifikation nicht dazugehören. Diese Technologien sind äußerst zuverlässig und funktionieren als eine Einheit unter dem Dach der HTML5-Spezifikation. HTML kümmert sich um die Struktur, CSS holt diese Struktur und ihren Inhalt auf den Bildschirm und JavaScript macht den Rest, der (wie Sie später noch sehen werden) gar nicht so unerheblich ist.

Trotz der Integration dieser Technologien ist die Struktur immer noch der wichtigste Bestandteil eines Dokuments. Sie stellt alle notwendigen Elemente für die Zuweisung von statischen oder dynamischen Inhalten bereit und dient auch als grundlegende Plattform für Anwendungen. Da heute für die Interaktion mit dem Web sehr unterschiedliche Geräte mit vielfältigen Schnittstellen zu Einsatz kommen, wird ein Grundaspekt, nämlich die Struktur, zu einem lebenswichtigen Be-

1.2 | HTML5-Dokumente

standteil des Dokuments. Diese Struktur sorgt für Form, Gliederung und Flexibilität und muss so grundsolide sein wie das Fundament eines Hauses.

Wer mit HTML5 arbeiten und Websites und Anwendungen erstellen möchte, sollte zuerst um den Aufbau dieser Struktur wissen. Wenn Sie jetzt ein gutes Fundament legen, wenden Sie später die übrigen Komponenten besser an, um diese neuen Möglichkeiten voll auszuschöpfen.

Fangen wir also bei den Grundlagen an, Schritt für Schritt. In diesem ersten Kapitel erfahren Sie, wie Sie mit den neuen Elementen, die HTML5 einführt, eine Vorlage für zukünftige Projekte gestalten.

Erstellen Sie mit Ihrem bevorzugten Editor ein leeres Dokument, um im Browser alle Codestücke zu testen, die in diesem Kapitel vorgestellt werden. So können Sie sich die Tags besser merken und mit den neuen Auszeichnungen vertraut werden.

Erinnerung

Ein HTML-Dokument ist eine Textdatei. Wenn Sie keine Entwicklungssoftware besitzen, können Sie einfach Windows Notepad oder irgendeinen anderen Editor benutzen. Die Datei sollte mit der Erweiterung *.html* und einem Namen Ihrer Wahl abgespeichert werden (zum Beispiel *meincode.html*).

Tipp

Die Codebeispiele zu diesem Buch können Sie auch von der Seite www.sybex.de/zusatzmaterial herunterladen. (Auf die jeweilige Beispieldatei wird in den Listingunterschriften hingewiesen.) Auf meiner eigenen Website www.minkbooks.com veröffentliche ich außerdem von Zeit zu Zeit Updates (auf Englisch), da HTML5 wie bereits erwähnt noch in der Entwicklung begriffen ist und es deswegen Änderungen geben kann, die sich auf die Beispiele und Beschreibungen auswirken.

1.2 Die übergeordnete Struktur

HTML-Dokumente sind streng gegliedert. Jeder Teil des Dokuments wird differenziert, deklariert und in bestimmte Tags eingeschlossen. In diesem Abschnitt werden Sie sehen, wie die globale Struktur eines HTML-Dokuments aufgebaut ist und welche neuen semantischen Elemente in HTML5 integriert wurden.

1.2.1 Das <doctype>-Element

Zuerst müssen Sie festlegen, welchen Typ Ihr Dokument haben soll. Das ist in HTML5 extrem einfach:

```
<!DOCTYPE html>
```

Warnung



Diese Zeile muss die erste Zeile Ihrer Datei sein, ohne ein Leerzeichen oder eine andere Zeile davor. Das aktiviert den Standardmodus und zwingt den Browser, HTML5 zu interpretieren, wenn es möglich ist, und es anderenfalls zu ignorieren.

Sie können jetzt direkt anfangen, den Code in Ihre HTML-Datei zu schreiben und später die neuen Elemente hinzufügen, wenn sie eingeführt werden.

1.2.2 Die HTML-Struktur aufbauen mit <html>

Nachdem der Typ des Dokuments deklariert wurde, müssen Sie die HTML-Baumstruktur aufbauen. Wie immer ist <html> das Stammelement für diesen Baum. Es umschließt allen HTML-Code.

```
<!DOCTYPE html>
<html lang="de">

</html>
```

Listing 1.1 Das <html>-Element

Das Attribut `lang` im öffnenden <html>-Tag ist das einzige Attribut, das Sie in HTML5 angeben müssen. Dieses Attribut gibt an, welchen Inhalt das Dokument verwendet, in diesem Fall Deutsch.

Erinnerung



HTML erstellt Webseiten mit Hilfe von HTML-Auszeichnungen. HTML-Tags sind Schlüsselwörter und Attribute in spitzen Klammern, zum Beispiel <html lang="de">. In diesem Fall ist `html` das Schlüsselwort und `lang` das Attribut, hier mit dem Wert `de`. Die meisten HTML-Tags treten paarweise auf, nämlich als öffnendes und schließendes Tag, und der Inhalt steht dazwischen. In diesem Fall zeigt <html lang="de"> den Beginn des HTML-Codes und </html> das Ende an. Wenn Sie das öffnende und das schließende Tag vergleichen, sehen Sie, dass im schließenden Tag ein Backslash vor dem Schlüsselwort steht (zum Beispiel </html>). Aller übriger Code wird zwischen diese beiden Tags eingefügt: <html> ... </html>.

1.2 | HTML5-Dokumente

Warnung



In HTML5 können die Struktur und die Elemente, aus denen sie gebaut ist, extrem flexibel gehandhabt werden. Das `<html>`-Element kann auch ohne Attribut stehen oder sogar gänzlich ignoriert werden. Wegen der Kompatibilität und aus einigen anderen Gründen rate ich Ihnen allerdings, einige Grundregeln zu befolgen und vermittele Ihnen im Folgenden daher eine Reihe von Best Practices.

Unter dem folgenden Link finden Sie andere Sprachen, die Sie in das `lang`-Attribut einfügen können: www.w3schools.com/tags/ref_language_codes.asp.

1.2.3 Den Seitenkopf definieren mit `<head>`

Als Nächstes bauen Sie die Vorlage weiter aus. Der HTML-Code zwischen den `<html>`-Tags muss in zwei Hauptabschnitte unterteilt werden. Wie schon in früheren HTML-Versionen ist der erste Abschnitt der Head und der zweite der Body. Im nächsten Schritt erstellen Sie folglich diese beiden Code-Abschnitte mit den bereits bekannten Elementen `<head>` und `<body>`.

An den Anfang kommt natürlich der `<head>`, der wie alle Strukturelemente ein öffnendes und ein schließendes Tag hat.

```
<!DOCTYPE html>
<html lang="de">
  <head>
  </head>
</html>
```

Listing 1.2 Das `<head>`-Element

Das `<head>`-Tag selbst hat sich gegenüber den Vorgängerversionen nicht geändert und dient auch noch genau demselben Zweck. Innerhalb des Tags definieren Sie den Titel der Webseite, deklarieren die Zeichencodierung, geben allgemeine Informationen über das Dokument und binden externe Dateien mit Styles, Scripts oder Bildern ein, um die Seite zu rendern.

