

## Kapitel 3



# Das HTML-Grundgerüst

*Worin man erfährt, dass HTML zum Etikettenkleben dient und ein solides Grundgerüst das Skelett einer jeden Webseite bildet.*

Die Themen im Überblick:

- HTML – Hypertext und Etiketten kleben, Seite 66
- Das HTML-Grundgerüst ist das Skelett einer Webseite, Seite 68
- Die Dokumenttyp-Definition: `<!DOCTYPE html>`, Seite 70
- Das Stammelement `<html>`, Seite 70
- Informationen über die Webseite: `<head>`, Seite 71
- Der Inhalt der Webseite: `<body>`, Seite 74
- HTML-Elemente: Anfangs-Tag, Inhalt und Ende-Tag, Seite 74
- Know-how: Wissenswertes über Zeichensätze, Seite 75
- Auf einen Blick, Seite 79

Jede der Milliarden von Webseiten dort draußen ist in einer Sprache namens HTML geschrieben; und jeder, der Sie glauben machen möchte, dass man gute Webseiten schreiben kann, ohne zu wissen, wie HTML funktioniert, stellt eine zumindest gewagte Behauptung auf.

HTML ist einfach und wird vielleicht gerade deshalb oft nicht ernst genommen. Wenn die Kästchen in HTML aber nicht sauber gebaut sind, führt das beim Gestalten mit CSS unter Umständen zu großen Problemen.

Anders ausgedrückt: Die Gestaltung von Webseiten beginnt mit soliden HTML-Grundkenntnissen, und die lernen Sie in den folgenden Kapiteln.

### 3.1 HTML – Hypertext und Etiketten kleben

HTML steht für **HyperText Markup Language** (engl. für »Sprache zur Markierung von Hypertext«). Diese Übersetzung stimmt zwar, ist aber nicht sehr aussagekräftig, und deshalb folgt zuerst eine etwas genauere Beschreibung dieser Abkürzung.

#### HT wie Hypertext: Hyperlinks erstellen

Hypertext ist Text mit automatischen Querverweisen, die wir im Alltag *Hyperlinks* oder *Links* nennen. Das World Wide Web besteht aus Milliarden von Webseiten, die durch Hyperlinks miteinander verbunden sind. Durch diese Verbindungen entsteht ein fein gesponnenes Gewebe von Webseiten, oder etwas prosaischer ausgedrückt:

*Hyperlinks sind die Fäden, mit denen das weltweite Gewebe gesponnen wird.*

Das *HT* in *HTML* besagt schlicht und einfach, dass mit der Sprache HTML Hyperlinks erstellt werden können.

#### M wie Markup: Etiketten kleben

Markup wird häufig mit »Auszeichnung« übersetzt. Das können Sie sich wie in einem Supermarkt vorstellen:

- Waren auszeichnen bedeutet »Etiketten drankleben«.
- HTML klebt buchstäblich Etiketten in den Text einer Webseite.

Die HTML-Etiketten stehen in spitzen Klammern und heißen »Tag« (*tähg* gesprochen), was auf Deutsch »Etikett« bedeutet:

`<p>Webseiten gestalten mit HTML und CSS.</p>`

Der Browser weiß durch die Tags `<p>` und `</p>`, dass der Text dazwischen ein ganz normaler Fließtextabsatz ist. Das *p* steht dabei übrigens für *paragraph*, auf Deutsch *Absatz*.

## L wie Language: Vokabeln und Grammatikregeln

HTML ist eine Sprache, und dementsprechend gibt es Vokabeln und Grammatikregeln. Tags, Elemente, Attribute und Grammatikregeln – alles muss gelernt und zum Teil penibel umgesetzt werden. Dieses Buch enthält keine vollständige Referenz, aber die wichtigsten Regeln und HTML-Elemente werden Sie so nach und nach kennenlernen.

Links zu englischen und deutschen HTML-Referenzen mit allen Elementen und mehr Syntaxspitzfindigkeiten, als Sie wahrscheinlich jemals wissen wollten, finden Sie in Kapitel 39.1, »Referenzen und Online-Quellen zu HTML« ab Seite 787.

