

## Kapitel 4

# Die ersten Schritte mit HTML und CSS

*HTML ist eine Sprache, und wie eine Sprache lernt man es: Man prägt sich die Bedeutung der Wörter ein und lernt den Aufbau aussagekräftiger Sätze, bis man damit auch komplexere Ideen ausdrücken kann.*

Dank der kostenlosen Anbieter haben Sie eine funktionierende Website im Netz, mit der Sie Ihre festgelegten Ziele fürs Erste erreichen. Es gibt aber gute Gründe dafür, nicht die Hände in den Schoß zu legen, sondern jetzt erst so richtig anzufangen.

### 4.1 Die Vorteile einer eigenen Website

Die Dienste für Web-Visitenkarten und Blogs haben natürlich den Vorteil, dass sie unkompliziert zu benutzen sind. Für die Umsetzung der Beispielprojekte und damit Ihrer ersten beiden Webauftritte mussten Sie nicht viel mehr können als für die Arbeit mit einem Textverarbeitungsprogramm.

Diese Einfachheit hat aber ihren Preis: Während die Auswahl und das Aussehen der Vorlagen anfangs beeindruckend sind und man die Gestaltung noch wochenlang weiter verfeinern kann, werden Sie früher oder später doch an die Grenzen des Vorlagensystems stoßen: Sie wollen selbst wählen, welche Seite Ihre Startseite ist, oder Sie wollen die Anordnung der Seitenelemente frei bestimmen.

Zu diesen kreativen Grenzen kommen technische Einschränkungen hinzu. So gehören Ihnen »Ihre« Website und die Inhalte je nach Anbieter nur begrenzt, und auf die Erreichbarkeit der Site im Internet haben Sie keinen Einfluss – wenn die Server des Anbieters ausfallen, können Sie nur warten, bis sie wieder online sind. Sie haben keinen direkten Zugriff auf die Dateien und keine Garantie, dass Ihre Site in ein paar Monaten noch verfügbar sein wird.

Auch entwickelt sich das Internet ständig weiter. Dadurch werden neue Funktionen und Benutzerinteraktionen möglich. So wurde etwa das Einbinden von Videos und Kar-

ten oder die Verknüpfung mit sozialen Netzwerken erst im Laufe der Zeit zu einem festen Bestandteil vieler Websites. Ein Dienst schränkt Sie hier insofern ein, als dass Sie warten müssen, bis er eine neue Funktion in seine Vorlagen eingebaut hat, bevor Sie sie auf Ihrer Website einsetzen können.

All diese kreativen und technischen Einschränkungen können Sie umgehen, wenn Sie die Grundlagen von HTML und CSS lernen, Ihre Website selbst erstellen und auf einem Webserver veröffentlichen, auf den Sie direkten Zugriff haben.

Aber da ist noch mehr: Wenn Sie die grundlegenden Webtechniken gelernt haben, sind Sie nicht nur dazu in der Lage, eine mit einem Dienst erstellte Website mit eigenen Mitteln nachzubauen. Das Gelernte ermöglicht Ihnen darüber hinaus, Hunderte kostenlose Werkzeuge zu benutzen. Mit diesen werden Ihre Websites noch nützlicher, interaktiver und schöner – Sie können damit an einem Tag Funktionen einbauen, für die Sie ohne fremde Hilfe Wochen oder Monate brauchen würden. Indem Sie selbst HTML und CSS lernen, können Sie von der kreativen Energie der gesamten Netzgemeinde profitieren und nicht nur von der Arbeit der 50 oder 100 Angestellten eines Blogdienstes. Wie Sie einige der besten Werkzeuge für Ihre Websites nutzen, zeigen wir Ihnen im vierten Teil des Buches.

## 4.2 Alle Webseiten bestehen aus HTML

Die Basis für die Entfaltung des ganzen Potenzials des Webs ist HTML, und zwar einfach deshalb, weil jede Webseite aus HTML besteht.

### Website oder Webseite?

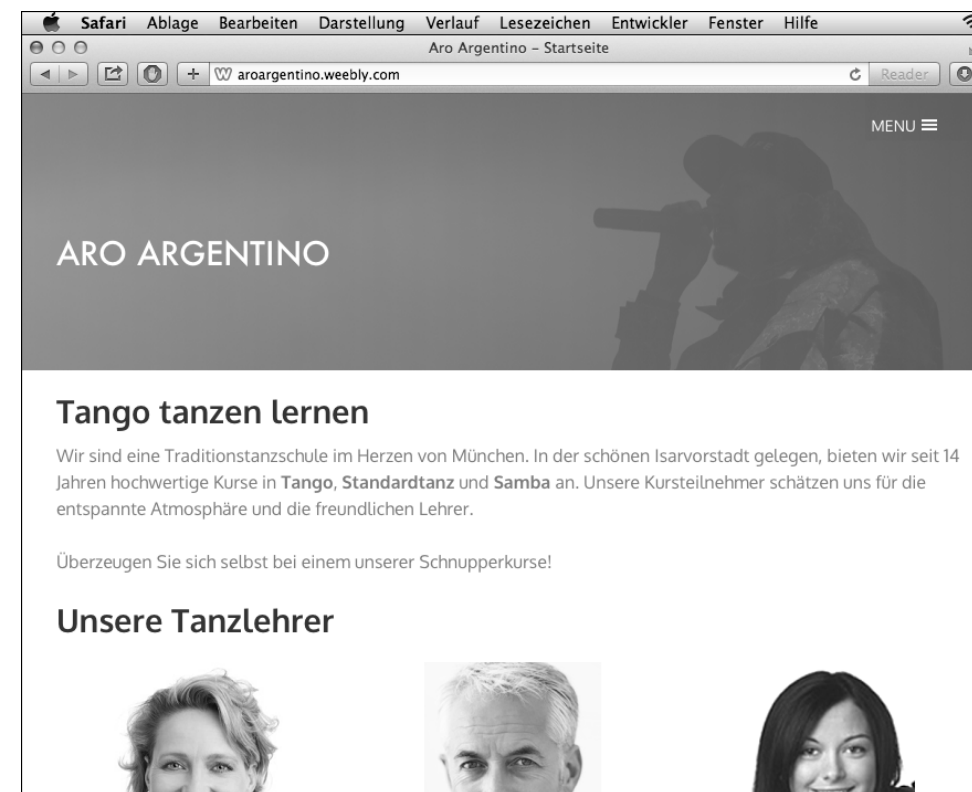
Zum Unterschied zwischen *Website* und *Webseite* siehe auch Kapitel 1, »Ein eigener Webauftritt in drei Minuten«. Die Website ist der gesamte Internetauftritt, eine Webseite dagegen ist eine einzelne Seite des Auftritts.

Dass auch Ihre mit *Jimdo*, *Weebly*, *Blogger* oder einem anderen Dienst erstellten Seiten aus HTML bestehen, können Sie mithilfe Ihres Browsers leicht selbst nachprüfen.

### Dafür benötigen Sie:

- Browser (*Firefox*, *Chrome*, *Safari*, *Internet Explorer* oder *Microsoft Edge*)

**Schritt 1** | Öffnen Sie mit Ihrem Browser die Startseite Ihres Blogs aus Kapitel 3, »Ein besserer Webauftritt in drei Stunden«, oder die Startseite unseres Beispielprojekts (unter [bnfr.de/q1303](http://bnfr.de/q1303)). Letzteres sehen Sie in Abbildung 4.1.



**Abbildung 4.1** Die Startseite unseres Beispielprojekts aus Kapitel 3, dargestellt im Browser Safari.

Was Sie auf Ihrem Bildschirm sehen, ist nicht das HTML der aktuellen Webseite, sondern eine grafische Darstellung, die Ihr Browser für Sie aus dem HTML erstellt hat. Sie können in Ihrem Browser statt der grafischen Darstellung auch das HTML selbst anzeigen – den sogenannten *Quelltext* der Webseite.

**Schritt 2** | Lassen Sie sich den Quelltext der aktuellen Webseite anzeigen. Bei den meisten Browsern finden Sie die entsprechende Funktion (SEITENQUELLTEXT ANZEIGEN o. Ä.) entweder nach einem Rechtsklick auf die dargestellte Seite oder nach Aufruf eines

Menüeintrags (ENTWICKLERWERKZEUGE o. Ä.). Wenn Sie die Funktion nicht finden, geben Sie den Namen Ihres Browsers und »quelltext anzeigen« in eine Suchmaschine ein. Damit finden Sie sicher, was Sie brauchen, auch wenn sich die Position oder der Name der Funktion bei Ihrem Browser wieder einmal geändert haben.

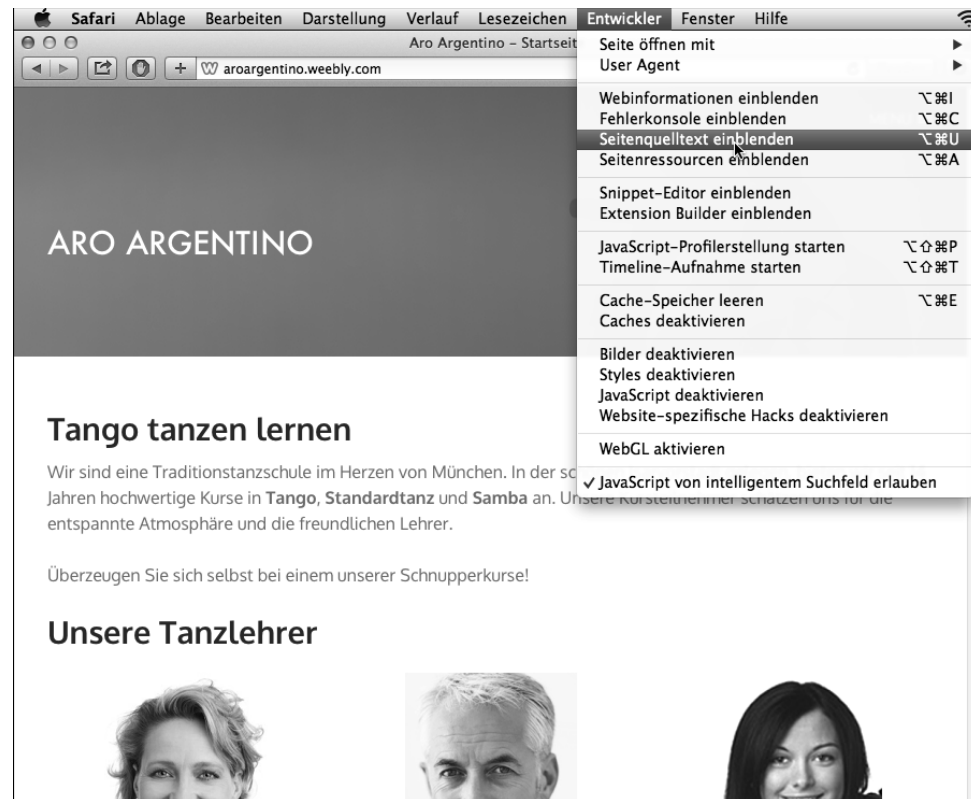


Abbildung 4.2 Das »Entwickler«-Menü von Safari. Je nach Browser unterscheidet sich der Menüaufbau etwas, aber alle neueren Browser können den Quelltext einer Seite anzeigen.

Daraufhin öffnet der Browser ein neues Fenster mit dem Quelltext. In Abbildung 4.3 sehen Sie das Fenster in Safari.

Wenn Sie einen anderen Browser benutzen, sieht die Anzeige des Quelltexts etwas anders aus, aber das Prinzip ist dasselbe.