Abgesehen vom Titel und einigen Icons bleiben die übrigen Informationen, die zwischen den `<head>`-Tags des Dokuments stehen, in der Regel unsichtbar.

1.2.4 Den sichtbaren Inhalt einführen mit <body>

Der nächste größere Abschnitt, der zur Hauptgliederung eines HTML-Dokuments gehört, ist der Body. Er ist der sichtbare Teil des Dokuments und steht im <body>-Tag, das sich ebenfalls gegenüber früheren HTML-Versionen nicht geändert hat:

```
<!DOCTYPE html>
<html lang="de">
  <head>
  </head>
  <body>
  </body>
</html>
```

Listing 1.3 Das <body>-Element

Erinnerung



Das Beispieldokument ist bislang noch sehr simpel, hat aber schon eine komplexe Struktur. Das liegt daran, dass HTML-Code keine sequenzielle Anweisungsfolge ist, sondern eine Auszeichnungssprache, bestehend aus Tags oder Elementen, die für gewöhnlich paarweise auftreten und geschachtelt (in etwas Anderes eingefügt) werden können. In der ersten Codezeile in Listing 1.3 steht ein einfaches Tag mit der Definition des Dokuments direkt neben dem öffnenden Tag <html lang="de">. Dieses und das schließende Tag </html> ganz unten markieren den Anfang und das Ende des HTML-Codes. Zwischen den <html>-Tags fügen Sie andere Tags ein, die zwei wichtige Teile dieser Grundstruktur spezifizieren: den <head> und den <body>. Auch diese beiden treten paarweise auf. Später werden Sie noch sehen, dass zwischen <head> und <body> auch noch andere Tags eingefügt werden. Diese Struktur gleicht einem Baum, dessen Wurzel das <html>-Tag ist.

1.2.5 Den Text definieren mit <meta>

Nun ist es Zeit, den Head des Dokuments zu erstellen. Darin gibt es einige Änderungen und Neuerungen, darunter das Tag, das die Zeichencodierung des Dokuments definiert. Dieses ist ein Meta-Tag, das anzeigt, wie der Text auf dem Bildschirm angezeigt werden soll.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
```

1.2 | HTML5-Dokumente

```
</head>  
<body>  
  
</body>  
</html>
```

Listing 1.4 Das <meta>-Element

Wie in den meisten Fällen in HTML5 besteht die Innovation für dieses Element in einer Vereinfachung. Das neue Meta-Tag für die Zeichencodierung ist kürzer und einfacher. Natürlich können Sie für `utf-8` die Codierung einsetzen, die Ihnen lieber ist, und andere Meta-Tags wie `description` oder `keywords` hinzufügen, wie im nächsten Beispiel gezeigt wird:

```
<!DOCTYPE html>  
<html lang="de">  
<head>  
  <meta charset="utf-8">  
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">  
  <meta name="keywords" content="HTML5, CSS3, Javascript">  
</head>  
<body>  
  
</body>  
</html>
```

Listing 1.5 Weitere <meta>-Elemente hinzufügen

Erinnerung



In einem Dokument können mehrere Meta-Tags zur Angabe allgemeiner Informationen eingesetzt werden, doch diese Informationen werden nicht im Browserfenster angezeigt, sondern sind nur für Suchmaschinen und Geräte wichtig, die eine Vorschau oder Zusammenfassung der relevanten Daten unseres Dokuments benötigen. Wie schon erwähnt: Außer dem Titel und einigen Symbolen bleiben die meisten Informationen zwischen den `<head>`-Tags für die Nutzer unsichtbar. In dem Code von Listing 1.5 gibt der Attributname innerhalb des `<meta>`-Tags seinen Typ an und `content` deklariert seinen Wert, aber keiner dieser Werte taucht auf dem Bildschirm auf.

In HTML5 ist es nicht nötig, Tags mit einem Schrägstrich am Ende selbst zu schließen, aber ich empfehle Ihnen sehr, dies aus Gründen der Kompatibilität zu tun. Das Ende des Codes können Sie wie folgt schreiben:

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8" />
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel" />
  <meta name="keywords" content="HTML5, CSS3, JavaScript" />
</head>
<body>

</body>
</html>
```

Listing 1.6 Selbstschließende Tags

1.2.6 Den Titel angeben mit <title>

Das <title>-Tag gibt wie üblich einfach nur den Titel des Dokuments an. Dazu gibt es nichts Neues zu sagen.

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Dieser Text ist der Dokumenttitel</title>
</head>
<body>

</body>
</html>
```

Listing 1.7 Das <title>-Element

Erinnerung



Der Text zwischen den <title>-Tags ist der Titel des gesamten Dokuments; er wird von Browsern in der Regel in der Titelleiste angezeigt.

1.2.7 Externe Dateien einbinden mit <link>

<link> ist ein weiteres wichtiges Element im Head des Dokuments. Es dient der Einbindung von Styles, Scripts, Bildern oder Symbolen von externen Dateien in einem Dokument. Eine der häufigsten Anwendungen von <link> besteht darin, durch Einfügen einer externen CSS-Datei Styles einzubinden:

1.2 | HTML5-Dokumente

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Dieser Text ist der Dokumenttitel</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>

</body>
</html>
```

Listing 1.8 Das <link>-Element

Da Sie in HTML5 nicht mehr angeben müssen, welche Art von Stylesheet Sie einfügen, wurde das `type`-Attribut abgeschafft. Sie benötigen lediglich zwei Attribute, um eine Styles-Datei einzubinden, nämlich `rel` und `href`. Das Attribut `rel` steht für *relation* und bezieht sich auf die Relation zwischen dem Dokument der Datei, die Sie einbinden. Hier hat das Attribut `rel` den Wert `stylesheet`, der dem Browser mitteilt, dass die Datei `mystyles.css` eine CSS-Datei mit Styles ist, die für das Rendering der Seite benötigt werden. (Mehr zu CSS-Styles erfahren Sie im nächsten Kapitel.)

Erinnerung ×

Ein Stylesheet ist eine Gruppe von Formatierungsregeln, die helfen, das Aussehen des Dokuments zu ändern, zum Beispiel die Größe und Farbe des Textes. Ohne diese Regeln würden der Text und jedes andere Element mit den Standard-Styles der Browser auf dem Bildschirm angezeigt (also den Standardgrößen, -farben usw.). Styles sind ganz einfach Regeln, die normalerweise nur wenige Codezeilen benötigen, die in demselben Dokument deklariert werden können. Später werden Sie noch sehen, dass es nicht unbedingt notwendig ist, sich diese Informationen von externen Dateien zu holen, aber ich empfehle es Ihnen. Wenn Sie die CSS-Regeln aus einem externen Dokument (einer anderen Datei) laden, können Sie das Hauptdokument gliedern, die Website lädt schneller und Sie profitieren von den neuen Funktionen von HTML5.

Mit der letzten Einfügung können Sie die Arbeit am Head der Vorlage als fertig betrachten. Nun können Sie am Body arbeiten, in dem die eigentliche Action stattfindet.

1.3 Die Struktur des Bodys

Die Body-Struktur (der Code zwischen den `<body>`-Tags) generiert wie bisher den sichtbaren Teil Ihres Dokuments, also die Webseite.

