



HTML, XHTML & CSS Der Meisterkurs

Lernen Sie HTML, XHTML & CSS
auf dem schnellsten und einfachsten Weg!

ELIZABETH CASTRO


Markt+Technik



3 Die grundlegende (X)HTML-Struktur

In diesem Kapitel geht es um die Grundlagen der (X)HTML-Elemente, mit denen Sie die Struktur Ihres Dokuments schaffen. Sie erfahren, wie Sie neue Absätze, Überschriften, Zeilenumbrüche, Kommentare usw. erstellen.

Wenn Sie mit einer klaren und konsistenten Struktur arbeiten, können Sie bei Ihrem Dokument viel einfacher mit Stylesheets arbeiten.

3.1 Erste Arbeiten für die Webseite

Beginnen Sie bei Ihrer Seite damit, über eine Erläuterung des Dokumenttyps, der sogenannten DOCTYPE-Deklaration (Abschnitt 1.7), festzulegen, mit welcher Art von HTML oder XHTML Sie arbeiten. Der DOCTYPE-Deklaration entnehmen Browser, was sie in einem Dokument zu erwarten haben, und die Validatoren wissen, wie Ihr Code bei der Syntaxprüfung zu beurteilen ist. Dann kündigen Sie den Beginn des eigentlichen Codes mit dem öffnenden html-Tag an.

Für eine Seite mit HTML 4 Transitional:

1. Tippen Sie `<!DOCTYPE HTML PUBLIC "-//W3C //DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`, um anzugeben, dass Sie in Ihrer Webseite Transitional HTML 4.01 verwenden.
2. Tippen Sie `<html>`, um mit dem eigentlichen HTML-Teil Ihres Dokuments zu beginnen.
3. Lassen Sie ein paar Zeilen leer, um darin (anhand dieses Buchs) den Rest der Seite zu erstellen.
4. Tippen Sie `</html>` ein.

Für eine Seite mit XHTML Transitional:

1. Geben Sie `<!DOCTYPE html PUBLIC "-//W3C //DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">` ein, um anzugeben, dass Sie in Ihrer Webseite Transitional XHTML verwenden.
2. Tippen Sie `<html xmlns="http://www.w3.org/1999/xhtml">`, um den XHTML-Abschnitt Ihrer Seite zu beginnen und den Namensraum (Namespace) () zu deklarieren.
3. Lassen Sie ein paar Zeilen leer, um darin (anhand dieses Buches) den Rest der Seite zu erstellen.
4. Tippen Sie `</html>` ein.

```
<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

</html>
```

Abbildung 3.1: Dies ist die DOCTYPE-Deklaration für ein Dokument mit HTML Transitional sowie die öffnenden und schließenden Tags. Das ist ein grässliches Stück Text. Ich empfehle dringend, dies von einem Dokument herauszukopieren, anstatt zu versuchen, den ganzen Kram händisch einzutippen.

HTML versus XHTML

Wenn Sie wissen wollen, ob Sie Ihre Webseiten mit HTML oder XHTML schreiben sollten, schauen Sie sich die Abschnitte E.6 und 1.6 an. Kurz gefasst lautet hier erst einmal die Antwort, dass es momentan egal ist, was Sie nehmen, solange Sie konsistent bleiben und nichts vermischen. Wenn Sie HTML schreiben, bleiben Sie bei HTML; wenn Sie sich für XHTML entscheiden, folgen Sie den XHTML-Regeln. In diesem Buch sind die Beispiele in XHTML geschrieben, weil XHTML Konsistenz und Standardkonformität verlangt. Konsistente und standardkonforme Webseiten haben größere Chancen, auf vielen verschiedenen Browsern und Plattformen korrekt und einheitlich dargestellt zu werden. Weil HTML es nicht so genau nimmt, führt das manchmal zu einer gewissen Nachlässigkeit, die dann wiederum zu fehlerhaften oder unerwarteten Darstellungen der Webseite führt, wenn diese auf verschiedenen Browsern und Plattformen dargestellt wird.

```
<!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">

</html>
```

Abbildung 3.2: Dies ist die DOCTYPE-Deklaration für ein XHTML Transitional-Dokument mit dem öffnenden html-Tag, der erforderlichen Namespace-Deklaration und dem schließenden html-Tag.