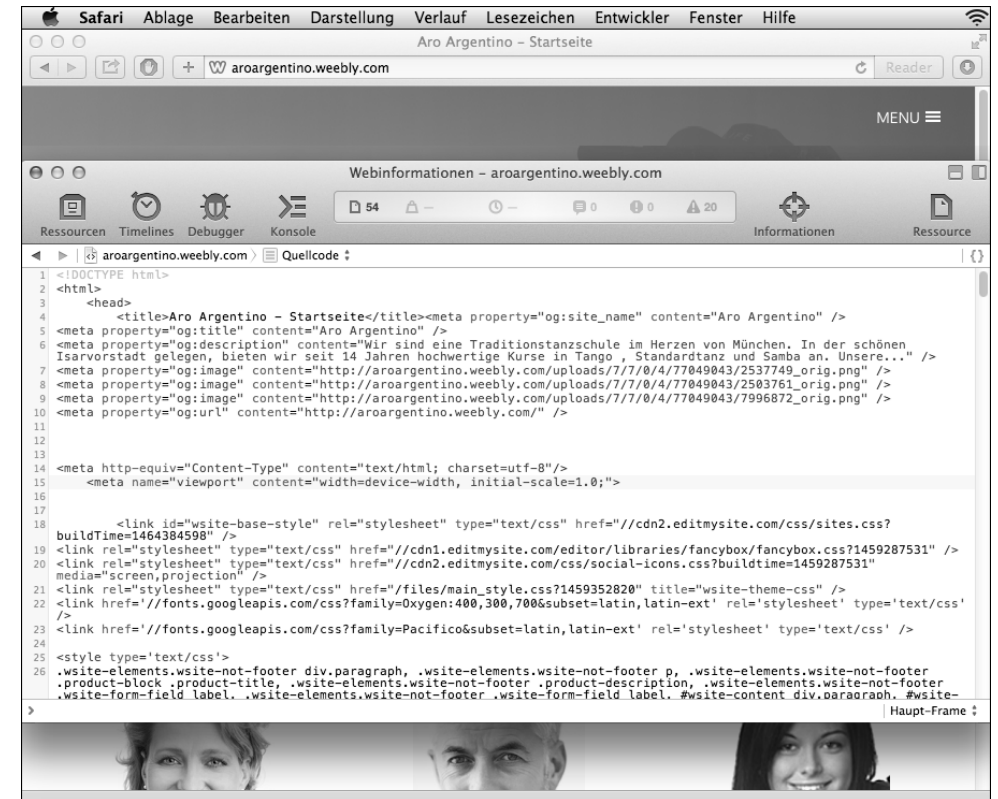


Abbildung 4.3 Neben der Anzeige des Quelltexts bieten moderne Browser eine Fülle von Werkzeugen zur Untersuchung einer Webseite.

Der Quelltext – das HTML der angezeigten Webseite – besteht also tatsächlich nur aus *Text*. Bilder, Farben, alles, was die Gestaltung Ihrer Seiten ausmacht, suchen Sie vergebens. Erst Ihr Browser fügt für die grafische Darstellung Formatierungen wie Fettschrift hinzu, bindet verknüpfte Bilder ein, setzt anklickbare Hyperlinks und so weiter.

#### Was bedeutet die Abkürzung HTML?

HTML steht für **H**yper**T**ext **M**arkup **L**anguage. Das bedeutet übersetzt »Sprache zur Auszeichnung von Hypertext«. Hypertext ist Text mit Querverweisen, also (Hyper-)Links. Mit »Auszeichnung« ist gemeint, dass Teile des Textes mit Anmerkungen versehen sind, die dem Browser sagen, wie er den Text darstellen soll.

Lassen Sie sich nicht davon verunsichern, dass der Quelltext Ihrer Seite aus einer Menge unverständlicher Zeichen besteht oder dass diese Zeichen auf den ersten Blick nicht ge-

rade dazu geeignet scheinen, eine stilvolle Tangoseite mit weinrotem Hintergrund zu formen. Mit HTML kann man ziemlich viele – auch ziemlich komplexe – Dinge machen. Aber das Schöne daran ist, dass die Grundlagen ganz einfach sind.

In Kapitel 1, »Ein eigener Webauftritt in drei Minuten«, haben wir erklärt, woher das HTML bei Websites kommt, die mithilfe eines Dienstes erstellt wurden: Der Dienst nimmt Ihre geschriebenen Texte, Ihre Formatierungsvorgaben und die gewählte Gestaltungsvorlage und übersetzt das alles in ein HTML-Dokument, das Sie wiederum jetzt gerade in Ihrem Browser sehen. ■

#### HTML: Dokument oder Seite?

Beim Thema HTML werden die Begriffe *Dokument* und *Seite* oft gleichbedeutend benutzt. Wenn es um die Datei auf Ihrem Computer geht, die angelegt, bearbeitet oder gespeichert wird, spricht man am ehesten von einem *HTML-Dokument*. Wenn der Inhalt des Dokuments dann im Browserfenster grafisch dargestellt wird, spricht man eher von einer *HTML-Seite*.

### 4.3 Sie schreiben selbst HTML

Für Ihre selbst erstellten Webseiten können Sie nicht mehr auf den oben beschriebenen Übersetzungsdienst zurückgreifen. Deshalb erzeugen wir die fertige HTML-Datei, die an den Browser eines Besuchers übertragen wird, jetzt selbst: Wir schreiben also direkt den Quelltext.

#### Dafür benötigen Sie:

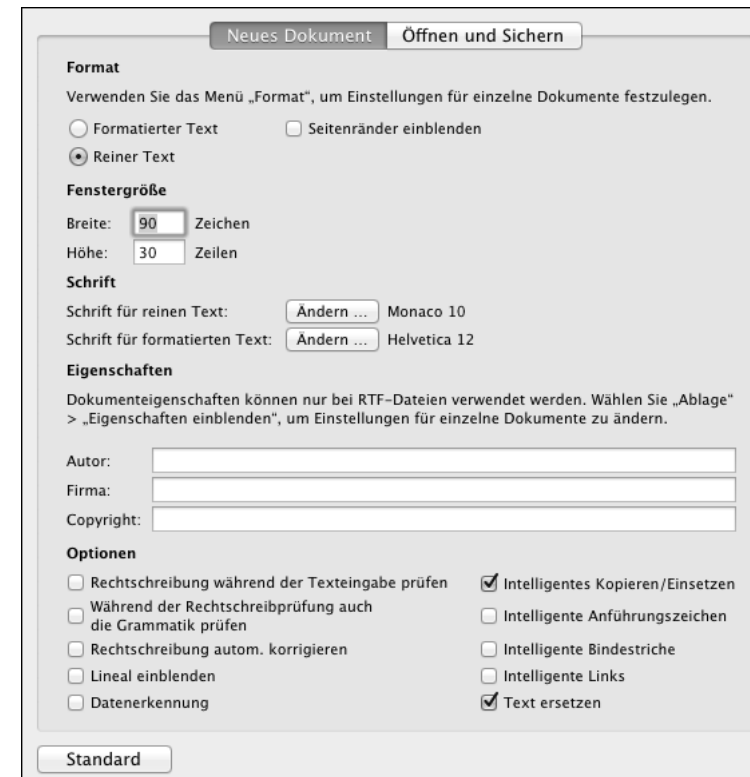
- ▶ einen kostenlosen Texteditor (z. B. *Editor* für Windows, *TextEdit* für Mac)
- ▶ einen Browser (Firefox, Chrome, Safari, Internet Explorer oder Microsoft Edge in der jeweils aktuellsten Version).

Die Versionsnummer Ihres Browsers erfahren Sie im entsprechenden Menüeintrag (meist heißt er ÜBER oder INFO). Neuere Versionen laden Sie kostenlos von der Website des Browserherstellers herunter.

**Schritt 1** | Öffnen Sie den einfachsten Texteditor, den Sie auf Ihrem Rechner finden können.

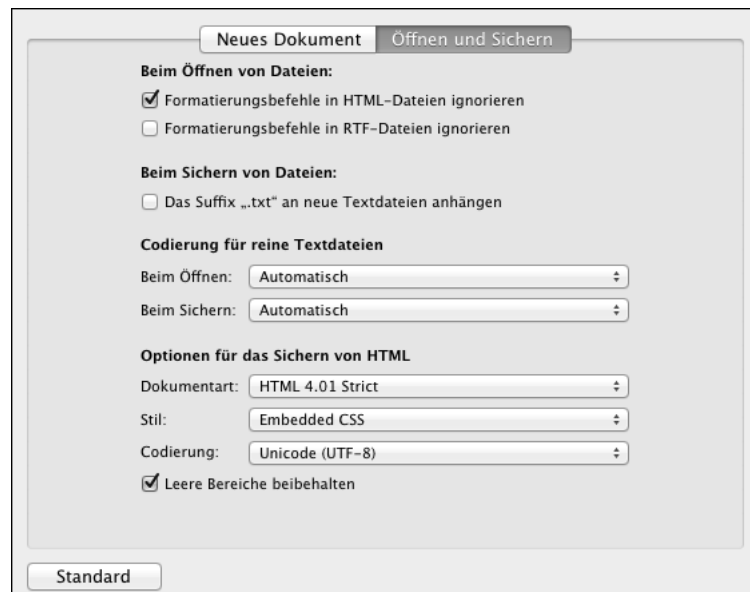
Unter Windows ist das z. B. der *Editor*, auf dem Mac das Programm *TextEdit*. Beide Programme werden mit dem Betriebssystem mitgeliefert und sollten bereits installiert sein.

**Schritt 2** | **Nur bei TextEdit:** Klicken Sie im Programmmenü auf TEXTEDIT und dann auf EINSTELLUNGEN. Wählen Sie im Reiter NEUES DOKUMENT unter FORMAT die Option REINER TEXT (siehe Abbildung 4.4).



**Abbildung 4.4** Selbst bei etwas so Einfachem wie einem simplen Textdokument gibt es unterschiedliche Formatierungsmöglichkeiten.

**Schritt 3** | **Nur bei TextEdit:** Klicken Sie dann auf den Reiter ÖFFNEN UND SICHERN, und wählen Sie dort unter BEIM ÖFFNEN VON DATEIEN die Option FORMATIERUNGSBEFEHLE IN HTML-DATEIEN IGNORIEREN. Entfernen Sie außerdem den Haken bei DAS SUFFIX ».TXT« AN NEUE TEXTDATEIEN ANHÄNGEN (siehe Abbildung 4.5).



**Abbildung 4.5** Wenn Sie unsere Anweisungen befolgen, wird Ihr Dokument korrekt als HTML formatiert.

Mit diesen Einstellungen stellen Sie sicher, dass TextEdit Ihr Dokument von Anfang an wie Quelltext behandelt und später beim erneuten Öffnen nicht etwa versucht, mit einer grafischen Darstellung selbst den Browser zu spielen.

Die weiteren Arbeitsschritte gelten wieder für alle Texteditoren:

**Schritt 4** | Erstellen Sie mit Ihrem Texteditor ein neues Dokument.

**Schritt 5** | Tippen Sie folgenden Text ein. Am besten fügen Sie zur besseren Übersichtlichkeit unter jeder Zeile eine Leerzeile ein:

```
<html>
<head>
```

#### Die richtige Taste

Die spitzen Klammern `<` und `>` finden Sie auf Standardtastaturen mit deutschem Tastenlayout links vom Buchstaben `Y`.

An den spitzen Klammern erkennt später der Browser, dass es sich um HTML-Bausteine, also *Markup* bzw. *Auszeichnungen* handelt.

**Schritt 6** | Der nächste Baustein beinhaltet den Titel der Seite – wir wählen »HTML«:

```
<title>HTML</title>
```

Sie können den Titel frei bestimmen, er sollte aber nicht zu lang sein. Drei bis vier Wörter sind genug.