In HTML konnte man die Informationen im Body eines Dokuments schon immer auf unterschiedliche Weise erstellen und gliedern. Eines der ersten Elemente, die für diesen Zweck angeboten wurden, war `<table>`. In Tabellen konnte der Autor Daten, Text, Bilder und Tools in Zeilen und Spalten von Zellen anordnen, auch wenn sie ursprünglich nicht für diesen Zweck geschaffen waren.

In der Frühzeit des Web waren Tabellen eine Revolution; sie verbesserten die visuelle Darstellung des Dokuments und das Erlebnis des Benutzers massiv. Später wurden die Funktionen der Tabelle allmählich durch andere Elemente ersetzt, die dasselbe leisteten, aber mit weniger Code und schneller, wodurch die Erstellung, Portierbarkeit und Wartung einfacher wurde.

Das `<div>`-Element begann, das Feld zu beherrschen. Mit der Entstehung interaktiver Webanwendungen und der Integration von HTML, CSS und JavaScript bürgerte sich das `<div>`-Tag überall ein. Doch ebenso wie `<table>` sagt auch das `<div>`-Element nicht viel über die Teile des Bodys aus, die es repräsentiert. Zwischen das öffnende und das schließende `<div>`-Tag können Sie alles Mögliche setzen: Menüs, Text, Links, Scripts, Formulare usw. Mit anderen Worten spezifiziert das Schlüsselwort `div` lediglich einen Bereich (englisch »division«) im Body, ähnlich wie eine Zelle in einer Tabelle, aber es gibt keinen Hinweis darauf, um welche Art von Bereich es sich handelt, was sein Zweck ist oder was er enthält.

Für die User sind diese Hinweise nicht wichtig, aber für die Browser ist eine richtige Interpretation dessen, was sich in dem zu verarbeitenden Dokument befindet, von zentraler Bedeutung. Seit es mobile Geräte gibt und die Menschen auf viele verschiedene Arten online gehen können, ist es wichtiger denn je, dass jeder Teil des Dokuments identifiziert werden kann.

Daher enthält HTML5 neue Elemente, die dabei helfen, jeden Teil des Dokuments zu identifizieren und den Body zu gliedern. In HTML5 sind die wichtigsten Teile eines Dokuments differenziert und die Hauptstruktur hängt nicht mehr alleine von `<div>`- oder `<table>`-Tags ab.

Es liegt an Ihnen, wie Sie diese neuen Elemente nutzen, aber die Keywords, die für jedes von ihnen ausgewählt wurden, geben Ihnen einen Hinweis auf ihre Funktion. Normalerweise ist eine Webseite oder Webanwendung in mehrere visuelle

1.3 | HTML5-Dokumente

Bereiche aufgeteilt, um das Nutzererlebnis und die Interaktivität zu verbessern. Sie werden bald sehen, dass die Keywords für die neuen HTML5-Elemente mit diesen visuellen Arealen aufs Engste verbunden sind.

1.3.1 Die Gliederung

Abbildung 1.1 zeigt ein Layout, das zurzeit in den meisten Websites vorkommt. Obwohl jeder Designer hier seine eigenen Designs erschafft, können Sie im Allgemeinen in jeder Website, die Sie studieren, die folgenden Abschnitte erkennen.

Oben, im Header, stehen normalerweise das Logo, der Name, der Untertitel und Kurzbeschreibungen Ihrer Website oder Webseite. Darunter erkennen Sie die Navigationsleiste, in der fast jeder Entwickler ein Menü oder eine Link-Liste zu Navigationszwecken anbietet. Von dieser Leiste aus wird der Nutzer zu verschiedenen Seiten oder Dokumenten, zumeist in derselben Website, geleitet.

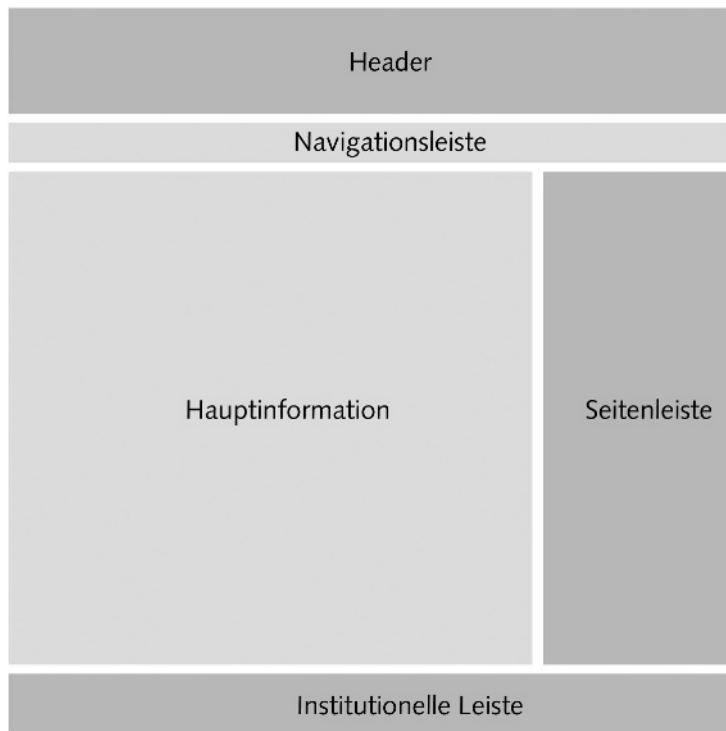


Abbildung 1.1 Visuelle Darstellung eines typischen Webseitenlayouts

Der wichtigste Inhalt der Seite steht normalerweise in der Mitte des Layouts. Dieser Abschnitt präsentiert wichtige Informationen und Links. Zumeist ist er in mehrere Zeilen und Spalten aufgeteilt. Im Beispiel in Abbildung 1.1 sehen Sie nur zwei Spalten namens Hauptinformationen und Seitenleiste, doch dieser Abschnitt ist extrem flexibel und die Designer passen ihn üblicherweise an ihre Bedürfnisse an, indem sie weitere Zeilen einfügen, die Spalten in kleinere Blöcke aufteilen oder andere Kombinationen und Distributionen erstellen. Der Inhalt hat in diesem Teil des Layouts normalerweise die höchste Priorität. Im Beispiel-Layout könnte die Hauptinformation eine Liste von Artikeln, Produktbeschreibungen, Blogbeiträgen oder andere wichtige Informationen aufführen und die Seitenleiste könnte eine Link-Liste enthalten, die auf diese einzelnen Elemente verweist. In einem Blog würde diese letzte Spalte beispielsweise eine Link-Liste zeigen, die auf die einzelnen Blogbeiträge, Informationen über den Autor und so weiter verweisen.