## Drei häufige Missverständnisse zu HTML

Drei Missverständnisse zu HTML sind so weit verbreitet, dass sie hier kurz korrigiert werden sollen:

- HTML ist **keine Programmiersprache**. HTML wird nicht programmiert, sondern schlicht und einfach *geschrieben*. HTML-Quelltext zu schreiben kommt besonders Einsteigern oft vor, als würden sie programmieren, weil es auf jedes Zeichen ankommt – aber das ist bei Diktaten ebenso, und die werden auch *geschrieben* und nicht programmiert.
- HTML ist **keine Seitenbeschreibungssprache**. HTML beschreibt nicht, wie eine Seite aussieht, HTML strukturiert sie lediglich. Gestaltet wird sie mit CSS. Seitenbeschreibungssprachen sind etwa *PostScript* oder *PDF*.
- HTML dient **nicht zum Gestalten**. »Gestalten mit HTML ist wie Malen mit Handschellen« – dieser Satz gilt heute genau wie vor zehn Jahren. Der Grund dafür ist, dass HTML nicht zum Gestalten gedacht ist. HTML erstellt nur die rechteckigen Kästchen, zum Gestalten gibt es CSS. Diese strikte Aufgabentrennung eröffnet später ungeahnte Möglichkeiten.

HTML wird also einfach nur geschrieben und dient zum Strukturieren von Text.

## 3.2 Das HTML-Grundgerüst ist das Skelett einer Webseite

Der Quelltext einer jeden Webseite lässt sich grob in vier Abschnitte aufteilen:

1. In der allerersten Zeile des Quelltextes steht der DOCTYPE.
2. Das Stammelement `html` umschließt den `head` und den `body`.
3. `head` zeigt Informationen über die Webseite und Elemente wie `title` und `meta`.
4. `body` enthält mit dem Inhalt, der im Browserfenster dargestellt wird.

Diese Einzelteile bilden zusammen das HTML-Grundgerüst, das einer Webseite wie ein Skelett eine unsichtbare Struktur gibt und sie zusammenhält.

### Eine gute Angewohnheit: `<!-- Kommentare -->`

Bevor es gleich mit dem Grundgerüst losgeht, noch ein kurzer Exkurs: Es ist eine gute Angewohnheit, den Quelltext mit *Kommentaren* zu versehen, damit Sie selbst auch nach einiger Zeit noch nachvollziehen können, was Sie sich bei bestimmten Quelltextpassagen gedacht haben.

Ein HTML-Kommentar beginnt mit `<!--` (kleiner als – Ausrufezeichen – zwei Bindestriche), hört mit `-->` (zwei Bindestriche – größer als) auf und ist im Browserfenster nicht zu sehen.

Hier ein Beispiel:

```
<!-- Dieser Text muss noch überarbeitet werden. -->
```

Wenn der Browser die Zeichenfolge `<!--` sieht, weiß er, dass der Text bis zur Zeichenfolge `-->` ein Kommentar ist und nicht im Browserfenster dargestellt werden soll.

### Tipp

#### Kommentare bleiben im Quelltext sichtbar

Denken Sie beim Verfassen der Kommentare daran, dass sie zwar nicht im *Browserfenster* erscheinen, aber doch im *Quelltext* stehen. Dort kann jeder Besucher sie lesen, wenn er will.

## Ein Grundgerüst für eine Webseite erstellen

Im folgenden ToDo erstellen Sie ein einfaches Grundgerüst und speichern es in einer Datei, die Sie als Vorlage für neue Webseiten benutzen können.

### ToDo: Ein HTML-Grundgerüst erstellen

1. Starten Sie Ihren Lieblingseditor, und erstellen Sie eine leere Datei.
2. Speichern Sie die Datei unter dem Namen *vorlage\_grundgeruest.html* in Ihrem Übungsordner.
3. Erstellen Sie das folgende HTML-Grundgerüst. Die Kommentare dienen nur zur Erläuterung und müssen nicht mit abgetippt werden:

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

<head> <!-- Im Head stehen Informationen über die Webseite -->
  <meta charset="utf-8">
  <title>HTML-Grundgerüst - Vorlage</title>
  <meta name="description" content="Beschreibung der
    Webseite">
</head>