**Schritt 7** | Die nächsten beiden HTML-Bausteine, die Sie in das Dokument schreiben, sehen so aus:

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

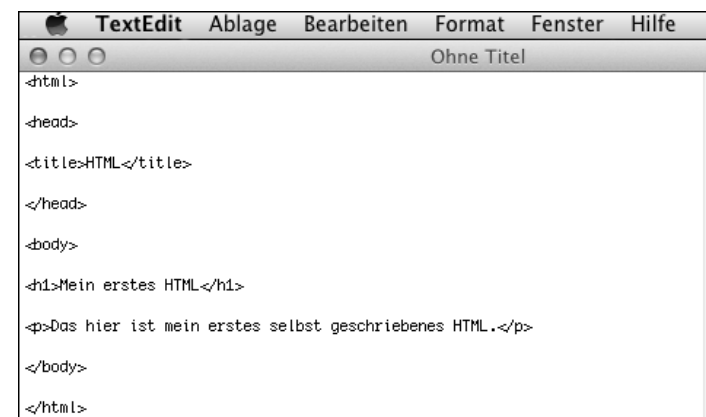
**Schritt 8** | Dann folgt der eigentliche Inhalt der Seite – eine Überschrift und ein kurzer Text:

```
<h1>Mein erstes HTML</h1>
<p>Das hier ist mein erstes selbst geschriebenes HTML.</p>
```

**Schritt 9** | Zum Schluss folgen noch mal zwei HTML-Bausteine:

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

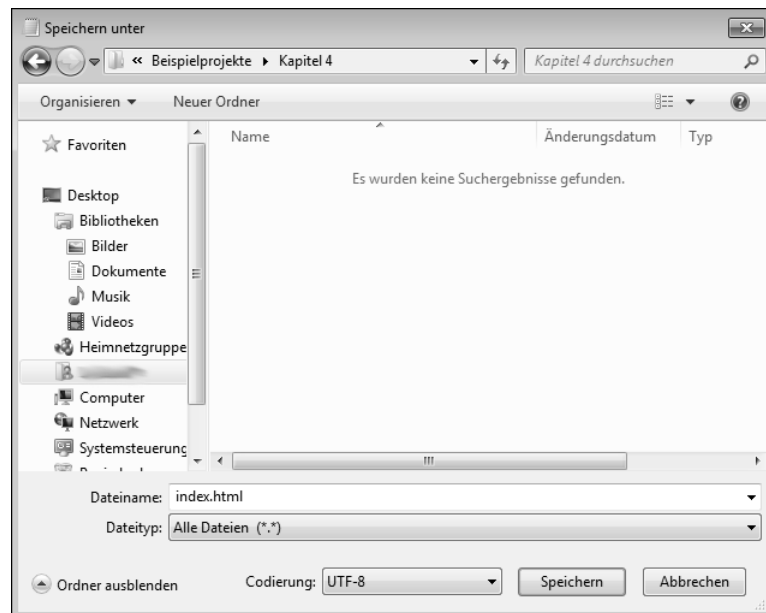
Wenn Sie nach jeder Zeile eine Leerzeile gelassen haben, sollte Ihr Dokument im Texteditor jetzt so ähnlich aussehen wie in Abbildung 4.6.



**Abbildung 4.6** Wenn Sie TextEdit benutzen und Ihr Text anders aussieht, prüfen Sie bitte, ob Sie unsere Hinweise in den Schritten 2 und 3 befolgt haben, bevor Sie ein neues Dokument erstellt haben.

**Schritt 10** | Speichern Sie das Dokument unter dem Namen *index.html*.

**Schritt 11** | **Nur beim Windows-Editor:** Dazu müssen Sie im SPEICHERN UNTER-Fenster bei DATEITYP die Option ALLE DATEIEN (\*.\*) wählen (siehe Abbildung 4.7).



**Abbildung 4.7** Auch beim »Editor« muss man ein paar Einstellungen vornehmen, damit das Dokument als HTML formatiert wird.

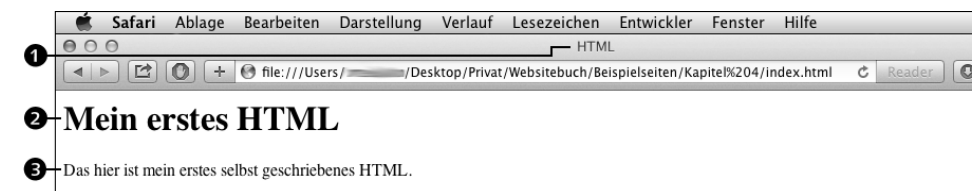
Wenn Sie einen anderen Texteditor nutzen und dieser beim Speichern die Dateierweiterung *.html* nicht erlaubt, speichern Sie das Dokument stattdessen als *.txt*, schließen den Editor und benennen die Dateierweiterung im Explorer (Windows) oder im Finder (Mac) von Hand in *.html* um.

#### Warum »index.html«?

Wenn Ihre Website aus nur einer Seite besteht, nennen Sie diese immer *index.html*. Wenn die Site aus mehreren Seiten besteht, nennen Sie immer die Startseite *index.html*. Der Grund dafür ist, dass die allermeisten Webserver einem Besucher standardmäßig diese Datei anzeigen, wenn er nur die Domain der Site (also etwa *www.facebook.com*) eingibt.

**Schritt 12** | Öffnen Sie Ihre HTML-Datei mit Ihrem Browser, indem Sie im Explorer (Windows) oder Finder (Mac) auf sie doppelklicken.

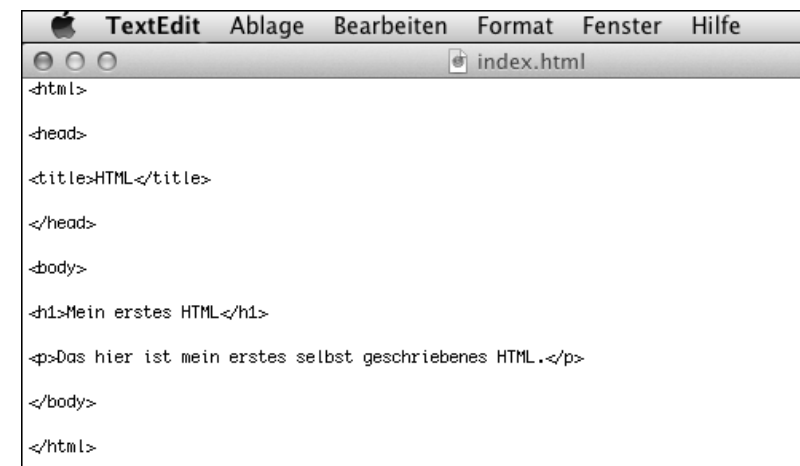
Ein Browser kann also nicht nur auf Dateien im Internet zugreifen, sondern auch auf Dateien auf Ihrem Rechner. Das Ergebnis sehen Sie in Abbildung 4.8.



**Abbildung 4.8** In der Adressleiste des Browsers ist immer der Speicherort einer Seite zu sehen – bei Ihrer ersten Seite liegt er eben auf der eigenen Festplatte.

Sie können sofort die im Texteditor angelegten Elemente wiedererkennen – den Titel, der in einer Leiste des Browserfensters ❶ angezeigt wird, die Überschrift ❷ und den kurzen Textabsatz ❸.

Und schon haben Sie Ihre erste selbst geschriebene HTML-Seite erstellt und im Browser betrachtet. Aber natürlich wollen Sie nicht nur eingeben, was wir Ihnen vorsagen, sondern Sie wollen verstehen, was genau Sie da eingegeben haben. Was waren das also für Bausteine, die Sie in das Dokument geschrieben haben? Wie ist HTML im Detail aufgebaut? Betrachten wir den im Editor eingegebenen Text nochmals (siehe Abbildung 4.9).



**Abbildung 4.9** Die Leerzeilen haben keinen Einfluss auf die Darstellung im Browser, erhöhen aber die Übersichtlichkeit.

Die Anordnung des Inhalts in einem HTML-Dokument entspricht zunächst der Logik, die wir von anderen Texten in Büchern usw. auch kennen: Gelesen wird von oben nach unten und von links nach rechts.

Die Grundbausteine für jedes HTML-Dokument sind *Elemente*. Diese werden durch sogenannte *Tags* in das Dokument eingefügt. An den Anfang des Elements kommt dessen *öffnender Tag* oder *Start-Tag*, ans Ende des Elements der *schließende Tag* oder *End-Tag*.

Den grundsätzlichen Aufbau eines Elements sehen Sie in Abbildung 4.10.

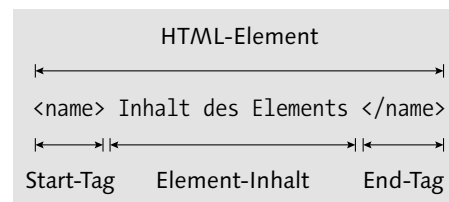


Abbildung 4.10 »Tag« ist hier Englisch und bedeutet »Markierung«.

Der Start-Tag besteht aus zwei spitzen Klammern und dem Namen des Elements. Der End-Tag sieht genauso aus, aber zusätzlich steht ein Schrägstrich zwischen der ersten Klammer und dem Elementnamen. Alles, was zwischen den Tags steht, ist der *Inhalt* des Elements.

Wenn Sie sich Ihr HTML-Dokument noch mal anschauen (siehe Abbildung 4.11), dann können Sie mehrere solcher Elemente mit Tags und Inhalt erkennen, etwa das `title`-, das `h1`- und das `p`-Element.

```

<html>
<head>
<title>HTML</title>
</head>
<body>
<h1>Mein erstes HTML</h1>
<p>Das hier ist mein erstes selbst geschriebenes HTML.</p>
</body>
</html>

```

Abbildung 4.11 Die drei im Browser direkt sichtbaren Elemente

Diese drei Elemente bzw. deren Inhalt sehen Sie ja (wie in Abbildung 4.8) direkt im Browser: Der Inhalt des `title`-Elements ist der Titel der Seite, der in der Kopfleiste des Browsers angezeigt wird. Das `h1`-Element ist eine Überschrift, was im Browser an der besonderen Schriftgröße und der Fettschrift zu erkennen ist.

### Was bedeutet das »h«?

*h* steht für *heading*, das englische Wort für *Überschrift*. `h1` ist eine Überschrift ersten Rangs, also diejenige, die von allen Überschriften auf einer Seite am deutlichsten hervorgehoben wird. Von `h1` bis `h6` kann man auf seinen Webseiten sechs unterschiedlich wichtige Überschriften verwenden, so wie zum Beispiel dieses Buch ebenfalls Überschriften verschiedenen Rangs enthält.

Das `p`-Element schließlich zeigt Ihr Browser als einen Absatz mit Text (*p* steht für *paragraph*, im Deutschen *Absatz*). Im Vergleich mit der Überschrift hat der Absatz eine kleinere Schriftgröße und keine Fettschrift.

In Ihrem HTML-Dokument, das Sie mit dem Texteditor erstellt haben, finden sich aber nicht nur diese drei Elemente. Sie erkennen weitere Start- und End-Tags, die aber nicht nebeneinander, sondern untereinander platziert sind. Auch diese markieren Elemente – nur besteht ihr Inhalt nicht aus einem zwischen dem Start- und End-Tag geschriebenen Text wie beim `title`-, `h1`- oder `p`-Element, sondern aus anderen Elementen. Auch bei diesen Elementen ist also alles, was zwischen Start- und End-Tag steht, deren Inhalt. Das veranschaulicht Abbildung 4.12.

```

<html>
  <head>
    <title>HTML</title>
  </head>
  <body>
    <h1>Mein erstes HTML</h1>
    <p>Das hier ist mein erstes selbst geschriebenes HTML.</p>
  </body>
</html>

```

Abbildung 4.12 Wir haben die einzelnen HTML-Elemente hier unterschiedlich weit nach rechts eingerückt, um zu veranschaulichen, woraus ihr Inhalt besteht.

Von oben nach unten und von links nach rechts gelesen, besteht Ihr erstes HTML-Dokument also aus folgenden Elementen:

- ▶ dem `html`-Element, das das gesamte Dokument von der ersten bis zur letzten Zeile umfasst
- ▶ dem `head`-Element, das Informationen für Ihren Browser enthält. Dazu gehört das `title`-Element mit dem Titel der Seite.

- ▶ dem `body`-Element, das den Großteil der Seitenelemente enthält, die der Nutzer sehen kann. Dazu gehören ein `h1`-Element (die Überschrift der Seite) sowie ein `p`-Element mit einem Absatz Text.

Wenn ein Element andere Elemente enthält, wie hier etwa das `html`-Element, das das `head`- und das `body`-Element umschließt, ist es ein *Elternelement*. Die darin enthaltenen Elemente werden als *Kindelemente* bezeichnet. Das `head`-Element zum Beispiel ist also ein Kindelement des `html`-Elements, während es gleichzeitig aber auch das Elternelement des `title`-Elements ist.