Im unteren Teil des typischen Layouts finden Sie die »institutionelle Leiste«. Ich nenne sie so, weil dies die Stelle im Layout ist, an der oft allgemeine Informationen

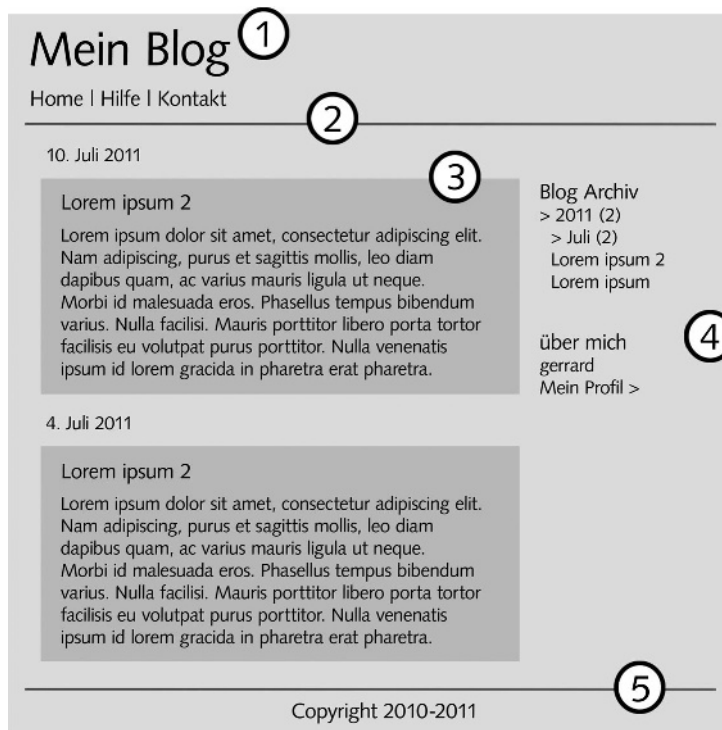


Abbildung 1.2 Visuelle Darstellung eines typischen Blog-Layouts

1.3 | HTML5-Dokumente

über die Website, den Autor oder das Unternehmen plus Regeln, AGBs, Anfahrtsskizzen und alle die anderen Daten stehen, die der Entwickler für mitteilenswert hält. Die institutionelle Leiste ist die Ergänzung zum Header und gehört heutzutage zur wesentlichen Struktur einer Webseite dazu.

Abbildung 1.2 ist ein Beispiel für einen normalen Blog. In diesem Beispiel können Sie leicht jeden Teil des soeben erläuterten Designs erkennen.

1. Header
2. Navigationsleiste
3. Hauptinformationsbereich
4. Seitenleiste
5. Fuß der Seite oder institutionelle Leiste

Diese einfache Darstellung eines Blogs kann Ihnen helfen, zu verstehen, dass jeder Abschnitt in einer Website seinen Zweck hat. Manchmal ist dieser Zweck nicht klar

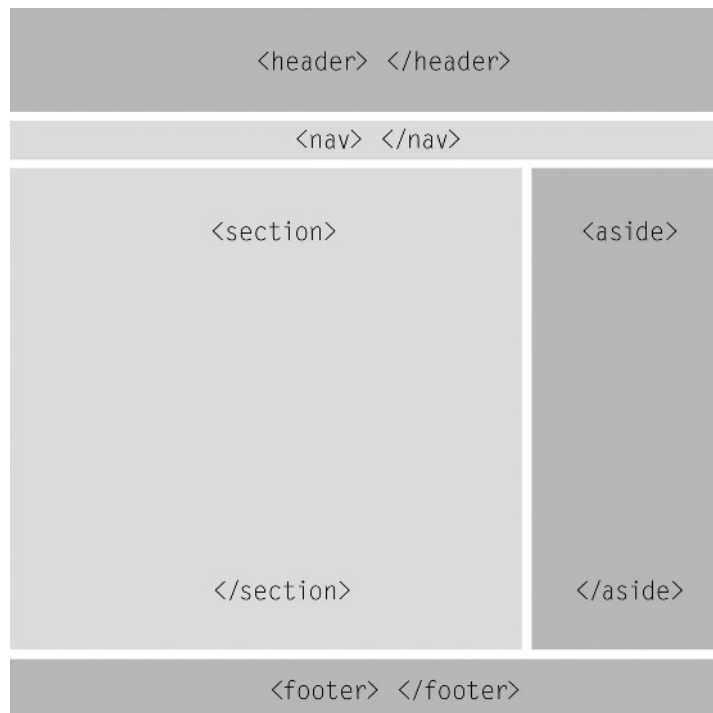


Abbildung 1.3 Darstellung der Anordnung von Abschnitten mit HTML5-Tags

ersichtlich, aber die Basis ist immer da und Sie werden überall und jederzeit alle oben beschriebenen Abschnitte erkennen können.

HTML5 berücksichtigt diese Grundstruktur und dieses Layout, stellt aber auch neue Elemente bereit, um alle zu differenzieren und zu deklarieren. Jetzt können Sie den Browsern mitteilen, wofür jeder Abschnitt da ist.

Abbildung 1.3 zeigt das typische Layout, das Sie schon zuvor benutzt haben, doch diesmal mit den entsprechenden HTML5-Elementen für jeden Abschnitt (sowohl öffnende als auch schließende Tags).

1.3.2 Abschnitte beschreiben mit `<header>`

Eines der neuen Elemente von HTML5 ist `<header>`. Verwechseln Sie `<header>` nicht mit dem `<head>`-Tag, mit dem Sie eben den Head des Dokuments angelegt hatten. Zwar ist `<header>` genau wie `<head>` für einleitende Informationen gedacht (zum Beispiel für Titel, Untertitel oder Logos), doch der Geltungsbereich der beiden Tags ist unterschiedlich. Während das `<head>`-Tag Informationen über das gesamte Dokument liefert, ist der `<header>` nur für den Body oder Einzelabschnitte in diesem Body gedacht.

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Dieser Text ist der Dokumenttitel</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <header>
    <h1>Dies ist der Haupttitel der Website</h1>
  </header>
</body>
</html>
```

Listing 1.9 Das `<header>`-Element (html01.html)

In Listing 1.9 wird der Titel der Webseite mit dem `<header>`-Tag definiert. Beachten Sie, dass dieser Header nicht dasselbe ist wie der Titel des Gesamtdokuments, der zuvor im Head definiert wurde. Das `<header>`-Element zeigt den Beginn des Bodys

1.3 | HTML5-Dokumente

und sichtbaren Teils des Dokuments an. Von hier an werden Sie das Ergebnis des Codes im Browserfenster sehen können.

Wenn Sie die Anleitungen vom Anfang dieses Kapitels nachvollzogen haben, haben Sie bereits eine Textdatei mit den bisher beschriebenen Codeteilen fertig zum Testen vorliegen. Wenn nicht, können Sie sich die Codebeispiele dieses Buches von der Seite www.sybex.de/zusatzmaterial herunterladen und die Datei *html01.html* aus dem Ordner *Kapitel_01* in einem Editor öffnen. Um das Script in Aktion zu sehen, öffnen Sie einen HTML5-kompatiblen Browser wie Chrome oder Firefox.

Erinnerung

Zwischen den `<header>`-Tags in Listing 1.10 steht `<h1>` – ein altes HTML-Element, das eine Überschrift definierte. Die Nummer gibt die Wichtigkeit der Überschrift und ihres Inhalts an. Da das Element `<h1>` die höchste und `<h6>` die niedrigste Überschriftsebene angibt, zeichnet `<h1>` den Haupttitel aus und die übrigen werden für untergeordnete Überschriften verwendet. Weiter unten werden Sie sehen, wie diese Elemente in HTML5 funktionieren.

1.3.3 Die Navigationsleiste anlegen mit `<nav>`

Der nächste Abschnitt in unserem Beispiel ist die Navigationsleiste, die in HTML5 mit dem `<nav>`-Tag angelegt wird.

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Dieser Text ist der Dokumenttitel</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <header>
    <h1>Dies ist der Haupttitel der Website</h1>
  </header>
  <nav>
    <ul>
      <li>Start</li>
      <li>Fotos</li>
```

```
<li>Videos</li>
<li>Kontakt</li>
</ul>
</nav>
</body>
</html>
```

Listing 1.10 Das `<nav>`-Element (html02.html)

Wie Sie in Listing 1.10 sehen, steht das `<nav>`-Element zwischen den `<body>`-Tags, aber hinter dem schließenden Tag des Headers (`</header>`) und nicht etwa zwischen den `<header>`-Tags. Denn `<nav>` gehört nicht zum Header, sondern zu einem neuen Abschnitt.