<body>
<!-- Der body erscheint im Browserfenster -->
</body>
</html>
```

4. Speichern Sie die Datei.

Im Browser sehen Sie von diesem Grundgerüst (fast) nichts, nur links oben in der Titelleiste steht der Titel »HTML-Grundgerüst - Vorlage«. In den folgenden Abschnitten werden die einzelnen Bestandteile dieses Grundgerüsts genauer erklärt.

Für uns Menschen empfiehlt es sich, den Quelltext möglichst übersichtlich über mehrere Zeilen zu verteilen und hierarchische Abhängigkeiten mit Einrückungen zu verdeutlichen. Unübersichtlich wird der Quelltext später ganz von allein. Dem Browser ist das egal. Von

ihm aus könnte der gesamte Quelltext in einer einzigen Zeile stehen. Whitespace wie Leerstellen, Tabstopps und Zeilenumbrüche ignoriert er einfach, ihn interessieren nur die Tags.

### 3.3 Die Dokumenttyp-Definition: <!DOCTYPE html>

Die Dokumenttyp-Definition, kurz DOCTYPE genannt, muss in der allerersten Zeile des Quelltextes stehen:

```
<!DOCTYPE html>
```

Groß- und Kleinschreibung spielt genau genommen keine Rolle, aber die gezeigte Variante mit einem großen DOCTYPE und einem kleinen html ist am weitesten verbreitet.

Der DOCTYPE hat nur eine Aufgabe, nämlich den Browser in den standardkonformen Modus zu versetzen. Mit einem nicht korrekten DOCTYPE gehen einige Browser in den sogenannten »Quirks-Modus« (»Pfuschmodus«), wodurch die Gestaltung einer Webseite per CSS unberechenbar werden kann. Achten Sie also darauf, dass in der allerersten Zeile des Quelltextes ein korrekter DOCTYPE steht.

Falls Sie sich schon mal mit HTML beschäftigt haben, kommt Ihnen der hier vorgestellte DOCTYPE vielleicht ein bisschen kurz vor. In früheren HTML-Versionen hatte der DOCTYPE eine eher unübersichtliche Syntax, und es gab ihn außerdem noch in mehreren Geschmacksrichtungen, wie *Transitional* und *Strict*. Darüber müssen Sie sich heute beim Lernen von HTML keine Gedanken mehr machen.

Draußen im Web gibt es noch Milliarden Webseiten mit diesen alten DOCTYPEs und für bereits vorhandene Webseiten muss nichts geändert werden. Die alten DOCTYPEs sind gut und funktionieren. Aber für neue Seiten gibt es keinen Grund, nicht den neuen DOCTYPE zu verwenden. Er hat keinerlei Nachteile, und die Browser verstehen ihn. Alle.

### 3.4 Das Stammelement <html>

Mit html folgt das bereits im DOCTYPE angekündigte Stammelement, von dem alle Elemente einer Webseite abstammen. html erstellt eine

(unsichtbare) Kiste, in der `head` und `body` und alle anderen Elemente enthalten sind. Direkt nach dem DOCTYPE steht der Anfangs-Tag `<html>`, in dem noch die Sprache der Webseite angegeben wird, und ganz am Ende in der letzten Zeile steht das Ende-Tag `</html>`:

```
<html lang="de"> ... </html>
```

Das Attribut `lang` (kurz für *language*) gibt die Sprache an, in der die Webseite geschrieben ist (`de` = deutsch). `lang` ist ein sogenanntes *globales Attribut*, kann also in fast allen HTML-Elementen benutzt werden.

In früheren HTML-Versionen war für das Stammelement manchmal die Definition eines so genannten XML-Namensraumes erforderlich. Da sowieso kaum jemand wirklich verstanden hat wozu das gut sein sollte, hat man in HTML5 beschlossen, es einfach wegzulassen. Sehr praktisch.

### **<html>: Wurzel- oder Stammelement?**

In vielen deutschsprachigen HTML-Dokumentationen wird `html` übrigens auch als *Wurzelement* bezeichnet, was eine wörtliche Übersetzung des englischen »root« ist. Ich bevorzuge die Bezeichnung *Stammelement*, denn alle HTML-Elemente einer Webseite stammen direkt oder indirekt von `html` ab.

**Tipp**

## **3.5 Informationen über die Webseite: <head>**

Zwischen `<head>` und `</head>` steht eine Art Vorspann für die Webseite, der nicht im Textfenster des Browsers erscheint und Informationen über das Dokument selbst enthält, die von Browsern oder Suchmaschinen ausgewertet werden. Die vorläufig wichtigsten Elemente im Kopfbereich sind `title` und `meta`.

### **Die Angabe des Zeichensatzes: <meta charset="utf-8">**

Das erste `meta`-Element folgt direkt nach dem öffnenden `<head>`-Tag und sagt dem Browser, dass er zur Darstellung der Webseite den Zeichensatz UTF-8 benutzen soll:

```
<meta charset="utf-8">
```

Die Datei muss in Ihrem Editor natürlich auch tatsächlich mit dem hier angegebenen Zeichensatz gespeichert worden sein. Wenn Ihnen das Wort »Zeichensatz« nicht so viel sagt, finden Sie am Ende dieses Kapitels ab Seite 75 eine kurze Einführung in das Thema.

## Der Name der Seite: <title>

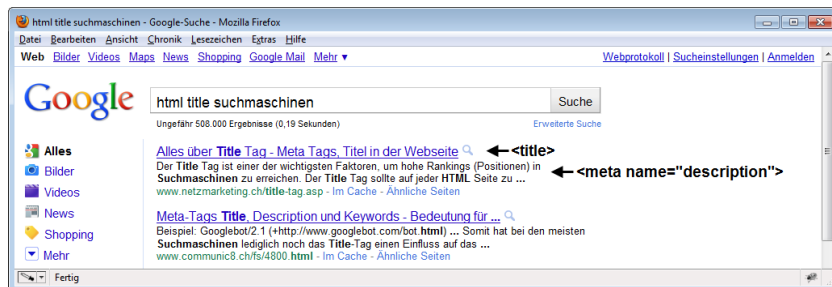
title ist wahrscheinlich das am meisten unterschätzte HTML-Element. Es enthält einen kurzen Text, der an verschiedenen Stellen wieder auftaucht:

- in der Browser-Tittleiste und den Ausklapplisten der Vor- und Zurück-Buttons
- als Beschriftung der Registerkarten (Tabs) in Browsern
- als Namensvorschlag für Favoriten bzw. Lesezeichen (Bookmarks)
- in den Ergebnislisten der Suchmaschinen als optisch hervorgehobener Link

Ein guter Titel ist kurz und beschreibt den Inhalt der Webseite, wenn man diese nicht sieht. *Startseite – Little Boxes (Webseiten gestalten mit HTML und CSS)* ist zum Beispiel informativer als ein simples *Willkommen* oder gar ein *Unbenanntes Dokument*.

Abbildung 3.1 zeigt die Ergebnisliste einer Suchmaschine. Die blaue, anklickbare erste Zeile eines Treffers ist der <title> der Webseite. Wenn Sie also nach einem guten Titel suchen, sollten Sie sich die Webseite in Gedanken in der Ergebnisliste einer Suchmaschine vorstellen und überlegen, was hier groß und blau erscheinen soll.

Abbildung 3.1:  
<title> und  
<meta name=  
"description">  
bei Google





## Infos für Suchmaschinen: <meta name="description">

Im head des Quelltextes können Sie neben dem Zeichensatz und einem aussagekräftigen title noch eine sehr nützliche meta-Angabe für Suchmaschinen und deren Robots machen:

```
<meta name="description" content="...">
```

Dieses Element enthält eine kurze Beschreibung des Inhalts der Webseite, die von vielen Suchmaschinen in der Trefferliste für den kurzen beschreibenden Text verwendet wird (siehe Abbildung 3.1).

Die genauen Empfehlungen für den beschreibenden Text variieren je nach Quelle, aber es sollten ungefähr zwei bis drei ganze Sätze mit einer Gesamtlänge zwischen 80 und 150 Zeichen sein. Sie sollten hier also keine Romane schreiben und möglichst die für diese Seite relevanten Suchbegriffe benutzen. Machen Sie Werbung für die Seite. Kurz und knackig. Dazu ist dieses Element da.

### Seiten, Sites und »meta«-Elemente

Im Deutschen werden die Begriffe *Webseite* und *Website* bzw. deren Kurzformen *Seite* und *Site* oft verwechselt oder durcheinander gebraucht. Eine *Seite* (mit sanftem S) ist ein einzelnes Dokument, eine *Site* (mit scharfem S) bezeichnet hingegen eine Menge von zusammengehörigen Seiten. Eine *Site* besteht also aus *Seiten*.

meta-Elemente gelten immer nur für die Seite, auf der sie stehen, nicht für die ganze Site.

### Tipp

## Weitere »meta«-Elemente

Mit einem weiteren meta-Element können Sie den Suchmaschinen sagen, ob und wie sie eine Seite indizieren sollen:

```
<meta name="robots" content="index, follow">
```

Damit erlauben Sie den Suchmaschinenrobots explizit, diese Seite zu indizieren und allen Links zu folgen. Das ist zwar nicht wirklich nötig, da ein wohlerzogener Robot das sowieso tut, aber es gibt auch noch die Werte *noindex* und *nofollow*. So bedeutet *noindex*, *nofollow* zum Beispiel »Diese Seite nicht indizieren und auch keinen Links fol-

gen«. Für eine Seite mit einem Kontaktformular wäre das durchaus adäquat. Eine gemischte Angabe wie *index,nofollow* geht natürlich auch.

Und was ist mit `<meta name="keywords" content="...">`? Irgendwie hat sich die Gewohnheit, gewünschte Suchbegriffe als *keywords* im head-Bereich zu speichern, aus den späten 1990er-Jahren in dieses Jahrtausend gerettet. Es schadet zwar nicht, die wichtigsten Stichwörter zum Inhalt der Webseite aufzulisten, aber für die Suchmaschinen können Sie sich die Mühe sparen. Diese sind dazu übergegangen, die Keywords schlicht und einfach zu ignorieren. Wichtiger ist, dass die Suchbegriffe an den richtigen Stellen auf der Seite auftauchen. Im title der Seite zum Beispiel, und im Text der Überschriften und Hyperlinks.

### 3.6 Der Inhalt der Webseite: `<body>`

Nach dem Kopf folgt der (Text-)Körper, auf Englisch *body*. Fast alles, was zwischen `<body>` und `</body>` steht, wird später im Browserfenster dargestellt.

Im Moment gibt es zu `body` noch nicht soviel zu sagen, aber ab dem nächsten Kapitel werden Sie die meiste Zeit mit dem Erstellen und Gestalten von HTML-Elementen zwischen `<body>` und `</body>` verbringen.

### 3.7 HTML-Elemente: Anfangs-Tag, Inhalt und Ende-Tag

Die Namen der HTML-Elemente sind Abkürzungen für einen englischen Begriff. Wie Sie bereits gesehen haben, wird ein normaler Fließtextabsatz von `<p>` und `</p>` umschlossen: Das *p* steht wie gesagt für *paragraph*, auf Deutsch *Absatz*.

Fast alle HTML-Elemente haben ein Anfangs- und ein Ende-Tag. Dabei unterscheidet sich das Ende-Tag vom Anfangs-Tag durch einen vorangestellten Schrägstrich:

`<p>Dieser Text ist ein Absatz.</p>`

`<p>` heißt für den Browser »Hier fängt ein Absatz an«, `</p>` dementsprechend »Hier ist ein Absatz zu Ende«. Da die Begriffe *Element* und *Tag* ständig zu mittelschweren Verwirrungszuständen führen, zeigt Abbildung 3.2 ein Beispiel.

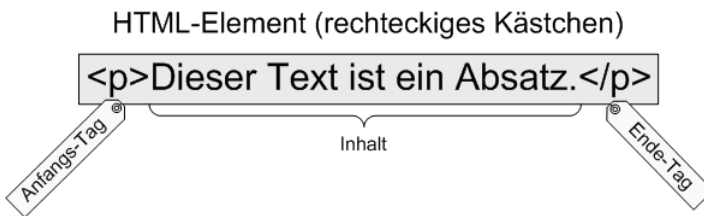


Abbildung 3.2:  
Schematischer  
Aufbau eines  
HTML-Elements

Ein HTML-Element besteht aus drei Teilen. Im Beispiel sind das:

- das *Anfangs-Tag* in spitzen Klammern: `<p>`
- der *Inhalt*: Dieser Text ist ein Absatz.
- das *Ende-Tag* ebenfalls in spitzen Klammern: `</p>`

Alle drei Teile zusammen heißen *HTML-Element*, und HTML-Elemente werden im Browserfenster immer als rechteckige Kästchen dargestellt. Alle Texte und Grafiken einer Webseite liegen in solchen Kästchen. Ohne Ausnahme. Man könnte so ein Kästchen auch *Container* nennen. Auf Englisch heißt es *box*. Eine Webseite besteht also buchstäblich aus *lauter little boxes*.

### 3.8 Know-how: Wissenswertes über Zeichensätze

Auf einer gut geschriebenen Webseite steht ziemlich am Anfang des Quelltextes ein *meta-Element*, das den Zeichensatz definiert, mit dem der Browser die Webseite darstellen soll. Was es damit auf sich hat und welche Zeichensätze zur Wahl stehen, erläutert der folgende Abschnitt.