Tipps

- Erstellen Sie mit der entsprechenden DOCTYPE-Deklaration und den html-Tags eine Vorlage (ein sogenanntes Template) als Ausgangspunkt für alle Ihre Seiten.
- DOCTYPE sowie das html-Element sind in HTML optional (sogar in HTML Strict). Bei XHTML sind beide erforderlich (wobei die Deklaration des Namensraums im öffnenden html-Tag erfolgt). Beachten Sie, dass es kein xhtml-Element gibt.
- Ich habe nur gezeigt, wie man die DOCTYPE-Deklaration für HTML Transitional und XHTML Transitional schreibt. Sie finden eine Liste von häufigen DOCTYPE-Deklarationen unter <http://www.thestyleworks.de/basics/doctypes.shtml> oder bei <http://www.w3.org>. Wie man den passenden DOCTYPE findet, wird in Abschnitt 1.7 erläutert.
- Wenn man gleich zu Beginn einer Webseite den DOCTYPE mit einem URL deklariert, schalten die Browser im Allgemeinen in den Standardmodus um. Somit können Sie mit standardkonformem Code arbeiten, um die Darstellung Ihrer Webseite besser zu steuern (Abschnitt 1.7).
- Wenn Sie nicht standardkonforme HTML-Tags nehmen, ist es sinnlos, eine DOCTYPE-Deklaration anzugeben. Fassen Sie Ihre Seite dann einfach in öffnende und schließende html-Tags ein. Aktuelle Browser werden Ihre Seiten somit im Quirks-Modus darstellen (Abschnitt 1.7).
- Wenn Sie die entsprechende DOCTYPE-Deklaration angeben, wissen die Validatoren, anhand welcher Spezifikation Ihr Code zu prüfen ist (Abschnitt 22.6).
- Beachten Sie, dass das Wort DOCTYPE (weil es ursprünglich aus einer anderen Sprache namens SGML stammt) sowohl in HTML als auch in XHTML in Großbuchstaben zu schreiben ist.
- XHTML-Seiten sollten, weil sie in XML geschrieben sind, technisch gesehen mit einer XML-Deklaration wie `<?xml version="1.0" encoding="ISO-8859-1"?>` beginnen. Allerdings schaltet eine solche Deklaration den Internet Explorer 6 für Windows unerklärlicherweise in den Quirks-Modus (Abschnitt 1.7). Das ist ein Riesen-Bug. Weil die meisten XHTML-Dateien zum Glück nicht als XML bereitgestellt oder behandelt werden, kann und sollte die Deklaration (und die Kodierung, wie sie in Abschnitt 3.3 deklariert wird) weggelassen werden.

3.2 Die Grundlagen

Die meisten Webseiten teilen sich in zwei Bereiche: den Kopf (Head) und den Körper (Body). Im Kopfabschnitt definieren Sie den Namen Ihrer Seite und ergänzen Informationen für Suchmaschinen wie Google, geben den Standort der Seite an, bauen Stilvorlagen ein und schreiben Skripte. Der Besucher sieht außer dem Titel (Abschnitt 3.4) erst einmal nichts vom Inhalt des Kopfabschnitts.

Den Header erstellen

1. Direkt nach dem öffnenden `html`-Tag (Abschnitt 3.1) tippen Sie `<head>`.
2. Lassen Sie ein paar Zeilen für die Inhalte der Kopfdaten frei.
3. Tippen Sie `</head>` ein.

Der Body Ihres (X)HTML-Dokuments umfasst die Inhalte Ihrer Webseite, also den Teil, den Ihre Besucher sehen werden, z.B. Text und Bilder.

Den Body erstellen

1. Nach dem letzten `</head>`-Tag tippen Sie `<body>`.
2. Lassen Sie etwas Platz für die Inhalte Ihrer Webseite (die Sie anhand dieses Buchs im weiteren Verlauf erstellen werden).
3. Tippen Sie `</body>` ein.

Tipps

- Die `head`- und `body`-Tags müssen in XHTML geschrieben werden. In HTML sind sie optional, doch auch wenn Sie sie gar nicht eingegeben haben, verhält sich der Browser so, als wären sie vorhanden, und Sie können sie sogar mit Stilvorlagen beeinflussen.
- Ein anderer Grund für die Verwendung von `head`- und `body`-Tags ist, dass man darüber steuern kann, wann ein bestimmtes Skript gestartet wird (Abschnitt 19.1).

```
<!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">
  <head>
  </head>
  <body>
  </body>
</html>
```

Abbildung 3.3: Anhand der Elemente `head` und `body` können Sie Ihre (X)HTML-Dokument strukturieren.

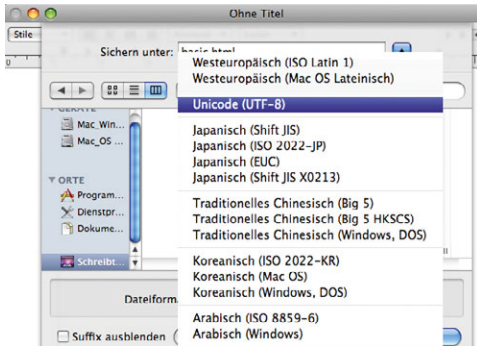


Abbildung 3.4: Ich habe meine Dateien in Unicode mit der UTF-8-Kodierung gespeichert. (Dies ist BBEdit. Weitere Details über die Speicherung von Dateien mit anderen als der Standardkodierung für Ihr System finden Sie in Kapitel 21.)

```
<!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">
<head>
<meta http-equiv="content-type" content="text/
html; charset=utf-8" />

</head>
<body>

</body>
</html>
```

Abbildung 3.5: Wenn der Browser des Surfers dieses meta-Tag sieht, weiß er, dass die Seite mit UTF-8 kodiert wurde, und stellt sie korrekt dar. Wesentlich dabei ist, dass die im meta-Tag deklarierte Kodierung zu der passt, mit der Sie die Datei auch wirklich gespeichert haben.