Abbildung 1.4 html02.html

Ich habe bereits erwähnt, dass Sie selbst entscheiden können, welche Struktur und Reihenfolge Sie in HTML5 verwenden möchten. Das bedeutet, dass HTML5 sehr vielseitig ist und Ihnen nur die Parameter und Grundelemente für Ihre Arbeit gibt, doch wie Sie diese einsetzen, entscheiden Sie selbst. Ein Beispiel für diese Vielseitigkeit ist, dass das `<nav>`-Tag auch in das `<header>`-Element oder jeden anderen Abschnitt des Bodys eingefügt werden könnte. Allerdings sollten Sie immer berücksichtigen, dass diese neuen Tags geschaffen wurden, um Browsern mehr Informationen zu geben und jedem neuen Programm und Gerät auf dem Markt zu helfen, die relevantesten Teile des Dokuments zu erkennen. Wenn Sie Ihren Code portierbar und lesbar halten möchten, sollten Sie sich an die Standards halten und

1.3 | HTML5-Dokumente

alles möglichst klar gestalten. Das `<nav>`-Element ist dazu gedacht, Navigationshilfen wie zum Beispiel das Hauptmenü oder größere Navigationsblöcke zu enthalten, und so sollte es auch benutzt werden.

Erinnerung



In dem Beispiel von Listing 1.10 führe ich die Menüoptionen für die Webseite auf. Zwischen den `<nav>`-Tags stehen zwei Elemente, die zur Erstellung einer Liste verwendet werden. Das `<ul>`-Element hat den Zweck, eine Liste zu definieren. Zwischen die `<ul>`-Tags sind mehrere `<li>`-Tags mit verschiedenem Text eingeschachtelt, die die Menüoptionen darstellen. Wenn Sie mehr Informationen zu den regulären Sprachelementen benötigen, ziehen Sie bitte zusätzlich ein Lehrbuch über die Grundkonzepte von HTML zurate.

1.3.4 Die Informationen gliedern mit `<section>`

Nun folgen in unserem Standarddesign der Hauptinformationsbereich und die Seitenleiste aus Abbildung 1.1. Wie bereits erwähnt, enthält der Hauptinformationsbereich die relevantesten Informationen des Dokuments und kann in verschiedenen Formen auftreten, zum Beispiel aufgeteilt in mehrere Blöcke oder Spalten. Da diese Spalten und Blöcke einem allgemeineren Zweck dienen, heißt das HTML5-Element, das sie spezifiziert, einfach `<section>`.

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Dieser Text ist der Dokumenttitel</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <header>
    <h1>Dies ist der Haupttitel der Website</h1>
  </header>
  <nav>
    <ul>
      <li>Start</li>
      <li>Fotos</li>
```

```
<li>Videos</li>
<li>Kontakt</li>
</ul>
</nav>
<section>
</section>
</body>
</html>
```

Listing 1.11 Das <section>-Element

Wie die Navigationsleiste ist auch der Hauptinformationsbereich ein eigener Abschnitt. Daher folgt der Abschnitt für die Hauptinformationsbereich unter dem schließenden </nav>-Tag.

Vergleichen Sie den letzten Code in Listing 1.11 mit dem Layout in Abbildung 1.3, um zu erkennen, wo die Tags im Code stehen und welchen Abschnitt diese Tags in der visuellen Darstellung der Webseite generieren.

Warnung



Die Tags, die die einzelnen Abschnitte des Dokuments repräsentieren, stehen im Code in einer Liste untereinander, aber in der Website zuweilen nebeneinander (wie zum Beispiel die Spalten der Hauptinformationsbereich und der Seitenleiste). In HTML5 wird die Bildschirmdarstellung dieser Elemente an CSS delegiert. Das Design entsteht, indem jedem Element CSS-Styles zugewiesen werden. Im nächsten Kapitel erfahren Sie mehr zu CSS.

1.3.5 Sekundärinformationen kennzeichnen mit <aside>

In einem typischen Website-Layout (Abbildung 1.1) steht neben dem Hauptinformationsbereich eine so genannte Seitenleiste. Dies ist eine Spalte oder ein Abschnitt, der üblicherweise Daten enthält, die zu den Hauptinformationen gehören, aber nicht ganz so relevant oder wichtig sind.

In dem Beispiel eines Standardlayouts für ein Blog (Abbildung 1.2) enthält die Seitenleiste eine Link-Liste (Nummer 4). In diesem Beispiel verwiesen diese Links auf die einzelnen Blogbeiträge und lieferten Zusatzinformationen über den Autor. Die Informationen innerhalb dieser Leiste stehen mit den Hauptinformationen in Zusammenhang, sind aber für sich gesehen nicht so relevant. Für das Blogbeispiel lässt sich sagen, dass die Blogbeiträge relevant sind, aber die Links und Kurzvor-

1.3 | HTML5-Dokumente

schau dieser Einträge nur als Navigationshilfe dienen und nicht das sind, wofür sich der Leser oder Nutzer am meisten interessiert.

In HTML5 können Sie solche Sekundärinformationen mit dem Element `<aside>` kenntlich machen.

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="utf-8">
  <meta name="Beschreibung" content="Dies ist ein HTML5-Beispiel">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <title>Dieser Text ist der Dokumenttitel</title>
  <link rel="stylesheet" href="mystyles.css">
</head>
<body>
  <header>
    <h1>Dies ist der Haupttitel der Website</h1>
  </header>
  <nav>
    <ul>
      <li>Start</li>
      <li>Fotos</li>
      <li>Videos</li>
      <li>Kontakt</li>
    </ul>
  </nav>
  <section>
    <aside>
      <blockquote>Artikel Nummer eins</blockquote>
      <blockquote>Artikel Nummer zwei</blockquote>
    </aside>
  </section>
</body>
</html>
```

Listing 1.12 Das `<aside>`-Element (html03.html)

Das `<aside>`-Element könnte auf unserer Beispielseite rechts oder links stehen; das Tag hat keine vordefinierte Position. Das `<aside>`-Element beschreibt lediglich die enthaltenen Informationen, aber keinen Ort in der Struktur. Das Element kann überall im Layout stehen und kann verwendet werden, solange sein Inhalt nicht als Hauptinhalt des Dokuments betrachtet wird. So können Sie das `<aside>`-Ele-

1.3 | HTML5-Dokumente

```
<aside>
  <blockquote>Artikel Nummer eins</blockquote>
  <blockquote>Artikel Nummer zwei</blockquote>
</aside>
<footer>
  Copyright &copy; 2012-2013
</footer>
</body>
</html>
```

Listing 1.13 Das <footer>-Element (html04.html)

Im Layout einer typischen Webseite würde der Abschnitt, den ich »institutionelle Leiste« getauft habe, mit <footer>-Tags definiert, weil er das Ende (den Fuß) des Dokuments darstellt. In diesem Teil der Webseite werden in der Regel allgemeine Informationen über den Autor oder über die Firma hinter dem Projekt gegeben, wie zum Beispiel Copyright, AGBs und so weiter.