#### Zeichensätze sind Schablonen für Buchstaben

Im Arbeitsspeicher eines Computers stehen keine Buchstaben, sondern Zahlen. Diese Zahlen werden anhand einer Schablone am Bildschirm als Zeichen dargestellt. Eine solche Schablone wird im Allge-

meinen »Zeichensatz« genannt und war lange Zeit auf maximal 256 Zeichen begrenzt.

Da in den auf diesem Planeten benutzten Sprachen weit mehr als 256 Zeichen vorkommen, gibt es verschiedene Zeichensätze mit jeweils unterschiedlichen Zeichen. Deshalb muss der Computer beim Speichern und bei der Darstellung am Bildschirm wissen, welche dieser Schablonen er benutzen soll.

Die der UNO unterstellte »International Standards Organization« (ISO) hat in der Norm mit der Nummer 8859 fünfzehn verschiedene Zeichensätze definiert, in denen je nach Sprache unterschiedliche Zeichen bereitgestellt werden. Der Zeichensatz ISO-8859-1 (der auch manchmal als ANSI bezeichnet wird) ist dabei für westeuropäische Sprachen gedacht und wird heutzutage noch häufig eingesetzt.

Um die Grenze von 256 Zeichen pro Schablone zu sprengen, wurde *Unicode* erfunden. Das »Uni« im Namen steht für *universell*, ein Zeichensatz für alle Zeichen. Das Unicode-Format *UTF-8* hat im Internet eine zentrale Bedeutung als globale Zeichenkodierung und kann bis zu 1.114.112 Zeichen abbilden.

UTF-8 hat sich in den letzten Jahren zum Zeichensatzstandard für das Web entwickelt, und moderne Betriebssysteme wie Windows (XP, Vista oder 7), Linux oder Mac OS X haben damit keine Probleme. Wenn nichts dagegenspricht, sollten Sie immer mit UTF-8 arbeiten, allerdings spielen bei Webseiten der Editor und der Webserver noch eine besondere Rolle:

- Ihr Editor muss Dateien als UTF-8 speichern können.
- Der Webserver muss sie dementsprechend ausliefern.

Mehr dazu erfahren Sie in den folgenden beiden Abschnitten.

## Tipp

### Details zu Zeichensätzen beim W3C

Eine gelungene Darstellung der Zusammenhänge und Empfehlungen im Umgang mit Zeichensätzen findet sich beim W3C (auf Deutsch):

- [w3.org/International/questions/qa-what-is-encoding.de.php](http://w3.org/International/questions/qa-what-is-encoding.de.php)
- [w3.org/International/tutorials/tutorial-char-enc/](http://w3.org/International/tutorials/tutorial-char-enc/)

## UTF-8 und der Editor

Zunächst zum Editor: Er muss die Speicherung von Dateien mit dem Zeichensatz UTF-8 unterstützen, und einige ältere Editoren können das nicht.

Idealerweise sollte der Editor UTF-8 in der Variante ohne *BOM* anbieten. Das *BOM* (kurz für »Byte Order Mark«) besteht aus drei Bytes, die ganz am Anfang des Dokuments stehen. Da das *BOM* unter UTF-8 optional ist und es bei der Ausgabe im Browser unter ungünstigen Umständen als wirre Zeichenfolge (ï»¿) am Anfang der Webseite erscheint und auch sonst nur Probleme macht, sollten die Dateien möglichst ohne *BOM* gespeichert werden.

Um herauszufinden, ob Ihr Editor »UTF-8 ohne BOM« kann, schauen Sie sich dessen Menüs oder Feature-Liste genauer an. Beim in Abbildung 3.3 dargestellten Editor wird der gewünschte Zeichensatz einfach im Dialogfeld **SPEICHERN UNTER ...** ausgewählt:

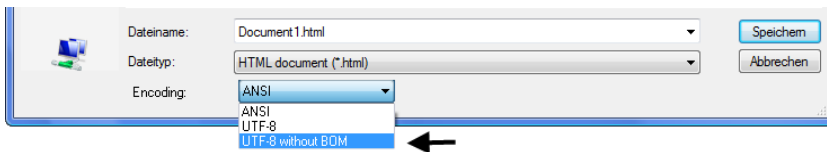


Abbildung 3.3:  
Ein Editor mit der  
Option »UTF-8  
ohne BOM«