3.3 Deklaration der Kodierung

Alle Textdokumente, also auch (X)HTML-Dateien, werden mit einer Zeichenkodierung abgespeichert. Weil es so viele Kodierungen auf der Welt gibt, ist es eine gute Idee, wenn Sie direkt im (X)HTML-Code deklarieren, mit welcher Kodierung Ihre Seite abgespeichert wurde. Somit können Browser auf Systemen mit anderen Standardkodierungen die Zeichen auf Ihrer Seite leichter korrekt darstellen.

Die Zeichenkodierung deklarieren

- Im Header Ihrer Seite tippen Sie `<meta http-equiv="content-type" content="text/html; charset=encoding" />` ein, wobei `encoding` für die Zeichenkodierung steht, mit der Sie die Datei abgespeichert haben.
- Die Zeichenkodierung Ihrer Webseite hängt davon ab, wie Sie die Seite abgespeichert haben. Wenn Sie sie in einem reinen Textformat abgespeichert (und keine spezielle Kodierung gewählt) haben, können Sie mit großer Sicherheit davon ausgehen, dass Ihr Dokument mit der Standardkodierung für Ihre Sprache gespeichert wurde. Die Standardkodierung für ein deutsches Windows-System ist beispielsweise `windows-1252` und für einen deutschen Macintosh `x-mac-roman`.

Tipps

- Zwar sind Kodierungen durch die Spezifikation streng genommen technisch nicht erforderlich, doch rate ich dringend dazu, eine Kodierung zu deklarieren.
- Wenn Sie beim Speichern der Datei eine bestimmte Kodierung wählen, ist das die Kodierung, die Sie auch im meta-Tag verwenden sollten.
- Sie finden eine Liste der häufigsten Zeichensatzcodes unter <http://www.w3.org/International/O-charset-lang.html>.
- Bei XHTML müssen Sie die Kodierung deklarieren, wenn es eine andere als der Standard UTF-8 oder UTF-16 ist.
- Weitere Details finden Sie in Kapitel 21.

3.4 Den Seitentitel erstellen

Jede (X)HTML-Seite muss ein `title`-Element für den Seitennamen aufweisen. Ein Seitentitel soll kurz und aussagekräftig sein. In den meisten Browsern erscheint der Name der Seite in der Titelleiste des Fensters (Abbildung 3.7). Vielleicht noch wichtiger ist, dass der Titel von Suchmaschinen wie Yahoo! und Google herangezogen wird und außerdem in den Chronik- bzw. Verlaufslisten und Bookmarks (Lesezeichen) der Browser erscheint.

Einen Titel erstellen

1. Platzieren Sie den Cursor zwischen den öffnenden und schließenden `head`-Tags (Abschnitt 3.2).
2. Tippen Sie `<title>` ein.
3. Geben Sie den Namen Ihrer Webseite ein.
4. Tippen Sie `</title>` ein.

Tipps

- Das `title`-Element muss vorhanden sein.
- Ein Titel darf keine Formatierungen, Bilder oder Links auf andere Seiten enthalten.
- Der Seitentitel wirkt sich bei vielen Suchmaschinen direkt auf das Ranking aus. Je mehr der Titel den exakten Worten gleichkommt, die ein potenzieller Besucher eintippt (ohne irgendwelche zusätzlichen Worte), desto höher wird die Seite in den Suchergebnissen erscheinen. Mit dem Titel wird Ihre Seite auch in den Suchergebnissen identifiziert (Abbildung 3.8).
- Der Seitentitel wird auch in den Listen des Browserverlaufs bzw. der Chronik oder den Favoriten-, Bookmarks- und Lesezeichenmenüs verwendet, um Ihre Seite zu identifizieren (Abbildung 3.9).
- Wenn der Seitentitel Sonderzeichen wie Akzente oder bestimmte Symbole enthält, müssen diese entweder wie die Seite kodiert werden (Abschnitt 3.3) oder Sie müssen sie mit Referenzen schreiben (Abschnitt 21.6).

```
<head>
<meta http-equiv="content-type"
content="text/html; charset=utf-8" />
<title>Antoni Gaudí - Einführung</title>
</head>
<body>
```

Abbildung 3.6: Das `title`-Element sollte im Header-Bereich platziert werden und ist unverzichtbar.

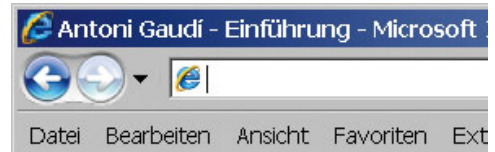


Abbildung 3.7: Der Name einer Webseite wird in der Titelleiste des Fensters angezeigt.