Das <footer>-Element markiert in der Regel das Ende eines Dokument-Bodys und dient hauptsächlich dem oben beschriebenen Zweck. Allerdings kann das <footer>-Tag in einem Body auch mehrmals verwendet werden, um jeweils das Ende der verschiedenen Abschnitte zu markieren. (Das <header>-Tag kann ebenfalls im Body mehrfach verwendet werden.) Diese Eigenschaft lernen Sie später noch genauer kennen.



Abbildung 1.5 html04.html

1.4 In der Tiefe des Bodys

Der Body Ihres Dokuments ist jetzt bereit. Die Grundstruktur der Website ist fertig, aber an dem Inhalt müssen Sie noch arbeiten. Die HTML5-Elemente, die Sie bisher kennen gelernt haben, helfen dabei, die Layout-Abschnitte zu identifizieren und jedem einen eigenen Zweck zuzuweisen, doch das eigentlich Wichtige an einer Website ist das, was in diesen Abschnitten steht.

Die meisten der bisher betrachteten Elemente wurden angelegt, um dem HTML-Dokument eine Struktur zu geben, die von Browsern und neuen Geräten identifiziert und erkannt wird. Sie haben erfahren, dass der Body oder sichtbare Teil des Dokuments mit den `<body>`-Tags deklariert wird, während die `<header>`-Tags wichtige Informationen für den Body umschließen, die `<nav>`-Tags Navigationshilfen geben, die `<section>`-Tags den relevanten Inhalt enthalten und die `<aside>`- und `<footer>`-Tags Zusatzinformationen bereitstellen. Doch keines dieser Elemente sagt irgendetwas über den tatsächlichen Inhalt aus. Alle haben einen ganz konkreten strukturellen Zweck.

Je tiefer Sie in das Dokument hineingehen, umso näher kommen Sie der Inhaltsdefinition. Diese Informationen setzen sich aus verschiedenen visuellen Elementen zusammen, zum Beispiel aus Titel, Texten, Bildern, Videos und interaktiven Anwendungen. Sie sollten in der Lage sein, diese Elemente auseinanderzuhalten und Beziehungen zwischen ihnen festzulegen.

1.4.1 Beiträge abgrenzen mit `<article>`

Das zuvor in Abbildung 1.1 dargestellte Layout ist derzeit die gebräuchlichste und essenzielle Struktur für Websites im Internet, aber auch repräsentativ für die Weise, wie Inhalte auf dem Bildschirm gezeigt werden. Genauso wie Blogs in Beiträge unterteilt sind, präsentieren Websites relevante Informationen in der Regel in Abschnitten, die ähnliche Merkmale haben. Diese Abschnitte können Sie mit dem `<article>`-Element kennzeichnen.

```
...
<section>
  <article>
    Dies ist der Text meines ersten Beitrags
  </article>
```

1.4 | HTML5-Dokumente

```
<article>
  Dies ist der Text meines zweiten Beitrags
</article>
</section>
...
```

Listing 1.14 Das `<article>`-Element (html05.html, gekürzt)

Wie Sie am Code von Listing 1.14 sehen, stehen die `<article>`-Tags zwischen den `<section>`-Tags. Die `<article>`-Tags gehören zu diesem Abschnitt, sie sind seine Kinder, ebenso, wie jedes Element zwischen den `<body>`-Tags ein Kind dieses Bodys ist. Doch ebenso wie die Kinder des Bodys werden die `<article>`-Tags hintereinander gesetzt, weil jedes ein eigenständiger Teil der `<section>` ist, wie in Abbildung 1.6 gezeigt.

Das `<article>`-Element ist durch seinen Namen nicht etwa auf Nachrichtenmeldungen oder Artikel festgelegt, sondern wurde geschaffen, um ein unabhängiges

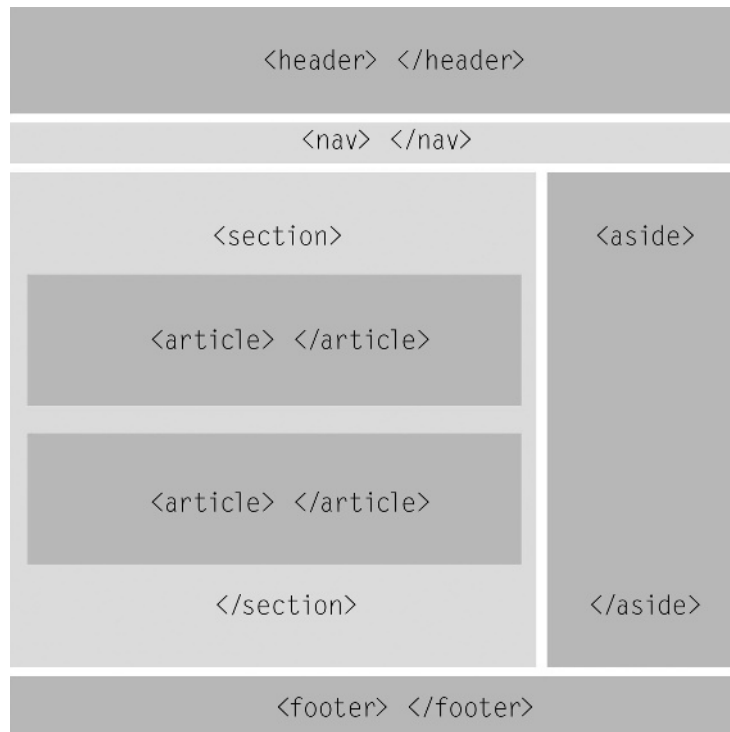


Abbildung 1.6 Visuelle Darstellung des `<article>`-Tags in dem Abschnitt, der für die relevanten Informationen der Webseite angelegt wurde

Stück Inhalt unterzubringen. Es kann also einen Forumsbeitrag, einen Zeitschriftenartikel, einen Blogbeitrag, einen Nutzerkommentar und so weiter enthalten. Dieses Element fasst verwandte Informationsteile zusammen, unabhängig davon, um welche Art von Informationen es sich handelt.

Erinnerung



Wie bereits erwähnt, kann die HTML-Struktur als Baum beschrieben werden, dessen Wurzel das `<html>`-Element ist. Man kann die Beziehungen zwischen den Elementen aber auch anhand ihrer Position in der Baumstruktur als Eltern, Kinder oder Geschwister bezeichnen. So ist in einem typischen HTML-Dokument das `<body>`-Element ein Kind des `<html>`-Elements und ein Geschwister des `<head>`-Elements. Sowohl der `<body>` als auch der `<head>` haben als Elternteil das `<html>`-Element.