Unter Mac OS X wird übrigens standardmäßig ohne *BOM* gespeichert, sodass ein Editor UTF-8 eigentlich immer ohne *BOM* abspeichert, solange nicht explizit »mit *BOM*« dransteht.

Bei Windows-Editoren ist die Lage unterschiedlich. Im weit verbreiteten PSPad zum Beispiel ist die Einstellung dafür gut versteckt: Im Menü **EINSTELLUNGEN** unter **PROGRAMM EINSTELLEN...** gibt es in der Kategorie **PROGRAMM-VERHALTEN** die Option **IDENT. BYTES BEI UTF-8 HINZUFÜGEN**. Diese Option muss ausgestellt sein, um eine Datei ohne *BOM* zu speichern.

### Die Kodierung der Sonderzeichen: &kuerzel;

Wenn Sie oder Ihr Editor sich die Mühe gemacht und alle Sonderzeichen nach dem Schema *&kuerzel;* kodiert haben, kommen sie auf jeden Fall richtig beim Besucher an, unabhängig vom verwendeten Zeichensatz.

### Tipp

## UTF-8 und der Webserver

Neben dem Editor spielt auch der Webserver noch eine gewichtige Rolle, denn er muss die Webseite entweder ohne oder mit der Angabe des korrekten Zeichensatzes ausliefern: Wenn Sie die Seite im Editor als UTF-8 speichern, der Webserver sie mit einem Hinweis auf ISO8859-1 ausliefert und Sie im Quelltext wieder UTF-8 deklarieren, kann es sein, dass die Browser durcheinander kommen und nicht kodierte Sonderzeichen unter Umständen falsch darstellen.

Um herauszufinden, wie Ihr Webserver sich verhält, surfen Sie zu Ihren Webseiten und rufen dann in der Symbolleiste *Web Developer* im Menü INFORMATIONEN ganz unten den Befehl ANTWORT-HEADER ANZEIGEN auf.

Dadurch wird der *HTTP-Header* sichtbar – die Antwort des Webserver auf die Anfrage des Browsers. Im angezeigten HTTP-Header ist die Zeile Content-Type wichtig. Wenn dort keine Zeichensatzangabe vorhanden ist, mischt der Webserver sich nicht ein. Ist dort aber ein Zeichensatz angegeben, sollte dieser mit dem von Ihnen beim Speichern verwendeten Zeichensatz übereinstimmen.

In Abbildung 3.4 gibt der Webserver die Webseite mit der Angabe des Zeichensatzes ISO-8859-1 aus.

Abbildung 3.4:  
Der Webserver  
nutzt  
»ISO-8859-1«.



### Tipp

#### Details zur Wahl des Zeichensatzes

Details zur Wahl des Zeichensatzes finden Sie in dem hervorragenden Artikel »The Definitive Guide to Web Character Encoding« von Tommy Olson:

■ [sitepoint.com/article/guide-web-character-encoding](http://sitepoint.com/article/guide-web-character-encoding)

### 3.9 Auf einen Blick

Hier sind noch einmal die wichtigsten Punkte dieses Kapitels im Überblick:

- HTML ist eine »Hypertext-Etiketten-Dranklebe-Sprache«. Es ist keine Programmier- oder Seitenbeschreibungssprache und dient nicht zum Gestalten.
- Jede Webseite hat ein HTML-Grundgerüst, das aus folgenden Teilen besteht:
  - dem DOCTYPE in der ersten Zeile des Quelltextes
  - dem Stammelement `html`
  - dem Vorspann `head` mit `title` und `meta`-Element(en)
  - dem Inhalt `body`, der im Browserfenster dargestellt wird
- Einige wissenswerte Details zu HTML-Elementen sind:
  - Die Namen der HTML-Elemente stammen aus dem Englischen.
  - Fast alle HTML-Elemente haben ein Anfangs- und ein Ende-Tag.
  - Ein HTML-Element besteht aus *Anfangs-Tag*, *Inhalt* und *Ende-Tag*.
- UTF-8 hat sich als universeller Zeichensatz im Web durchgesetzt.
- Wenn Sie `<meta charset="utf-8">` in den Quelltext schreiben, muss die Seite im Editor auch tatsächlich mit dem Zeichensatz UTF-8 gespeichert werden. Am besten »ohne BOM«.