### Element oder Bereich?

Im Zusammenhang mit HTML hören Sie auch oft den Begriff *Bereich*, der sich auf bestimmte Elemente bezieht. Der *Kopfbereich* einer Seite ist zum Beispiel nichts anderes als das `head`-Element inklusive Inhalt.

Nun, da Sie mit dem grundsätzlichen Aufbau eines HTML-Dokuments vertraut sind, betrachten Sie noch einmal den Quelltext der Startseite unseres Beispielprojekts, der in Abbildung 4.13 zu sehen ist.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Aro Argentino - Startseite</title><meta property="og:site_name" content="Aro Argentino" />
5 <meta property="og:title" content="Aro Argentino" />
6 <meta property="og:description" content="Wir sind eine Traditionstanzschule in Herzen von München. In der schönen
7 <meta property="og:image" content="http://aroargentino.weebly.com/uploads/7/7/0/4/77049043/2537749_orig.png" />
8 <meta property="og:image" content="http://aroargentino.weebly.com/uploads/7/7/0/4/77049043/2503761_orig.png" />
9 <meta property="og:image" content="http://aroargentino.weebly.com/uploads/7/7/0/4/77049043/7996872_orig.png" />
10 <meta property="og:url" content="http://aroargentino.weebly.com/" />
11
12
13
14 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
15 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
16
17 <link id="wsite-base-style" rel="stylesheet" type="text/css" href="//cdn2.editsite.com/css/sites.css?
18 buildTime=1464384598" />
19 <link rel="stylesheet" type="text/css" href="//cdn1.editsite.com/editor/libraries/fancybox/fancybox.css?1459287531" />
20 <link rel="stylesheet" type="text/css" href="//cdn2.editsite.com/css/social-icons.css?buildTime=1459287531"
21 media="screen,projection" />
22 <link rel="stylesheet" type="text/css" href="/files/main_style.css?1459352820" title="wsite-theme-css" />
23 <link href="//fonts.googleapis.com/css?family=Oxygen:400,300,700&subset=latin,latin-ext" rel="stylesheet" type="text/css" />
24 <link href="//fonts.googleapis.com/css?family=Pacifico&subset=latin,latin-ext" rel="stylesheet" type="text/css" />
25 <style type="text/css">
26 .wsite-elements.wsite-not-footer div.paragraph, .wsite-elements.wsite-not-footer p, .wsite-elements.wsite-not-footer
27 .product-block .product-title, .wsite-elements.wsite-not-footer .product-description, .wsite-elements.wsite-not-footer
28 .wsite-form-field label, .wsite-elements.wsite-not-footer .wsite-form-field label, #wsite-content div.paragraph, #wsite-
29

```

Abbildung 4.13 Im Quelltext der Beispielseite lassen sich jetzt bekannte HTML-Elemente ausmachen.

Der Quelltext kommt Ihnen wahrscheinlich immer noch unübersichtlich und etwas kryptisch vor. Aber mit Ihrem neuen Wissen erkennen Sie jetzt, dass es sich bei dem Quelltext um ein HTML-Dokument handelt: Es besteht aus Elementen mit Start-Tags, Elementinhalten und End-Tags. Bereits bekannt sind Ihnen ❶ das `html`-Element und das `head`-Element, deren Start-Tags Sie in Zeile 2 und 3 des Quelltexts sehen, sowie das `title`-Element in Zeile 4.

HTML dient also dazu, den Aufbau einer Webseite zu strukturieren: Indem Sie definierte Elemente in einer bestimmten Reihenfolge schreiben, teilen Sie dem Browser zum Beispiel mit, dass auf der Seite zuerst eine Überschrift und dann ein Textabsatz oder auch ein Bild angezeigt werden soll.

## 4.4 Woher kommt die Gestaltung?

Aber woher weiß der Browser, wie genau er diese Elemente grafisch darstellen soll? Woher weiß er, wie viel größer die Schriftgröße einer Überschrift ist oder welche Farbe der Text überhaupt haben soll?

Um Browsern solche Gestaltungsanweisungen zu geben, gibt es drei unterschiedliche Methoden:

- ▶ style-Attribute
- ▶ das style-Element
- ▶ externe CSS-Dateien

### 4.4.1 Das style-Attribut

HTML-Elemente können außer ihren Tags und ihrem Inhalt auch *Attribute* besitzen, die dem Browser weitere Informationen über das Element geben. Attribute bestehen aus einem *Namen* und einem *Wert*, wobei dazwischen ein Ist-gleich-Zeichen und um den Wert herum Anführungsstriche geschrieben werden müssen. Attribute werden immer hinter dem Elementnamen in dessen Start-Tag geschrieben. Sie sehen das Prinzip in Abbildung 4.14.



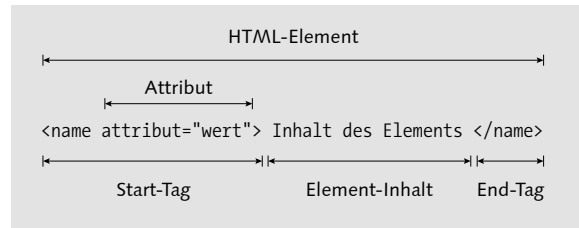


Abbildung 4.14 Der Aufbau eines HTML-Elements mit Attribut.

Es gibt viele unterschiedliche Element-Attribute, die unterschiedliche Zwecke erfüllen. Um dem Browser Gestaltungsvorgaben für ein HTML-Element zu geben, verwendet man das *style-Attribut*. Damit legen wir jetzt neue Schriftgrößen für das h1- und das p-Element auf unserer Seite fest.

**Schritt 1** | Fügen Sie Ihrem Überschrift-Element h1 ein Attribut mit dem Namen style und dem Wert font-size:64px; hinzu.

Der Start-Tag des Überschriften-Elements sieht jetzt so aus:

```
<h1 style="font-size:64px;">
```

**Schritt 2** | Fügen Sie auch Ihrem Textabsatz-Element p ein style-Attribut hinzu, allerdings mit dem Wert font-size:32px;:

```
<p style="font-size:32px;">
```

Ihr HTML-Dokument sollte im Texteditor jetzt so aussehen wie in Abbildung 4.15.



Abbildung 4.15 Mit den »style«-Attributen machen wir für das »h1«- und das »p«-Element eigene Gestaltungsvorgaben.

Damit haben Sie dem h1- und dem p-Element in Ihrem Dokument eine font-size (engl. für *Schriftgröße*) von 64 bzw. 32 Pixeln zugewiesen, die der Browser von nun an bei der grafischen Darstellung berücksichtigt. Öffnen Sie Ihre Seite im Browser (oder laden Sie die Seite neu, wenn sie bereits geöffnet ist), um die Auswirkungen des Attributs zu sehen (siehe Abbildung 4.16).



Abbildung 4.16 Dank des »style«-Attributs werden Überschrift und Absatztext jetzt doppelt so groß angezeigt wie vorher.

Der Text wird doppelt so groß dargestellt wie beim ersten Betrachten Ihrer HTML-Seite, als Sie noch kein style-Attribut zur Formatierung der Elemente angegeben hatten.

Mithilfe von Attributen können Sie also die Gestaltung eines einzelnen Elements festlegen, die der Browser dann bei der grafischen Darstellung Ihrer Webseite umsetzt. ■

#### 4.4.2 Das style-Element

Die meisten Webseiten enthalten aber nicht nur zwei sichtbare Elemente wie unser Beispieldokument. Oft enthalten sie so viel Text oder Bilder, dass der Besucher die Seite nach unten scrollen muss, um alles sehen zu können. Entsprechend lang ist bei einer solchen Seite auch das HTML-Dokument in Ihrem Texteditor, und entsprechend weit sind die vielen verschiedenen style-Attribute der Seitenelemente verstreut.

Das nervt spätestens dann, wenn man bei einer bereits öffentlich zugänglichen Seite nur noch ab und zu ein Detail der Gestaltung ändern will. Wenn man dann nicht wegen jeder Änderung einer Schriftgröße das ganze HTML-Dokument nach dem richtigen Attribut absuchen will, kann man ein Element verwenden, das alle Gestaltungsanweisungen für den Browser übersichtlich im oberen Dokumentbereich sammelt: das style-Element, das wir jetzt ausprobieren.

Wenn man ein style-Element in seinem HTML-Dokument nutzt, muss es immer ein Kindelement des head-Elements sein.

**Schritt 1** | Schreiben Sie also nach dem `title`-Element, das ja ebenfalls ein Kind des `head`-Elements ist:

```
<style type="text/css">
</style>
```

Weil es viele unterschiedliche Arten von Gestaltungsvorgaben gibt, spezifiziert man mit dem Attribut `type` für den Browser, um was für eine Art Gestaltungsvorgabe es sich bei diesem Element handelt.

Bislang haben wir aber nur den Start- und End-Tag des `style`-Elements. Der Inhalt besteht aus sogenannten *Formaten*, die HTML-Elementen ihre Gestaltungsvorgaben zuweisen. Ein Format hat eine fest vorgegebene Syntax, also Schreibweise, die Sie in Abbildung 4.17 sehen können.



**Abbildung 4.17** Die Syntax eines Formats, das Gestaltungsvorgaben für HTML-Elemente enthält

#### Alternative Begriffe

Statt *Format* wird auch der Begriff *Regel* benutzt, statt *Eigenschaft* auch der Begriff *Befehl*.

Ein Format besteht also zuerst aus einem *Selektor*. Dieser gibt an, für welches HTML-Element das Format gilt. Denn wenn Gestaltungsvorgaben nicht mehr per Attribut in dem betreffenden HTML-Element selbst enthalten sind, muss der Browser ja irgendwoher wissen, welches Element Sie gestalten wollen.

Auf den Selektor folgen die Gestaltungsvorgaben in Form von Eigenschaft-Wert-Paaren, die von geschweiften Klammern umfasst werden. Innerhalb dieser Klammern können Sie so viele Eigenschaft-Wert-Paare notieren, wie Sie wollen. Wichtig ist hier nur, dass bei jedem Paar nach der Eigenschaft ein Doppelpunkt `:` und nach dem Wert ein Semikolon `;` folgt.

Versehen wir das neue `style`-Element in Ihrem Dokument also mit Formaten für die Überschrift und das Absatz-Element:

**Schritt 2** | Legen Sie dazu im `style`-Element ein Format mit dem Selektor `h1` an.

**Schritt 3** | Diesem geben Sie die Eigenschaft `font-style` mit dem Wert `italic`.

#### Die richtige Taste

Auf Windows-Tastaturen geben Sie geschweifte Klammern ein, indem Sie `[Alt Gr] + [7]` (öffnende Klammer) bzw. `[Alt Gr] + [0]` (schließende Klammer) drücken. Auf Mac-Tastaturen drücken Sie `[Alt] + [8]` bzw. `[Alt] + [9]`.

**Schritt 4** | Legen Sie ein zweites Format mit dem Selektor `p` an.

**Schritt 5** | Dieses Format bekommt die Eigenschaft `font-weight` mit dem Wert `bold`.

Das `style`-Element in Ihrem HTML-Dokument sollte nun so aussehen wie in Abbildung 4.18.