Abbildung 3.8: Vielleicht am wichtigsten ist, dass Ihre Seite in den Suchresultaten von Google und anderen Suchmaschinen anhand des Titels beschrieben wird. Überdies ist der Titel einer der wahrscheinlich wichtigsten Faktoren, um die Relevanz und Rangfolge einer Seite bei Suchergebnissen zu bestimmen.



Abbildung 3.9: Der Seitenname erscheint auch im Browser-Verlauf (hier gezeigt) sowie in den Favoriten- bzw. Bookmarks-Listen.

```

<head>
<meta http-equiv="content-type"
content="text/html; charset=utf-8" />
<title>Antoni Gaudí - Einführung</title>
</head>
<body>

<h1>Antoni Gaudí</h1>

<h2>La Casa Milà</h2>

<h2>La Sagrada Família</h2>

</body>

```

Abbildung 3.10: Sie können anhand von Überschriften Ihr Dokument in der Art einer Gliederung strukturieren.



Abbildung 3.11: Die übliche Standarddarstellung für Überschriften erster Ordnung ist 24 Pixel, Times New Roman in fett.

3.5 Abschnittsüberschriften erstellen

Bei (X)HTML können Sie für Ihre Webseite bis zu sechs Überschriftsebenen vergeben, um die Seite zu unterteilen.

Eine Webseite mit Überschriften strukturieren

1. Im Body-Abschnitt des (X)HTML-Dokuments tippen Sie `<hn>` ein, wobei abhängig von der Überschriftsebene, die Sie erstellen wollen, *n* für eine Zahl zwischen 1 und 6 steht.
2. Geben Sie den Inhalt der Überschrift ein.
3. Tippen Sie `</hn>` ein, wobei *n* die gleiche Zahl ist wie in Schritt 1.

Tipps

- Stellen Sie sich die Überschriften als hierarchische Unterteilungen vor. Verwenden Sie sie konsistent.
- Die einzige offizielle Regel zum Thema Überschriften lautet, dass sie umso auffälliger dargestellt werden sollen, je wichtiger sie sind (und je kleiner somit ihre Zahl ist). Nichtsdestotrotz werden sie aktuell von allen wichtigen Browsern gleich dargestellt: mit Times New Roman, in Fett und mit 24, 18, 14, 12, 10 bzw. 8 Pixel (9 Pixel auf dem Mac).
- Sie können die Überschriften z.B. mit einer bestimmten Schriftart, Schriftgröße bzw. Schriftfarbe formatieren. Weitere Details dazu finden Sie in Kapitel 10.
- Ergänzen Sie die Überschriften mit einem Verweis (*anchor* oder *id*), damit Sie Links direkt mit dieser Überschrift verknüpfen können (Abschnitt 6.2).
- Auf Wunsch können Sie den Text in der Überschrift auch ausrichten, indem Sie im öffnenden Tag `align="direction"` eintippen, wobei die Ausrichtung entweder `left`, `right`, `center` oder `justify` ist (links, rechts, zentriert oder Blocksatz). Doch sollten Sie berücksichtigen, dass das Attribut `align` als veraltet gilt und stattdessen Stylesheets bevorzugt werden sollten (Abschnitt 10.3).

3.6 Neuen Absatz beginnen

(X)HTML erkennt keine Absatzendezeichen oder Leerzeichen, die Sie in Ihren Texteditor eintippen (Abschnitt 1.2). Um einen neuen Absatz auf der Webseite zu beginnen, nehmen Sie das p-Tag.

Einen neuen Absatz beginnen

1. Tippen Sie `<p>` ein.
2. Geben Sie den Inhalt des neuen Absatzes ein.
3. Tippen Sie `</p>`, um den Absatz zu beenden.

Tipps

- Das schließende `</p>`-Tag ist sowohl in XHTML erforderlich als auch dann, wenn ein Absatz anhand von Stylesheets formatiert wird. Von daher empfehle ich, einen Absatz immer mit `</p>` zu beenden. In HTML ist das schließende `</p>` optional.
- Sie können die Überschriften z.B. mit einer bestimmten Schriftart, Schriftgröße bzw. Schriftfarbe formatieren. Weitere Details zu diesem Thema finden Sie in Kapitel 10.
- Wie der Zeilenabstand geregelt wird, erfahren Sie in Abschnitt 10.6. Um die Abstände zwischen den Absätzen anzugeben, schauen Sie in den Abschnitten 11.5 oder 11.6 nach.
- Um zwischen den Absätzen schnell (und zulässig) Leerraum zu schaffen, tippen Sie zwischen die zusätzlichen p-Elemente ` ` (*non-breaking space*, geschütztes Leerzeichen) ein. Doch auf jeden Fall ist es besser, mit CSS zu arbeiten (Abschnitt 11.5 bis 11.6).
- Sie können den Text im Absatz ausrichten, indem Sie im öffnenden p-Tag `align="direction"` eintippen, wobei die Ausrichtung entweder `left`, `right`, `center` oder `justify` ist (links, rechts, zentriert oder Blocksatz). Doch sollten Sie berücksichtigen, dass das Attribut `align` als veraltet gilt und Stylesheets verwendet werden sollten (Abschnitt 10.3).

```
</head>
<body>
<h1>Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona, um sich die unbeschreibliche Architektur von Antoni Gaudí anzuschauen.</p>
<p>Im Jahre 2002 feierte Barcelona den 150. Geburtstag von Gaudí.</p>
<h2>La Casa Milà</h2>
<p>Das Werk von Gaudí ist ausgesprochen nützlich. La Casa Milà ist ein Apartmenthaus, in dem echte Menschen leben.</p>
<h2>La Sagrada Familia</h2>
<p>Die unvollendete Kirche mit dem eigenartigen Namen Sühnekirche der Heiligen Familie ist das am meisten besuchte Gebäude in Barcelona.</p>
</body>
```