Da `<article>`-Elemente ein unabhängiger Teil des Dokuments sind, hat ihr Inhalt jeweils eine eigene Struktur. Zur Strukturierung dieses Inhalts leisten uns die zuvor beschriebenen `<header>`- und `<footer>`-Tags durch ihre Vielseitigkeit gute Dienste. Diese Tags sind portierbar und können nicht nur im Body, sondern auch in jedem Abschnitt eines Dokuments eingesetzt werden.

```
...
<section>
  <article>
    <header>
      <h1>Titel von Beitrag eins</h1>
    </header>
    Dies ist der Text meines ersten Beitrags
    <footer>
      <p>Kommentare (0)</p>
    </footer>
  </article>
  <article>
    <header>
      <h1>Titel von Beitrag zwei</h1>
    </header>
    Dies ist der Text meines zweiten Beitrags
    <footer>
      <p>Kommentare (0)</p>
    </footer>
  </article>
</section>
...
```

Listing 1.15 Die Struktur des `<article>` (html06.html, gekürzt)

1.4 | HTML5-Dokumente



Abbildung 1.7 html06.html

Die beiden in Listing 1.15 eingefügten Beiträge wurden je mit einem `<article>`-Element erstellt und besitzen eine bestimmte Struktur: Zuerst kommt das `<header>`-Tag, das den mit dem `<h1>`-Element definierten Titel enthalten. Darunter steht der Inhalt selbst, also der Text des Beitrags. Und zum Schluss, hinter dem Text, folgt das `<footer>`-Tag, das die Anzahl der Kommentare angibt.

1.4.2 Überschriften gruppieren mit `<hgroup>`

In jedem `<header>`-Element, am Anfang des Bodys oder der einzelnen `<article>`, haben Sie für die Titelangaben bereits `<h1>`-Tags eingebunden. Diese Tags werden benötigt, um die Überschrift jedes Dokumentteils zu erstellen. Manchmal benötigen Sie aber auch Untertitel oder weitere Informationen, um zu erklären, worum

es in der Website oder dem Abschnitt geht. Das `<header>`-Element ist tatsächlich dafür geschaffen, auch andere Elemente aufzunehmen, wie zum Beispiel ein Inhaltsverzeichnis, Suchformulare oder kurze Texte und Logos.

Zum Anlegen der Überschriften können Sie auch die anderen `h`-Tags einsetzen: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` und `<h6>`. Doch für die interne Verarbeitung und um während der Interpretation des Dokuments nicht mehrere Abschnitte oder Unterabschnitte zu generieren, müssen diese Tags in Gruppen zusammengefasst werden. Hierzu bietet HTML5 das `<hgroup>`-Element an.

```
...
<section>
  <article>
    <header>
      <hgroup>
        <h1>Titel von Beitrag eins</h1>
        <h2>Untertitel von Beitrag eins</h2>
      </hgroup>
      <p>erschieden am 10.12.2012</p>
    </header> Dies ist der Text meines ersten Beitrags
    <footer>
      <p>Kommentare (0)</p>
    </footer>
  </article>
  <article>
    <header>
      <hgroup>
        <h1>Titel von Beitrag zwei</h1>
        <h2>Untertitel von Beitrag zwei</h2>
      </hgroup>
      <p>erschieden am 15.12.2012</p>
    </header>
    Dies ist der Text meines zweiten Beitrags
    <footer>
      <p>Kommentare (0)</p>
    </footer>
  </article>
</section>
...
```

Listing 1.16 Das `<hgroup>`-Element (html07.html, gekürzt)

`h`-Tags müssen ihre Hierarchie beibehalten, das heißt, Sie müssen zuerst den Titel mit dem `<h1>`-Tag deklarieren, dann den Untertitel mit `<h2>` und so weiter. Allerdings

1.4 | HTML5-Dokumente

können Sie in HTML5 (im Gegensatz zu den früheren HTML-Versionen) die h-Tags wiederverwenden und diese Hierarchie in jedem Abschnitt des Dokuments immer und immer wieder aufbauen. In Listing 1.16 ist jeder Beitrag mit Untertitel und Metadaten versehen und Titel und Untertitel mit `<hgroup>` zusammengefasst. Die Hierarchie mit `<h1>` und `<h2>` wird in jedem `<article>`-Element wiederverwendet.

Warnung

Das Element `<hgroup>` ist notwendig, wenn Sie in demselben `<header>` einen Titel und Untertitel oder mehrere h-Tags zusammenfassen. Da dieses Element nur h-Tags enthalten darf, haben wir die Metadaten in unserem Beispiel weggelassen. Wenn Sie nur das `<h1>`-Tag oder das `<h2>`-Tag mit Metadaten haben, müssen Sie diese Elemente nicht zusammenfassen. In dem `<header>` des Bodys haben wir zum Beispiel auf dieses Element verzichtet, weil nur ein einziges h-Element darinsteht. Vergessen Sie niemals, dass `<hgroup>`, wie sein Name schon sagt, ausschließlich dazu da ist, h-Tags zusammenzufassen.

Browser und Programme, die Websites ausführen und wiedergeben, lesen den HTML-Code und erstellen ihre eigene, interne Struktur, um jedes Element zu interpretieren und zu verarbeiten. Diese interne Struktur ist in Abschnitte unterteilt, die nichts mit den Bereichen im Design oder dem `<section>`-Element zu tun haben. Sie sind konzeptionelle Abschnitte, die während der Code-Interpretation generiert werden. Das `<header>`-Element erstellt von alleine nicht einen dieser konzeptionellen Abschnitte; das heißt, dass die Elemente innerhalb des `<header>` verschiedene Ebenen darstellen und intern verschiedene Abschnitte generieren könnten. Das `<hgroup>`-Element wurde geschaffen, um h-Tags zusammenzufassen und den Browser vor Fehlinterpretationen zu bewahren.

Erinnerung

Metadaten sind Daten, die andere Daten beschreiben und Informationen über sie liefern. In dem Beispiel sind die Datumsangaben der Veröffentlichungen Metadaten.