```
<html>
<head>
<title>HTML</title>
<style type="text/css">
h1 {font-style: italic;}
p {font-weight: bold;}
</style>
</head>
<body>
<h1 style="font-size:64px;">Mein erstes HTML</h1>
<p style="font-size:32px;">Das hier ist mein erstes selbst geschriebenes HTML.</p>
</body>
</html>
```

**Abbildung 4.18** Ein HTML-Dokument mit einem »style«-Element, das zwei Formate enthält

**Schritt 6** | Um das Ergebnis zu testen, speichern Sie Ihr Dokument und öffnen es ein weiteres Mal im Browser (siehe Abbildung 4.19).



**Abbildung 4.19** Der Browser befolgt die Gestaltungsvorgaben aus dem »style«-Element.

Die Überschrift wird jetzt in Schrägschrift, der Textabsatz in Fettschrift angezeigt. Die Eigenschaften `font-style` und `font-weight` bestimmen also – wie das bereits bekannte `font-size` – die Gestaltung der Schrift bei Textelementen. ■

Anhand der Schriftgrößen von Überschrift und Absatztext erkennen Sie auch, dass die Gestaltungsanweisungen aus den Elementattributen immer noch umgesetzt werden. Ein Dokument kann also Anweisungen in Attributen und in einem `style`-Element enthalten, und der Browser berücksichtigt beide Arten gleichzeitig.

Das `style`-Element macht also besonders das Pflegen einer einzelnen HTML-Seite einfacher, wenn sie bereits veröffentlicht wurde und nur noch ab und zu Details an der Gestaltung geändert werden sollen.

Die meisten Websites bestehen aber aus mehreren Seiten. Wollte man sie alle per `style`-Element gestalten, dann müsste man in das `head`-Element jeder einzelnen Seite ein `style`-Element mit denselben Formaten einfügen. Bei Änderungen müssten Sie die Formate jeder Seite einzeln aktualisieren. Glücklicherweise gibt es auch hier eine Lösung, die Ihnen viel Arbeit erspart.

#### 4.4.3 Die externe CSS-Datei

Sie können einfach die Formate für alle Ihre HTML-Seiten in eine neue Datei auslagern, anstatt sie in jedem Dokument in ein separates `style`-Element zu schreiben. Damit der Browser auch weiß, dass er in dieser Datei nach Gestaltungsvorgaben suchen soll, verweisen Sie einfach in jedem HTML-Dokument darauf.

Die externe Datei für die Formate nennt man *CSS-Datei*, weil Formate für HTML-Elemente in der Sprache CSS geschrieben werden.

##### Was bedeutet die Abkürzung CSS?

CSS steht für *Cascading Style Sheets*, übersetzt etwa *kaskadierende Formatvorlagen*. Formatvorlagen kennen Sie von Programmen wie Word; *kaskadieren* klingt für deutsche Ohren dagegen etwas befremdlich. »Cascading« bedeutet hier sinngemäß so viel wie »abstufend« oder »vererbend«, also »in eine bestimmte Reihenfolge bringend«.

Neben den möglichen Eigenschaften (wie etwa `font-size`), die man HTML-Elementen zuweisen kann, und der Syntax der Formate beinhaltet die Sprache CSS zwei wichtige Dinge, ohne die man Struktur und Gestaltung einer Seite nicht trennen kann: das Zuweisen von Formaten zu HTML-Elementen mithilfe von Selektoren sowie ein Regelwerk zur Bestim-

mung des richtigen Formats, wenn einem Element mehrere Formate zugewiesen werden. Damit in der CSS-Datei zusätzlicher Platz gespart werden kann, bestimmen die CSS-Regeln außerdem das Vererben von Formaten vom Elternelement an das Kindelement.

Eine CSS-Datei lässt sich mit demselben einfachen Texteditor erstellen wie das HTML-Dokument und enthält nichts außer den bereits aus dem `style`-Element bekannten Formaten.

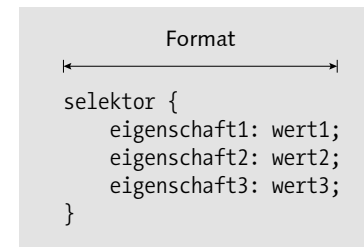
**Schritt 1** | Löschen Sie zuerst das `style`-Element und die `style`-Attribute aus Ihrem HTML-Dokument.

**Schritt 2** | Prüfen Sie durch Speichern und erneutes Öffnen im Browser, ob alle Gestaltungsvorgaben gelöscht sind, die Sie bisher gemacht haben. Es darf jetzt keine (gegenüber der Standard-Darstellung des Browsers) veränderte Schriftgröße, keine Schräg- und Fettschrift mehr zu sehen sein.

**Schritt 3** | Erstellen Sie mit Ihrem Texteditor eine neue Datei.

**Schritt 4** | Fügen Sie ein Format mit dem Selektor `h1` ein, das zwei Eigenschaften enthält: `font-size` mit dem Wert `64px` sowie `font-style` mit dem Wert `italic`.

Bei mehreren Eigenschaften in einem Format empfiehlt sich die Schreibweise aus Abbildung 4.20.



```
Format  
selektor {  
  eigenschaft1: wert1;  
  eigenschaft2: wert2;  
  eigenschaft3: wert3;  
}
```

**Abbildung 4.20** Die Verwendung einer neuen Zeile je Eigenschaft-Wert-Paar sorgt für Übersicht.

**Schritt 5** | Fügen Sie ein Format mit dem Selektor `p` ein, das ebenfalls zwei Eigenschaften enthält: `font-size` mit dem Wert `32px` sowie `font-weight` mit dem Wert `bold`.

Vergessen Sie die Semikola (Strichpunkte) am Ende jedes Eigenschaft-Wert-Paares nicht, sonst ignoriert der Browser Ihre Anweisungen. Das Dokument im Texteditor sollte jetzt so aussehen wie in Abbildung 4.21.

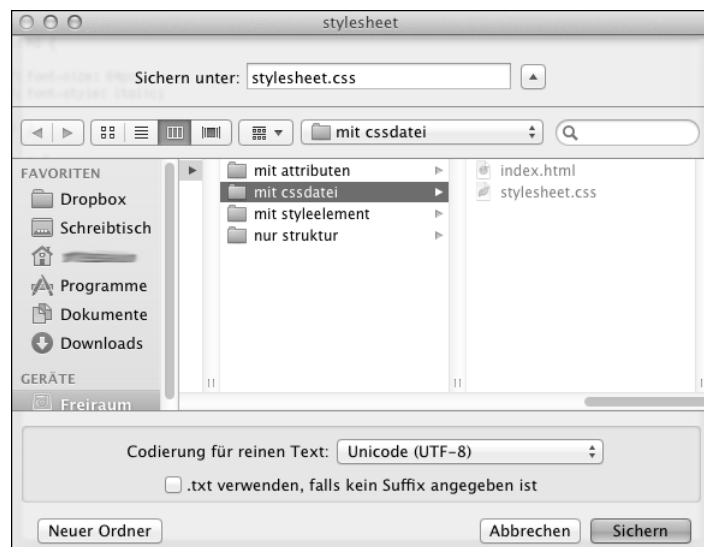


**Abbildung 4.21** Zur besseren Übersicht schreiben wir jedes Eigenschaft-Wert-Paar in eine neue Zeile.

**Schritt 6** | Speichern Sie die Datei unter dem Namen *stylesheet.css* ab.

**Schritt 7 | Nur Editor:** Beachten Sie dazu dieselben Hinweise wie beim Speichern der HTML-Datei. Wählen Sie also im SPEICHERN UNTER-Fenster bei DATEITYP die Option ALLE DATEIEN (\*.\*) und außerdem bei CODIERUNG die Option UTF-8, bevor Sie den Dateinamen und die Dateierweiterung eingeben (mehr zur Zeichencodierung folgt in Abschnitt 4.6.4).

Sie können auch einen anderen Namen wählen, müssen diesen dann aber auch verwenden, wenn Sie den Verweis in das HTML-Dokument einbinden (siehe Schritt 8). Als Speicherort wählen Sie dasselbe Verzeichnis, in dem auch Ihr HTML-Dokument liegt (wie in Abbildung 4.22 zu sehen ist).



**Abbildung 4.22** Speichern Sie die CSS-Datei im selben Verzeichnis wie Ihr HTML-Dokument.

Jetzt müssen Sie im HTML-Dokument noch auf die CSS-Datei verweisen, sonst weiß der Browser nicht, woher er die Formate für die Elemente nehmen soll.

Dazu benutzen wir ein neues Element im HTML-Dokument: das `link`-Element. Man verwendet es immer dann, wenn man dem Browser mitteilen möchte, dass er für die Verarbeitung dieses HTML-Dokuments Informationen aus anderen Dateien benötigt.

Ein typisches `link`-Element hat die Attribute `rel`, `type` und `href`. Die ersten beiden Attribute geben die Art der Informationen an, die in der anderen Datei enthalten sind, und das `href`-Attribut gibt den Speicherort der Datei an.

**Schritt 8** | Um also auf die CSS-Datei hinzuweisen, fügen Sie Folgendes in das `head`-Element Ihrer HTML-Datei ein:

```
<link rel="stylesheet" type="text/css" href="stylesheet.css">
```

Wenn Sie im letzten Schritt die CSS-Datei unter einem anderen Namen gespeichert haben, verwenden Sie im Wert des `href`-Attributs jetzt natürlich den von Ihnen gewählten Dateinamen.

Hier stoßen wir auf etwas Neues: ein Element, das keinen gesonderten Start- und End-Tag besitzt. Stattdessen besteht es aus einem einzigen Tag, der vorne mit einer spitzen Klammer geöffnet wird, den Elementnamen und einige Attribute enthält und dann hinten durch eine spitze Klammer geschlossen wird.

Wenn Sie Ihre HTML-Seite das nächste Mal im Browser betrachten, wird dieser die im `link`-Element angegebene CSS-Datei nach Formaten durchsuchen. In unserem Beispiel findet er darin Gestaltungsvorgaben für zwei Elemente: für das `h1`- und das `p`-Element. Also wird er diese Elemente entsprechend gestalten, und zwar genau so, als hätten Sie die beiden Elemente im HTML-Dokument mit einem `style`-Attribut versehen.

**Schritt 9** | Um den Effekt der verknüpften CSS-Datei zu testen, speichern Sie das HTML-Dokument und öffnen es erneut im Browser.

Die Seite sollte jetzt wieder so aussehen wie in Abbildung 4.19, mit denselben Schriftgrößen und Schriftstilen. ■

#### Unsere Beispieldateien zum Download

Die in diesem Kapitel erstellten HTML- und CSS-Dateien finden Sie unter [bnfr.de/ql401](http://bnfr.de/ql401) auf der Website zum Buch.

Dank der externen CSS-Datei haben Sie mehr Übersicht und weniger Arbeit, wenn Sie mit komplexeren Websites und mehreren Unterseiten arbeiten.

Es bleibt aber noch eine Frage: Als Sie Ihre HTML-Seite zum allerersten Mal mit dem Browser betrachtet haben, hatten Sie davor weder ein `style`-Attribut noch ein `style`-Element noch eine CSS-Datei festgelegt. Warum hat der Browser das `h1`-Element trotzdem in einer größeren Schriftart dargestellt als das `p`-Element?

Die Antwort lautet: Der Hersteller des Browsers hat dem Programm Standard-CSS-Dateien mitgegeben, in denen Gestaltungsanweisungen für die meisten HTML-Elemente enthalten sind. Damit sorgt der Hersteller dafür, dass sein Browser jede Webseite zumindest gut lesbar darstellt, auch wenn der Autor der Webseite keine Gestaltungsvorgaben gemacht hat. Die browsereigenen CSS-Dateien finden Sie teilweise in dessen Programmverzeichnis, wenn Sie neugierig sind.

#### Verwenden Sie immer eine externe CSS-Datei

Sie haben in diesem Abschnitt drei Möglichkeiten zur Formatierung von HTML kennengelernt: das `style`-Attribut, das `style`-Element und die externe CSS-Datei. Wir raten Ihnen, immer die dritte Möglichkeit zu nutzen. In einer externen Datei haben Sie immer alle Formate für Ihre Site auf einen Blick. Wenn Sie Änderungen an der Gestaltung vornehmen, müssen Sie dann nicht umständlich jede Unterseite einzeln anpassen.

Weil Sie aber auch die anderen beiden Methoden der Gestaltungsvorgabe kennengelernt haben, können Sie den Quelltext von Webseiten verstehen, die `style`-Attribute oder das `style`-Element verwenden.

Damit beherrschen Sie schon die Grundlagen von HTML und CSS und wissen, wie man eine Webseite selbst schreiben kann. Damit Ihnen das noch besser gelingt, zeigen wir Ihnen als Nächstes Programme für Einsteiger, die etwas mehr Komfort bieten als die Standard-Texteditoren Ihres Betriebssystems. In den weiteren Abschnitten des Kapitels geben wir Ihnen außerdem Tipps dazu, wie Sie Ihre Seiten so schreiben, dass sowohl der Browser als auch Sie selbst sie immer verstehen.

Mit diesem Wissen werden Sie dann gut gewappnet sein für das nächste Beispielprojekt in Kapitel 5, »Die eigene Website erstellen«, bei dem Sie eine vollständige Website aus HTML und CSS erstellen. Im Laufe des Beispielprojekts werden wir Sie Schritt für Schritt in die vielen HTML-Elemente und CSS-Eigenschaften einführen, die man für eine richtige Webseite braucht. Bis Ihre Websites genauso aufwendig gestaltet sind wie etwa die mit einem Blogsystem erstellte Beispielsite *Aro Argentino*, wird es zwar noch eine Weile dauern. Aber in den nächsten Kapiteln eignen Sie sich nach und nach die dafür notwendigen Kenntnisse an.

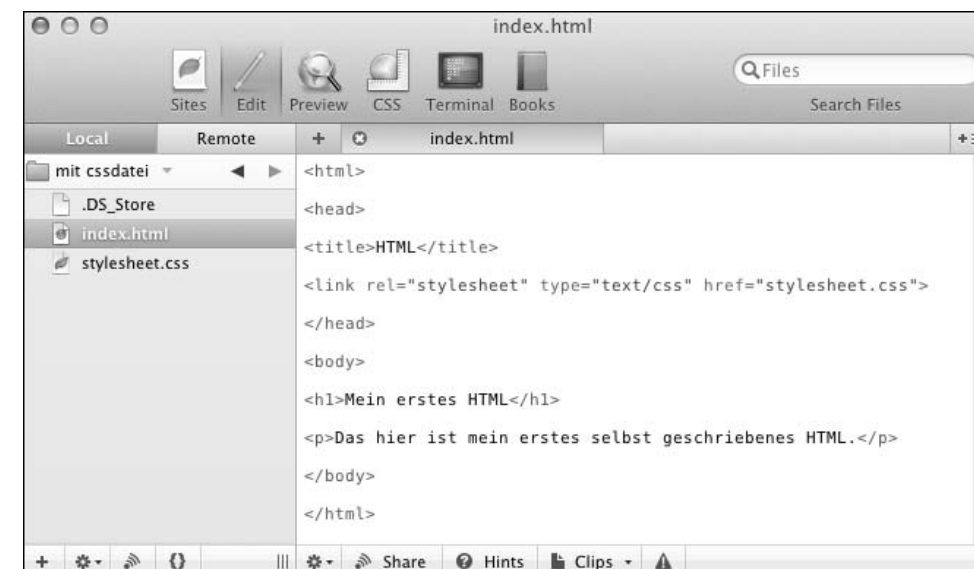
## 4.5 Das richtige Werkzeug – der HTML-Editor

Sie könnten alle Seiten des nächsten Beispielprojekts mit dem simplen Texteditor schreiben, den Sie bis jetzt genutzt haben. Es gibt aber spezielle *HTML-Editoren*, die Ihnen die Arbeit erleichtern. Und weil schon kostenlose Programme Vorteile bieten, empfehlen wir Ihnen auch, einen HTML-Editor zu nutzen.

### 4.5.1 Was sollte ein HTML-Editor für Einsteiger können?

Um sich für einen geeigneten HTML-Editor entscheiden zu können, sollten Sie erst einmal wissen, welche Funktionen in der Praxis den Arbeitskomfort erhöhen. Dabei geht es nicht nur um Bequemlichkeit. Denn wenn der Editor Ihnen die Arbeit erleichtert, machen Sie weniger Fehler und müssen seltener nachbessern.

Eine wichtige Funktion, die das Bearbeiten von HTML-Dokumenten gerade für Anfänger sehr viel übersichtlicher macht, ist die farbige Markierung einzelner HTML-Bausteine, die Sie in Abbildung 4.23 am Beispiel des Editors *Coda* sehen.



**Abbildung 4.23** Die Datei »DS\_Store«, die Sie im Dateibrowser links sehen, hat nichts mit der Website zu tun. Sie ist eine automatisch angelegte Systemdatei von OS X.

So werden im Editor zum Beispiel die Start- und End-Tags von HTML-Elementen, die Attributnamen, die Attributwerte und der Inhalt von Textelementen wie `h1` oder `p` in unterschiedlichen Farben angezeigt.

Diese farbige Hervorhebung hilft Ihnen besonders bei längeren Dokumenten, den Aufbau schnell zu erkennen und eventuelle Fehler schneller zu finden.

Eine zweite nützliche Eigenschaft eines HTML-Editors ist die sogenannte *Debugging*-Funktion. *Debugging* ist das Aufspüren und Entfernen von Fehlern aus Computersoftware aller Art. (Auf Deutsch hieße *Debugging* »Entwanzen«. Im Englischen werden Fehler im Softwarebereich *bugs* genannt, was auch *Käfer* oder *Wanze* bedeutet.)

Die häufigsten Fehler sind Syntaxfehler, also falsche Schreibweisen. So kann es durchaus vorkommen, dass man stundenlang einen Fehler in seiner Seite gesucht hat, nur weil irgendwo eine spitze Klammer oder ein Anführungszeichen fehlte! Die Debugging-Funktion des HTML-Editors markiert in Ihrem Dokument automatisch solche Fehler, in Abbildung 4.24 beispielsweise einen falsch geschriebenen Attributnamen.

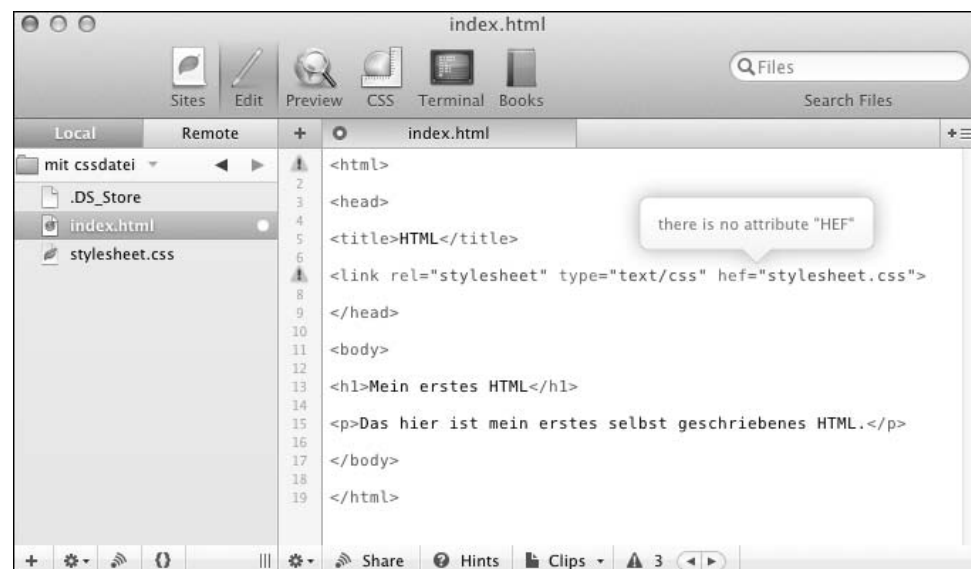


Abbildung 4.24 Ein einfacher Tippfehler kann die gesamte Seite unbrauchbar machen. Die Debugging-Funktion eines HTML-Editors hilft, solche Fehler aufzuspüren.

#### Debugging auch ohne Editor

Die Debugging-Funktion ist etwas aufwendiger zu programmieren und findet sich deshalb nicht in allen kostenlosen Editoren. Wenn ein HTML-Editor keine Debugging-Funktion hat, können Sie Ihr HTML-Dokument aber problemlos mit dem W3C-Validator unter [bnfr.de/ql402](http://bnfr.de/ql402) auf Fehler überprüfen. (Siehe dazu auch Abschnitt 4.6.7, »Validieren Sie Ihre Seiten«.)

#### 4.5.2 Empfehlenswerte HTML-Editoren

Es gibt Hunderte HTML-Editoren auf dem Markt. Egal ob für Windows-PC oder Mac, Sie haben die Wahl zwischen kostenlosen und kostenpflichtigen Programmen mit ganz unterschiedlichem Funktionsumfang. Wir haben bewusst möglichst einfache Programme ausgewählt, die Einsteiger nicht durch unnötige Zusatzfunktionen ablenken. Wenn Sie mit Ihren ersten Webprojekten ein wenig Übung haben, können Sie später umfangreichere und noch komfortablere Editoren ausprobieren. Aber nun geht es erst einmal darum, zu lernen, wie man alles von Grund auf anlegt. Wenn Sie das können, können Sie sich einen Teil der Routinearbeit abnehmen lassen.

Wir arbeiten in diesem Buch mit *Coda* für Mac, aber welches Programm Sie wählen, ist ganz Ihren persönlichen Vorlieben überlassen. Unsere Beispielprojekte sind mit jedem der hier vorgestellten HTML-Editoren durchführbar.

#### gedit

- ▶ Download unter: [bnfr.de/ql403](http://bnfr.de/ql403)
- ▶ Betriebssystem: Windows, Mac, Linux
- ▶ Sprache: Deutsch (Benutzeroberfläche), Englisch (Download, Installation)
- ▶ Preis: Für alle Projekte kostenlos

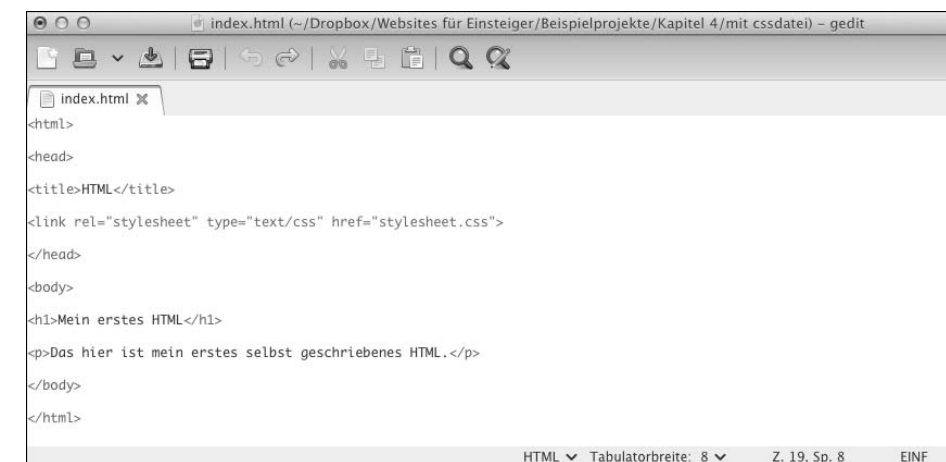


Abbildung 4.25 Nicht mehr, als man braucht: »gedit« für Windows, Mac und Linux

- ▶ **Funktionen:** Farbige Markierung der HTML-Elemente
- ▶ **Wichtige Einstellungen:** Keine

- **Empfehlung:** Wenn die englischsprachige Anbieterwebsite und (trotzdem simple) Installation für Sie keine Hürden sind, ist *gedit* der perfekte Editor für HTML-Anfänger: Er hat nicht einen einzigen überflüssigen Button, der Sie vom Schreiben Ihrer HTML- und CSS-Zeilen ablenken würde.

### Notepad++

- Download unter: [bnfr.de/ql404](http://bnfr.de/ql404)
- Betriebssystem: Windows
- Sprache: Deutsch (Benutzeroberfläche), Englisch (Download, Installation)
- Preis: Für alle Projekte kostenlos

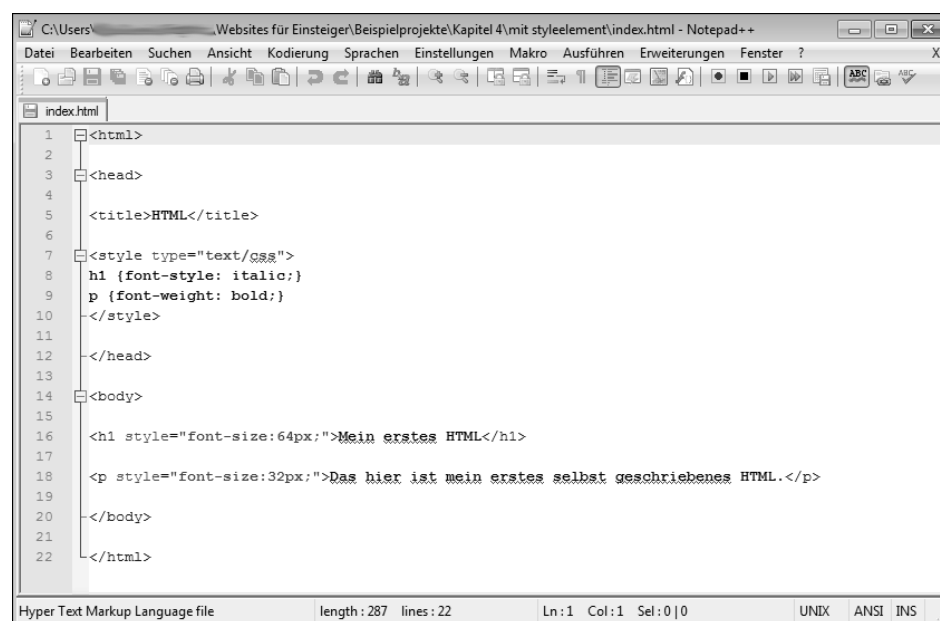


Abbildung 4.26 Mehr Funktionen, weniger Übersicht: »Notepad++« für Windows

- **Funktionen:** Farbige Markierung der HTML-Elemente
- **Wichtige Einstellungen:** Da das Programm neben HTML viele andere Sprachen beherrscht, sollten Sie in den Einstellungen die Optionen HTML für die Standardsprache, UTF-8 für die Kodierung und WINDOWS für das Zeilenende auswählen.
- **Empfehlung:** *Notepad++* ist ein Multitalent für Dutzende Programmiersprachen. Deshalb hat dieser Editor mehr Funktionen. Die allermeisten werden Sie nicht brauchen, sie können anfangs aber vielleicht ablenken. Eine gute und für jedes Projekt

kostenlose Alternative ist das Programm trotzdem. Eine schlankere Alternative für Windows, die aber nicht mehr weiterentwickelt wird, ist Programmer's Notepad ([pnotepad.org](http://pnotepad.org)).

### Phase 5

- Download unter: [bnfr.de/ql405](http://bnfr.de/ql405)
- Betriebssystem: Windows
- Sprache: Deutsch (Download, Installation, Benutzeroberfläche)
- Preis: Für private Sites kostenlos, für kommerzielle Sites ca. 50 EUR

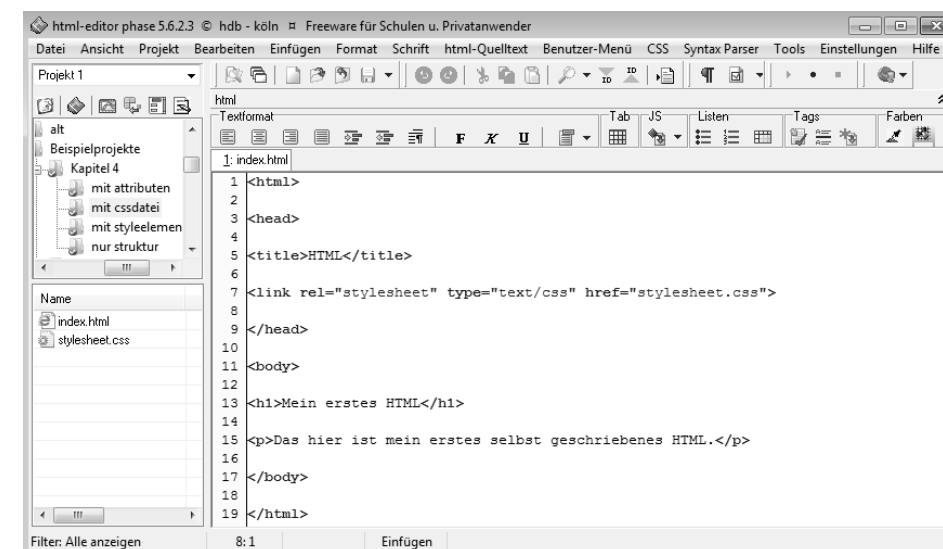


Abbildung 4.27 Viele Funktionen und komplett in Deutsch: Phase 5 für Windows

- **Funktionen:** Farbige Markierung der HTML-Elemente, Debugging
- **Wichtige Einstellungen:** Keine
- **Empfehlung:** Wenn man von der Herstellerwebsite bis zum letzten Button wirklich alles komplett in Deutsch braucht und mit Windows arbeitet, ist Phase 5 eine Option. Das Programm ist aber nur für Privatanwender kostenlos, die nicht-kommerzielle Websites erstellen. Firmen, Selbstständige oder auch Vereine müssen eine Unternehmenslizenz für rund 50 Euro kaufen. Auch hatten wir Probleme mit Umlauten in HTML-Dateien, die mit anderen Editoren erstellt und dann in Phase 5 geöffnet wurden. Am besten testen Sie zuerst die kostenlose Version, ob bei Ihnen alles wie erwartet klappt.

## Coda 2

Download unter: [bnfr.de/ql406](http://bnfr.de/ql406)

Betriebssystem: Mac

Sprache: Englisch (Download, Installation, Benutzeroberfläche)

Preis: ca. 100 USD



Abbildung 4.28 Aufgeräumte Oberfläche trotz großem Funktionsumfang: »Coda 2« für den Mac

- **Funktionen:** Farbige Markierung der HTML-Elemente, Debugging
- **Wichtige Einstellungen:** Keine
- **Empfehlung:** Die Autoren dieses Buches verwenden Coda. Gerade im Vergleich zu den Open-Source-Lösungen auf dem Mac empfinden wir die Bedienoberfläche als schöner und das Programm insgesamt als komfortabler. Wenn Sie mit dem Mac arbeiten, Englisch Sie nicht abschreckt und Sie mehr als eine oder zwei Websites im Jahr erstellen, ist das Geld unserer Meinung nach gut angelegt.

## 4.6 Besser lesbares HTML für Browser und Mensch

Es gehört mehr zu einem gut geschriebenen HTML-Dokument als fehlerfreie Syntax und übersichtliche Anordnung: Erstens braucht der Browser sogenannte Metainformationen, damit er Ihre Seite so darstellen kann, wie Sie sich das wünschen. Zweitens soll-

ten Sie, wenn Sie das HTML-Dokument nach längerer Zeit wieder betrachten, möglichst schnell verstehen, was Sie da vor sich haben und wie die Seiten funktionieren.

Damit Sie ein besseres Verständnis für das Verhältnis zwischen Ihnen, HTML und dem Browser erhalten, klären wir erst einmal die Frage, woher HTML und CSS überhaupt kommen. Für Ihre ersten selbst geschriebenen Zeilen war das noch nicht wichtig, aber für die zukünftige Arbeit mit komplexeren Seiten brauchen Sie etwas Hintergrundwissen.

### 4.6.1 Woher kommen HTML und CSS?

Schon vor der Erfindung von HTML gab es Sprachen zur Auszeichnung von Textdokumenten (und auch das Internet, also vernetzte Computer, gab es schon vorher). Aber 1989 machte ein am Forschungszentrum CERN angestellter Physiker, Tim Berners-Lee, den entscheidenden Schritt: Er schlug vor, zum besseren Austausch der Forschungsergebnisse ein System vernetzter Hypertextdokumente zu verwenden. Das CERN war noch nicht interessiert genug, um Berners-Lee mit der Entwicklung zu beauftragen, also begann er eigenständig mit der Umsetzung. Aus Diskussionen mit Gleichgesinnten entstanden 1991 der erste Vorschlag für die Sprache HTML und die ersten Elemente. Berners-Lee programmierte nach eigenen Angaben auch den ersten HTML-Browser und den ersten HTML-Webserver.

Die Sprache war für jedermann frei zugänglich, sodass schon bald viele Menschen daran mitarbeiteten. Um die Weiterentwicklung von HTML unter einem Dach koordinieren zu können, gründete Tim Berners-Lee 1994 mit anderen das *World Wide Web Consortium (W3C)*. Die Mitglieder entwerfen nicht nur neue Elemente oder Attribute, sondern legen auch fest, was die korrekte Schreibweise für HTML ist – man spricht auch von *validem HTML*.

In dieser Hinsicht ist das W3C ein bisschen wie die Redaktion des deutschen Dudens: Millionen Menschen benutzen täglich eine Sprache, und das W3C gibt regelmäßig Erklärungen dazu ab, wie man korrekt schreibt.

HTML wurde von immer mehr verschiedenen Personen und Organisationen eingesetzt, und diese hatten natürlich unterschiedliche Vorstellungen davon, wie eine Webseite gestaltet sein sollte – welche Schrift ist schöner, welche Farbe soll ein Link haben usw. Um einerseits diesen Menschen mehr Kontrolle über die Gestaltung Ihrer Seiten zu geben und andererseits HTML unabhängig von der Gestaltung weiterentwickeln zu können, wurde die Verwendung von *Stylesheets* vorgeschlagen, zu Deutsch etwa *Stilblätter*. Das sind Dokumente, die ausschließlich Gestaltungsvorgaben für die Darstellung anderer Dokumente enthalten und es damit möglich machen, Struktur und Ge-



staltung einer Seite voneinander zu trennen. 1996 erklärte das W3C dann die Stylesheet-Sprache CSS zur Standard-Gestaltungssprache seiner HTML-Dokumente.

Bis heute sind sowohl HTML als auch CSS frei zugängliche und lizenzfreie Sprachen, deren Weiterentwicklung öffentlich einsehbar ist. Sie dürfen die Sprachen für alle Web-Projekte benutzen, ohne dafür Lizenzgebühren zahlen oder Copyright-Vermerke anbringen zu müssen. Ohne die von Anfang an gewollte Offenheit hätte sich das Internet auf der Basis von HTML nie so schnell entwickeln können.

#### 4.6.2 HTML5 – was bedeutet das?

Mit der Zeit kamen neue Seitenelemente, Attribute und CSS-Eigenschaften hinzu, und alte wurden gestrichen. Diese Entwicklung wird vom W3C mit Versionsnummern festgehalten. HTML5 und CSS3 sind die aktuellen Versionen.

Aufgrund innovativer Neuerungen in HTML5 wurde dessen Versionsnummer zu einem relativ verbreiteten Schlagwort, aber man sollte das als Anfänger nicht überbewerten: Eine Website »mit HTML5!« ist nicht automatisch gut, und eine Website ohne HTML5-Elemente ist nicht hoffnungslos veraltet.

Im Kern ist nämlich jedes Dokument gleich, egal ob es mit HTML1 oder HTML5 geschrieben wurde: Es besteht aus einem `html`-Element, das die Kindelemente `head` und `body` enthält – alles ist so, wie Sie es schon aus Ihrem ersten selbst geschriebenen Dokument kennen. Auch bei den grundlegenden Inhaltselementen wie `p` oder `h1` hat sich seit HTML3 wenig getan. Neue Versionen der Sprache enthalten in erster Linie Elemente, die das Einbinden von Multimedia-Inhalten wie Videos erleichtern. HTML5 hat zwar wirklich auch an anderen Stellen die Arbeit von Webdesignern verändert, aber die Grundlagen, die Sie lernen müssen, bleiben die gleichen.

Hinzu kommt, dass die jeweils aktuelle Browsergeneration nicht immer die neuesten HTML-Elemente darstellen kann – es dauert meistens ein wenig, bis die Hersteller diese unterstützen. Auf der anderen Seite stellen auch die Browser von morgen noch eine Website korrekt dar, die vor drei Jahren veröffentlicht wurde, solange sie sauber geschrieben ist. HTML ist also eine abwärtskompatible Sprache.

Für Ihre HTML-Projekte ist dieser kleine Geschichts- und Versionsexkurs nur deshalb wichtig, damit Sie folgende Dinge wissen:

- ▶ Es gibt jemanden, der die Rechtschreibung und Grammatik von HTML und CSS festlegt: das W3C.

- ▶ Weil die Browserhersteller sich an den Vorgaben des W3C orientieren, müssen Sie das ebenfalls tun, damit die Browser Ihre Seiten so darstellen, wie Sie sich das vorstellen.
- ▶ Es gibt unterschiedliche Versionen von HTML.

Mit diesem Hintergrundwissen gerüstet, können wir uns genauer anschauen, welche Informationen über Ihr HTML-Dokument Sie den Browsern mitteilen müssen, damit diese Ihre Seiten so darstellen, wie Sie es möchten. Diese Informationen nennt man *Metainformationen*.

#### 4.6.3 Der doctype

Die erste Metainformation ist der *doctype* (kurz für das englische *document type*, im Deutschen *Dokumenttyp*). Er teilt dem Browser mit, welche Art von Dokument der Browser sich gerade vom Server herunterlädt. Der *doctype* enthält nicht nur die HTML-Versionsnummer des W3C, nach deren Regeln der Browser sich beim Anzeigen der Seite richten soll, sondern auch das Dateiformat des Dokuments. Denn neben HTML gibt es noch weitere Textauszeichnungssprachen, mit denen ein Dokument auf einem Webserver formatiert sein könnte.

Der *doctype* unterscheidet sich von allem, was Sie bisher über den Aufbau eines HTML-Dokuments gelernt haben: Er wird nämlich in keines der in dem HTML-Dokument enthaltenen Elemente geschrieben, nicht einmal in das bisher alles andere umfassende `html`-Element. Stattdessen steht er in der allerersten Zeile des Dokuments:

```
<!DOCTYPE html>
```

Dieser *doctype* teilt dem Browser mit, dass es sich hier um ein HTML-Dokument handelt und dass der Browser sich beim Anzeigen nach den Bestimmungen von HTML5 richten soll.

Bei älteren HTML-Versionen ist der *doctype* etwas ausführlicher:

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

Dieser *doctype* deklariert ebenfalls ein HTML-Dokument, allerdings soll der Browser sich bei der Anzeige nach den Bestimmungen der HTML-Versionsnummer 4.01 richten. In der zweiten Zeile ist eine URL angegeben, unter der eine detaillierte Spezifikation dieses *doctype* zu finden ist.

Alle Beispielseiten, die wir in diesem Buch erstellen, richten sich nach der HTML-Version 5 und verwenden den entsprechenden *doctype*. Das empfehlen wir Ihnen auch für

Ihre eigenen Webseiten, denn den HTML5-doctype verstehen alle Browser, auch ältere. Nur wenn Sie unbedingt später einmal HTML-Elemente verwenden müssen, die der HTML5-Standard nicht mehr erlaubt, brauchen Sie einen anderen doctype. Welche Elemente mit HTML5 weggefallen sind, erfahren Sie unter [bnfr.de/ql407](http://bnfr.de/ql407).

#### 4.6.4 Das charset

Für andere Metainformationen werden meta-Elemente verwendet, die Kinder des head-Elements sein müssen und am besten vor allen anderen Kindern geschrieben werden. Unterschiedliche Attribute des meta-Elements stehen für unterschiedliche Arten von Informationen, und das wichtigste ist das Attribut charset:

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

Dieses Attribut benutzen Sie, um dem Browser mitzuteilen, welche Schriftzeichen er für die Darstellung dieses Dokuments beherrschen muss. Da es weltweit ja sehr viel mehr Schriftzeichen gibt als die 26 lateinischen Buchstaben, ist diese Angabe sehr wichtig. Allein schon die deutschen Umlaute ä, ö und ü machen ohne Angabe der Zeichencodierung meist Probleme. Mit der Codierung UTF-8 decken Sie alle gängigen Sonderzeichen von Sprachen mit lateinischer Schrift ab, inklusive der Sonderzeichen skandinavischer, osteuropäischer und romanischer Sprachen.

##### Wichtig: Einstellungen im HTML-Editor

Zum Thema Zeichenkodierung gibt es nicht nur innerhalb des HTML-Dokuments, sondern auch bei den Einstellungen des HTML-Editors einiges zu beachten:

- ▶ Achten Sie hinsichtlich der Codierung darauf, dass Sie nicht einfach nur das charset in einem meta-Element spezifizieren. Das HTML-Dokument muss natürlich im Editor auch tatsächlich mit der entsprechenden Codierung gespeichert werden. Überprüfen Sie dazu die Einstellungen in Ihrem Text- oder HTML-Editor.
- ▶ Sollte der Browser ein vorhandenes Dokument auch dann noch nicht korrekt darstellen, wenn Sie im Editor das entsprechende Format gewählt und das Dokument erneut gespeichert haben, kopieren Sie den Inhalt des alten Dokuments in ein ganz neues Dokument und speichern dieses ab. Bei manchen Editoren wird dadurch das Problem gelöst.

- ▶ Und wenn am Anfang einer HTML-Seite die Zeichenfolge »ï»¿« erscheint, dann sollten Sie die Seite mit der Zeichenkodierung »UTF-8 ohne BOM« speichern.
- ▶ Zur richtigen Formatierung gehört auch die richtige Einstellung für Zeilenumbrüche. Speichern Sie Dateien immer mit der Einstellung »Windows« oder »CR LF«, damit die Umbrüche auf allen Betriebssystemen angezeigt werden.

Wenn Sie von nun an ein neues HTML-Dokument erstellen, benutzen Sie also am besten immer folgendes Grundgerüst, bevor Sie weitere HTML-Elemente einfügen:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>
  </body>
</html>
```