Abbildung 3.12: Beginnen und beenden Sie jeden Absatz mit öffnenden und schließenden p-Tags. Wenn Sie die Tags nicht schließen (was in HTML völlig okay, aber in XHTML verkehrt ist), werden die Stylesheets nicht korrekt angewendet.



Abbildung 3.13: Der Abstand zwischen den Absätzen ist proportional zur Textgröße.

```

</head>
<body>
<h1 id="gaudi">Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona, um sich die unbeschreibliche Architektur von Antoni Gaudí anzuschauen.</p>
<p>Im Jahre 2002 feierte Barcelona den 150. Geburtstag von Gaudí.</p>
<h2 class="building">La Casa Milà</h2>
<p>Das Werk von Gaudí ist ausgesprochen nützlich. La Casa Milà ist ein Apartmenthaus, in dem echte Menschen leben.</p>
<h2 class="building">La Sagrada Família</h2>
<p>Die unvollendete Kirche mit dem eigenartigen Namen Sühnekirche der Heiligen Familie ist das am meisten besuchte Gebäude in Barcelona.</p>
</body>

```

Abbildung 3.14: Vergeben Sie das Attribut *id* an ein eindeutiges Element, um es später formatieren oder darauf verlinken zu können. Geben Sie einer Gruppe von Elementen das Attribut *class*, um alle Elemente in dieser Gruppe auf einen Rutsch formatieren zu können.



Abbildung 3.15: Die Attribute *id* und *class* verändern für sich genommen nicht das Erscheinungsbild eines Elements. Sie müssen mit einer CSS-Formatierung kombiniert werden, um voll zum Einsatz zu kommen (siehe Kapitel 10 und 11).

3.7 Elemente benennen

Sie können Ihren (X)HTML-Elementen entweder einen eindeutigen Namen geben oder einen, mit dem sie als zu einer bestimmten Klasse gehörig identifizierbar sind. Dann können Sie auf alle Elemente mit einem bestimmten Namen Stylesheets anwenden.

Elemente eindeutig benennen

Im öffnenden Tag des Elements tippen Sie `id="name"`, wobei das Element über `name` seinen eindeutigen Namen bekommt.

Elementgruppen benennen

Im öffnenden Tag des Elements tippen Sie `class="name"`, wobei `name` der Name der Klasse ist.

Tipps

- Jede `id` in einem (X)HTML-Dokument muss eindeutig sein. Anders gesagt dürfen nicht zwei Elemente mit der gleichen `id` bezeichnet werden.
- Mehrere Elemente können zu einer Klasse gehören und dann mit dem gleichen Klassennamen bezeichnet sein.
- Wie man Stylesheets auf ein Element mit einer bestimmten `id` oder Klasse anwendet, erfahren Sie in Abschnitt 9.3.
- Die Attribute `class` und `id` können bei den meisten (X)HTML-Elementen hinzugefügt werden, sind aber besonders bei den Elementen `div` und `span` praktisch (Abschnitte 3.8 und 3.9).
- Das Attribut `id` verwandelt Elemente automatisch in einen Anker (*anchor*), mit dem Sie eine direkte Verknüpfung (Link) vornehmen können. Weitere Details erfahren Sie in Abschnitt 6.2.
- Außerdem kann das `id`-Attribut auch dafür verwendet werden, Elemente zu identifizieren, die auf eine Skriptsprache wie JavaScript reagieren sollen.

3.8 Eine Seite in Abschnitte einteilen

Wenn Sie Ihre Seite in Abschnitte einteilen, können Sie Stylesheets für mehrere Bereiche auf einmal anwenden. Das ist vor allem dann sehr praktisch, wenn Layouts mit CSS gestaltet werden (siehe Kapitel 11).

Seite in Bereiche aufteilen

1. Am Anfang des Bereichs tippen Sie `<div>` (für *division*, Bereich) ein.
2. Auf Wunsch können Sie auch `id="name"` eingeben, wobei `name` den Bereich eindeutig identifiziert.
3. Sie können außerdem `class="name"` eintippen, wobei `name` der Name der zu dem Bereich gehörenden Klasse ist.
4. Schließen Sie das öffnende `div`-Tag dann mit `>` ab.
5. Geben Sie den Inhalt des Bereichs ein.
6. Am Ende des Bereichs tippen Sie `</div>` ein.