1.4.3 Visuelle Inhalte einbinden mit `<figure>` und `<figcaption>`

Das `<figure>`-Tag wurde erstellt, um den Inhalt des Dokuments genauer deklarieren zu können. Bevor dieses Element eingeführt wurde, konnten Inhalte, die Teil der Informationen waren, aber in sich abgeschlossen, wie zum Beispiel Illustrationen, Bilder, Videos und so weiter, nicht identifiziert werden. Diese Elemente gehören im Allgemeinen zum relevanten Inhalt, können aber nicht verschoben

werden, ohne den Fluss des Dokuments zu beeinträchtigen oder zu unterbrechen. Wenn solche Daten vorhanden sind, können sie mit `<figure>`-Tags gekennzeichnet werden.

```
...
<section>
  <article>
    <header>
      <hgroup>
        <h1>Titel von Beitrag eins</h1>
        <h2>Untertitel von Beitrag eins</h2>
      </hgroup>
      <p>erschieden am 10.12.2012</p>
    </header>
    Dies ist der Text meines ersten Beitrags
    <figure>
      
      <figcaption>
        Dies ist ein Bild des ersten Beitrags
      </figcaption>
    </figure>
    <footer>
      <p>Kommentare (0)</p>
    </footer>
  </article>
  <article>
    <header>
      <hgroup>
        <h1>Titel von Beitrag zwei</h1>
        <h2>Untertitel von Beitrag zwei</h2>
      </hgroup>
      <p>erschieden am 15.12.2012</p>
    </header> Dies ist der Text meines zweiten Beitrags
    <footer>
      <p>Kommentare (0)</p>
    </footer>
  </article>
</section>
...
```

Listing 1.17 Die Elemente `<figure>` und `<figcaption>` (html08.html, gekürzt)

In Listing 1.17 wird im ersten Beitrag direkt hinter dem Text ein Bild eingefügt (``). Das ist so üblich, häufig wird Text mit Bildern oder

1.4 | HTML5-Dokumente



Abbildung 1.8 html08.html

Videos angereichert. Mit den `<figure>`-Tags können Sie diese visuellen Dreingaben umschließen und von den relevantesten Informationen abheben.

In Listing 1.17 erkennen Sie auch ein zusätzliches Element in `<figure>`. Normalerweise werden Informationen wie Bilder oder Videos durch einen kurzen Text darunter beschrieben. HTML5 bietet ein Element, mit dem Sie diesen beschreibenden Text positionieren und kennzeichnen können: Die `<figcaption>`-Tags umschließen die Bildunterschrift zur `<figure>` und setzen beide Elemente und ihre Inhalte in Beziehung.

1.5 Neue und alte Elemente

HTML5 wurde entwickelt, um den Code einfacher, spezifischer und ordentlicher zu gestalten. Dazu wurden Tags und Attribute hinzugefügt und HTML mit CSS und JavaScript integriert. Diese Neuerungen und Verbesserungen gegenüber den Vorversionen erstrecken sich nicht nur auf neue Elemente, sondern auch auf den Umgang mit den alten.

1.5.1 Strings hervorheben mit `<mark>`

Das `<mark>`-Tag wurde hinzugefügt, um einen Teil eines Textes hervorzuheben, der ursprünglich nicht als wichtig angesehen wurde, aber jetzt aufgrund der aktuellen Nutzeraktivität relevant ist. Das beste Beispiel hierfür ist ein Suchergebnis. Das `<mark>`-Element hebt nämlich auch den Teil des Textes hervor, der mit dem Suchstring übereinstimmt.

```
<span>Mein <mark>Auto</mark> ist rot</span>
```

Wenn zum Beispiel jemand nach dem Wort »Auto« sucht, könnten die Ergebnisse mit dem obigen Code angezeigt werden. Der kurze Text stellt die Ergebnisse der Suche dar und die `<mark>`-Tags dazwischen umschließen den Suchtext, nämlich das Wort »Auto«. In manchen Browsern wird dieses Wort standardmäßig mit gelbem Hintergrund hervorgehoben, aber solche Styles können Sie mit CSS jederzeit durch eigene ersetzen, wie Sie in späteren Kapiteln noch sehen werden.

Früher erzielten Sie dieses Ergebnis mit einem `<b>`-Element. Dadurch, dass es jetzt `<mark>` gibt, konnten sich die Bedeutung und der Zweck dieses und anderer, verwandter Elemente jetzt ändern.

1.5 | HTML5-Dokumente

- `<em>` sollte zur Betonung verwendet werden (anstelle des `<i>`-Tags, das zuvor dafür verwendet wurde).
- `<strong>` steht für die Wichtigkeit.
- `<mark>` hebt Text hervor, der den Umständen entsprechend relevant ist.
- `<b>` sollte nur verwendet werden, wenn es für die Situation kein besser geeignetes Element gibt.

1.5.2 Das Kleingedruckte einfügen mit `<small>`

Die neue Präzision von HTML zeigt sich auch an Elementen wie `<small>`. Dieses Element war früher dazu gedacht, Text zu präsentieren, der in kleiner Schrift verfasst ist. Das Schlüsselwort bezog sich auf die Größe des Textes, unabhängig von seiner Bedeutung. In HTML5 hat das `<small>`-Element die Aufgabe, das »Kleingedruckte« darzustellen, wie zum Beispiel Copyright-Angaben, Haftungsausschlüsse und so weiter.

```
<small>Copyright &copy; 2013 Sybex</small>
```

1.5.3 Zitieren mit `<cite>`

Ein anderes Element, das im Wesen spezifischer geworden ist, ist `<cite>`. Jetzt umschließen die `<cite>`-Tags den Titel eines Werks, wie zum Beispiel eines Buches, Films, Lieds und so weiter.

```
<span>Ich mag den Film <cite>Temptations</cite></span>
```

1.5.4 URLs einfügen mit `<address>`

Das `<address>`-Element ist ein altes Element, das in ein Strukturelement verwandelt wurde. Früher brauchten Sie es nicht zu verwenden, um eine Vorlage zu erstellen. Doch in manchen Situationen könnte es perfekt passen, um die Kontaktdaten für den Inhalt eines `<article>`-Elements oder des gesamten `<body>` zu präsentieren.

Dieses Element sollte in einem `<footer>` stehen, wie im folgenden Beispiel:

```
<article>  
  <header>  
    <h1>Titel von Beitrag zwei</h1>  
  </header>
```

```
Dies ist der Text des Beitrags
<footer>
  <address>
    <a href="http://www.sybex.de">Sybex</a>
  </address>
</footer>
</article>
```

Listing 1.18 Einem `<article>` Kontaktdaten hinzufügen

1.5.5 Datum und Uhrzeit einfügen mit `<time>`

In der letzten Vorlage von Listing 1.17 haben Sie jedem `<article>` ein Datum gegeben, um anzuzeigen, wann der Beitrag veröffentlicht wurde. Sie haben dafür ein einfaches `<p>`-Element im `<header>` der Beiträge verwendet, doch HTML5 bietet auch ein spezifisches Element für genau diesen Zweck. Mit dem `<time>`-Element können Sie einen maschinenlesbaren Zeitstempel und eine für Menschen lesbare Datums- und Uhrzeitangabe deklarieren.

```
<article>
  <header>
    <h1>Titel von Beitrag zwei</h1>
    <time datetime="2012-12-10" pubdate>erschieden am 10.12.2012
  </time>
  </header>
  Dies ist der Text des Beitrags
</article>
```

Listing 1.19 Datum und Uhrzeit mit dem `<time>`-Element angeben

In Listing 1.19 wurde das in den vorigen Beispielen verwendete `<p>`-Element durch das neue `<time>`-Element ersetzt, um anzuzeigen, an welchem Tag der Beitrag erschienen ist. Das Attribut `datetime` hat den maschinenlesbaren Zeitstempel-Wert. Dessen Format kann wie in diesem Beispiel aussehen: 2012-12-10T12:10:45. Das Attribut `pubdate` besagt einfach nur, dass der Wert des Attributs `datetime` das Erscheinungsdatum des Artikels ist.