Tipps

- Ein `div` ist ein Block-Element. Das bedeutet, dass dessen Inhalt automatisch in einer neuen Zeile beginnen.
- Tatsächlich sind die Zeilenumbrüche die einzige Formatierung, die einem Bereich eigen sind. Wenden Sie zusätzliche Formatierungen an, indem Sie der `class` oder `id` des Bereichs weitere Stylesheets zuweisen (wird in den Kapiteln 7 bis 14 beschrieben).
- Es ist nicht erforderlich, alle Bereiche mit `class` oder `id` zu kennzeichnen, obwohl Sie dadurch eine Menge Vorteile bekommen.
- Sie können einem `div`-Element sowohl ein `class`- als auch ein `id`-Attribut zuweisen, obwohl es wahrscheinlich üblicher ist, entweder das eine oder das andere zu nehmen. Der Hauptunterschied ist, dass `class` für eine Gruppe von Elementen und `id` zur Identifizierung von einzelnen, eindeutigen Elementen verwendet wird.

```
</head><body>
<div id="gaudi">
  <h1>Antoni Gaudí</h1>
  <p>Viele Touristen kommen extra nach Barcelona,
  um sich die unbeschreibliche Architektur von
  Antoni Gaudí anzuschauen.</p>
  <p>Im Jahre 2002 feierte Barcelona den 150.
  Geburtstag von Gaudí.</p>

  <div class="works">
    <h2>La Casa Milà</h2>
    <p>Das Werk von Gaudí ist ausgesprochen
    nützlich. La Casa Milà ist ein Apartmenthaus,
    in dem echte Menschen leben.</p>
  </div>

  <div class="works">
    <h2>La Sagrada Familia</h2>
    <p>Die unvollendete Kirche mit dem eigenartigen
    Namen Sühnekirche der Heiligen Familie ist das
    am meisten besuchte Gebäude in Barcelona. </p>
  </div>
</div>
</body></html>
```

Abbildung 3.16: Ein großer, umschließender Bereich beginnt mit der Überschrift erster Ordnung und reicht direkt bis zum schließenden `body`-Tag, und zwei innere Bereiche beinhalten die Überschriften zweiter Ordnung und deren Absätze.



Abbildung 3.17: Sie können die Auswirkungen der Einteilung in Bereiche im Allgemeinen erst dann sehen, wenn Sie Stylesheets hinzufügen (siehe Kapitel 8). Dann zahlt sich diese Struktur aus. Sie können diese Seite mit Stylesheets auf meiner Website sehen (Abschnitt E.9).

```

<body>
<div id="gaudi">
<h1>Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona,
um sich die unbeschreibliche Architektur von
Antoni Gaudí anzuschauen.</p>
<p>Im Jahre 2002 feierte Barcelona den 150.
Geburtstag von Gaudí.</p>

<div class="works">
<h2>La Casa Milà</h2>
<p>Das Werk von Gaudí ist ausgesprochen
nützlich. La Casa Milà ist ein Apartmenthaus, in
dem <span class="emph">echte Menschen</
span> leben.</p>
</div>

<div class="works">
<h2>La Sagrada Família</h2>
<p>Die unvollendete Kirche mit dem eigenartigen
Namen Sühnekirche der Heiligen Familie ist das
<span class="emph">am meisten besuchte</
span> Gebäude in Barcelona. </p>
</div>
</div>
</body></html>

```

Abbildung 3.18: Mit dem `span`-Tag wird ein bestimmter Bereich des Inline-Inhalts markiert. Dann können Sie den markierten Inhalt nach Belieben formatieren (das gehen wir in den Kapiteln 10 und 11 an).



Abbildung 3.19: Wiederum verleiht das `span`-Element Ihrem Dokument die zugrunde liegende Struktur. Sie sehen dessen Auswirkungen erst, wenn Sie Stylesheets anwenden (siehe Kapitel 8). Sie können diese Seite mit Stylesheets auf meiner Website sehen (Abschnitt E.9).

3.9 Einen Inline-Bereich erstellen

Sie können größere Abschnitte Ihrer Webseite in head- und body-Abschnitte, in Bereiche oder auch Überschriften (h1, h2 etc.) aufteilen, aber auch kleinere Bereiche oder Textstücke oder andere Inline-Elemente bezeichnen, um sie zu kennzeichnen und Stylesheets darauf anzuwenden.

Inline-Bereiche bezeichnen

1. Zu Beginn des Inline-Inhalts tippen Sie `<span` ein.
2. Auf Wunsch können Sie auch `id="name"` eingeben, wobei `name` den Bereich des `span`-Inhalts eindeutig identifiziert.
3. Sie können außerdem `class="name"` eintippen, wobei `name` der Name der Klasse ist, zu der der `span`-Bereich gehört.
4. Schließen Sie das öffnende `span`-Tag dann mit `>` ab.
5. Erstellen Sie die Inline-Inhalte, die gekennzeichnet werden sollen.
6. Am Ende des `span`-Bereichs tippen Sie `` ein.

Tipps

- Weitere Einzelheiten über die Unterschiede zwischen Block- und Inline-Inhalten finden Sie in Abschnitt 1.1.
- Ein `span` hat keine „eingebaute“ Formatierung. Es kommt dann zum Einsatz, wenn Sie Stylesheets darauf anwenden, im Allgemeinen durch seine `class` oder `id` (mehr darüber in den Kapiteln 9 bis 14).
- Sie können einem `span`-Element sowohl ein `class`- als auch ein `id`-Attribut zuweisen, obwohl es wahrscheinlich üblicher ist, entweder das eine oder das andere zu nehmen. Der Hauptunterschied ist, dass `class` für eine Gruppe von Elementen und `id` zur Identifizierung von einzelnen, eindeutigen Elementen verwendet wird.

3.10 Einen Zeilenumbruch erstellen

Browser brechen Text entsprechend der Breite des Blockelements oder des Fensters automatisch um und erstellen bei Bedarf neue Zeilen. Sie können zwar mit dem p-Tag einen neuen Absatz erstellen (Abschnitt 3.6), aber Zeilenumbrüche an beliebigen Stellen auch manuell einfügen.

Das br-Tag passt sehr gut für Gedichte oder andere kurze Textzeilen, die ohne viel Zwischenraum dicht aufeinanderfolgen sollen.

Einen Zeilenumbruch einfügen

- Tippen Sie `
` ein, wo der Zeilenumbruch erfolgen soll. Es gibt kein separates schließendes br-Tag.

Tipps

- Der schließende Slash (Schrägstrich, /) ist nur bei XHTML-Dokumenten erforderlich, um die Regel zu erfüllen, dass alle Elemente korrekt geschlossen werden (Abschnitt 1.6). Achten Sie darauf, dass sich zwischen br und Slash ein Leerzeichen befindet. Sie können den Slash in HTML-Dokumenten ganz weglassen, obwohl es auch nicht schadet, ihn einzufügen.
- Nehmen Sie weitere br-Tags, um mehr Platz zwischen Zeilen oder Absätzen zu schaffen.
- Über Stylesheets können Sie den Abstand zwischen den Zeilen in einem Absatz (Abschnitt 10.6) und zwischen den Absätzen selbst steuern (Abschnitte 11.5 und 11.6).
- Das br-Tag wurde früher gemeinsam mit dem veralteten `clear`-Attribut verwendet, um Text zu steuern, der Bilder umfließen soll (Abschnitt 5.11). Diese Aufgabe hat die CSS-Eigenschaft `clear` übernommen (Abschnitt 11.11).
- Mit der CSS-Eigenschaft `white-space` können die ursprünglichen Seitenumbrüche bewahrt werden (Abschnitt 10.12).

```
<body>
<div id="toc">Antoni Gaudí<br />La Casa
Milà<br />La Sagrada Família</div>
<div id="gaudi">
<h1>Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona,
um sich die unbeschreibliche Architektur von
Antoni Gaudí anzuschauen. </p>
<p>Im Jahre 2002 feierte Barcelona den 150.
Geburtstag von Gaudí.</p>
```

Abbildung 3.20: Ich habe oben auf der Seite ein neues div erstellt, das für das Inhaltsverzeichnis verwendet werden kann. Es kommen drei Zeilen vor (dank des br-Tags) mit dem minimalen Abstand dazwischen.

```
<body>
<div id="toc">Antoni Gaudí
<br />La Casa Milà
<br />La Sagrada Família</div>
<div id="gaudi">
<h1>Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona,
um sich die unbeschreibliche Architektur von
Antoni Gaudí anzuschauen.</p>
<p>Im Jahre 2002 feierte Barcelona den 150.
Geburtstag von Gaudí.</p>
```

Abbildung 3.21: Denken Sie daran, dass in Ihrem Code Absatzendezeichen stets ignoriert werden. Dieser Code entspricht dem, der darüber in Abbildung 3.20 gezeigt wird, obwohl er einfacher zu lesen ist.

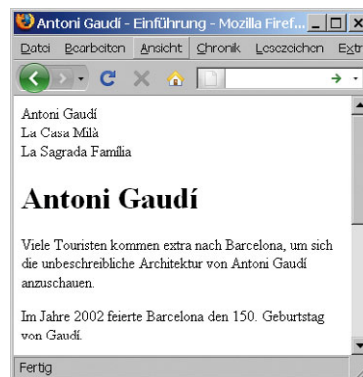


Abbildung 3.22: Wegen des br-Elements beginnen die nachfolgenden Elemente in einer neuen Zeile.

```

<body>
<!--Hier ist das Inhaltsverzeichnis, das in einem
echten Dokument viel länger sein kann.-->
<div id="toc">Antoni Gaudí<br />La Casa
Milà<br />La Sagrada Família</div>
<div id="gaudi">
<h1>Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona,
um sich die unbeschreibliche Architektur von
Antoni Gaudí anzuschauen.</p>
<p>Im Jahre 2002 feierte Barcelona den 150.
Geburtstag von Gaudí.</p>

```

Abbildung 3.23: Sie können Ihren Text mit Kommentaren versehen, um sich besser an bestimmte Dinge zu erinnern. Außerdem kann man darüber ideal Überarbeitungen nachverfolgen.



Abbildung 3.24: Kommentare sind unsichtbar (obwohl man sie lesen kann, wenn man sich den Quellcode anzeigen lässt – siehe Abschnitt 2.9).

3.11 Kommentare einfügen

Sie können in Ihren (X)HTML-Dokumenten Kommentare einbauen, damit Sie (oder zukünftige Bearbeiter) sich besser daran erinnern können, was Sie mit Ihren (X)HTML-Tags erreichen wollten. Diese Kommentare werden erst sichtbar, wenn das Dokument mit einem Texteditor geöffnet wird. Besucher sehen sie in ihrem Browser nicht.

Kommentare auf einer HTML-Seite einfügen

1. In Ihrem (X)HTML-Dokument tippen Sie dort, wo Sie einen Kommentar einfügen wollen, `<!--` ein.
2. Tippen Sie die entsprechenden Kommentare ein.
3. Mit Eingabe von `-->` schließen Sie den kommentierten Text ab.

Tipps

- Kommentare eignen sich vor allem dafür, um festzuhalten, warum man ein bestimmtes Tag verwendet hat und was man damit bezwecken wollte.
- Ein weiterer guter Grund für Kommentare ist, sich (und zukünftige Bearbeiter) daran zu erinnern, bestimmte Abschnitte aufzunehmen, zu entfernen oder zu aktualisieren.
- Sie sollten die kommentierte Seite vor der Veröffentlichung mit einem Browser anschauen, um zu vermeiden, dass Ihre (möglicherweise privaten) Kommentare ebenfalls veröffentlicht werden.
- Vorsicht allerdings vor Kommentaren, die tatsächlich allzu privat sind. Man kann sie im Browser zwar nicht gleich sehen, aber sie können ganz einfach sichtbar gemacht werden, wenn der User die Seite als (X)HTML-Quellcode speichert. Weitere Informationen über die Anzeige von Seitenquellcode finden Sie in Abschnitt 2.9.
- Kommentare dürfen nicht in anderen Kommentaren verschachtelt werden.

3.12 Elemente auf einer Webseite etikettieren

Sie können mit dem `title`-Attribut praktisch alle Bereiche Ihrer Website mit Tooltips (eine Art Sprechblase) ergänzen. Das ist besonders praktisch, um Ihren Besuchern Hinweise zu geben, was in bestimmten Formularelementen erforderlich ist, doch auch alles andere können Sie damit kennzeichnen.

Elemente auf einer Webseite etikettieren

- Im (X)HTML-Tag des zu kennzeichnenden Elements fügen Sie `title="label"` ein, wobei `label` der Text ist, der im Tooltip erscheinen soll, wenn ein Besucher mit der Maus auf das Element zeigt.

Tipp

- Der Internet Explorer für Windows macht aus dem `alt`-Attribut, das in `image`-Tags verwendet wird, ebenfalls Popup-Labels bzw. Tooltips (Abschnitt 5.6). Doch wenn in einem `image`-Tag sowohl das `title`- und auch das `alt`-Attribut vorhanden sind, wird der Tooltip aus dem Inhalt des `title`-Tags befüllt und nicht aus dem `alt`-Tag. Wenn der Internet Explorer für Windows also nicht das `alt`-Tag als Tooltip nehmen soll, verwenden Sie ein leeres `title`-Tag: `title=""`.

```
<body>
<!--Hier ist das Inhaltsverzeichnis, das in einem
echten Dokument viel länger sein kann.-->
<div id="toc" title="Inhaltsverzeichnis">Antoni
Gaudí<br />La Casa Milà<br />La Sagrada
Familia</div>
<div id="gaudi">
<h1>Antoni Gaudí</h1>
<p>Viele Touristen kommen extra nach Barcelona,
um sich die unbeschreibliche Architektur von
Antoni Gaudí anzuschauen.</p>
<p>Im Jahre 2002 feierte Barcelona den 150.
Geburtstag von Gaudí.</p>
<div class="works">
<h2>La Casa Milà</h2>
<p>Das Werk von Gaudí ist ausgesprochen
nützlich. La Casa Milà ist ein Apartmenthaus, in
dem <span class="emph">echte Menschen</
span> leben.</p>
</div>
```

Abbildung 3.25: Sie können jedem beliebigen Element einen Titel geben.



Abbildung 3.26: Wenn der Besucher auf das gekennzeichnete Element zeigt, erscheint der Titel.