Auf der Website zum Buch können Sie unter [bnfr.de/ql401](http://bnfr.de/ql401) eine außer diesem Grundgerüst leere HTML-Datei herunterladen. Dadurch sparen Sie sich das Abtippen.

Durch die Angabe von Metadaten erleichtern Sie dem Browser das korrekte Lesen Ihres HTML-Dokuments und die grafische Darstellung. Mit den folgenden Tipps machen Sie es sich selbst leichter, Ihre HTML-Dokumente auch dann noch zu verstehen, wenn Sie schon lange nicht mehr daran gearbeitet haben.

#### 4.6.5 Schaffen Sie Platz für mehr Übersicht

Wie Sie mittlerweile wissen, ist die Struktur eines HTML-Dokuments sehr verschachtelt. Das body-Element zum Beispiel enthält mehrere Kindelemente, die ihrerseits wiederum beliebig viele Kindelemente enthalten können. Um dabei nicht den Überblick zu verlieren, sollten Sie Ihr HTML im Editor übersichtlich anordnen, indem Sie die Zeilen mit Umbrüchen und Tabulatoren sinnvoll einrücken und Abstände zwischen den Zeilen lassen (siehe Abbildung 4.29).

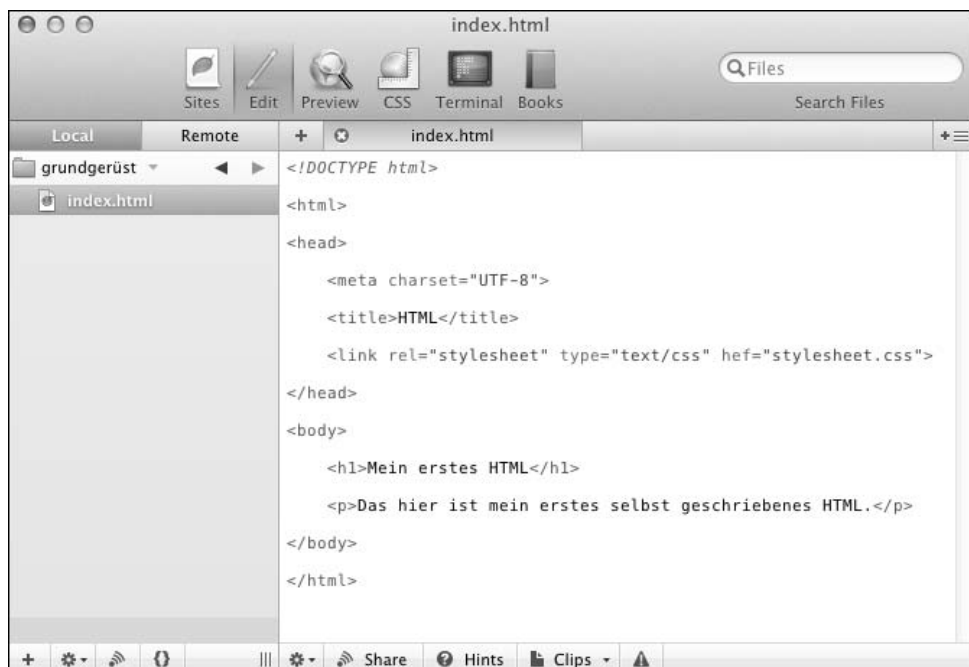


Abbildung 4.29 Den Browser stören die Umbrüche und Tabulatoren nicht, er ignoriert sie bei der Darstellung einfach.

So sehen Sie auf einen Blick, über wie viele Zeilen sich ein Element erstreckt und welche anderen Elemente es beinhaltet.

#### Wo ist meine Leerzeile hin?

Manche HTML-Editoren löschen Leerzeilen automatisch aus Ihrem HTML-Dokument. Bei manchen können Sie dieses Verhalten deaktivieren. Die Option dafür heißt meist ORIGINAL-FORMATIERUNG BEHALTEN.

#### 4.6.6 Notizen für Sie selbst: Kommentare

Oft werden HTML- und CSS-Dokumente erst nach langer Pause wieder geöffnet – zum Beispiel weil Sie einen kleinen Fehler im Text korrigieren wollen, den Sie bislang übersehen haben, oder weil Sie ein Projekt an jemand anderes abgegeben haben, der mit Ihren Dokumenten weiterarbeiten soll. Dank Leerzeilen und Einrückung erkennt man zwar

schnell, wie ein Dokument aufgebaut ist und welche Elemente es enthält – man versteht aber manchmal nicht mehr, warum man ein bestimmtes Element oder eine bestimmte CSS-Eigenschaft benutzt hat.

Um den Sinn des Geschriebenen festzuhalten, können Sie zwischen Ihrem HTML oder CSS platzierte *Kommentare* verwenden. Die Kommentare werden vom Browser bei der Darstellung ignoriert, wenn sie richtig formatiert sind. So schreiben Sie einen Kommentar in HTML-Dateien:

```
<!--Hier steht Ihr Kommentar-->
```

Versehen Sie Ihr HTML also so, wie in Abbildung 4.30 zu sehen ist, mit Kommentaren, um die Verständlichkeit weiter zu erhöhen.



Abbildung 4.30 Kommentare in einem HTML-Dokument

Richtig sinnvoll werden die Kommentare natürlich erst bei komplexeren Dokumenten, etwa an Stellen, für die Sie erst etwas Neues lernen mussten. Gedanken wie »Das charset muss immer das erste Kindelement im head-Element sein« können Sie ruhig festhalten.

So vergessen Sie das Gelernte nicht und verhindern, dass Sie eine eigentlich sinnvolle Seitenstruktur bei einer späteren Überarbeitung verändern, nur weil Sie in dem Moment nicht daran denken, warum Sie damals gerade diese Seitenstruktur gewählt hatten.

In CSS-Dateien haben Kommentare eine andere Syntax:

```
/*Das ist ein Kommentar innerhalb von CSS*/
```

#### Kommentare sind nicht völlig unsichtbar

Bedenken Sie aber, dass Kommentare im Quelltext Ihrer Seite stehen und von anderen Nutzern gesehen werden können, wenn diese sich den Quelltext anzeigen lassen. Schreiben Sie also nichts in die Kommentare, was Ihnen peinlich wäre, und schon gar nicht sensible Daten wie Passwörter!

#### 4.6.7 Validieren Sie Ihre Seiten

Auch ein HTML-Dokument, das für Sie übersichtlich strukturiert ist und das der Browser so anzeigt, wie Sie es geplant haben, ist nicht unbedingt vollständig korrekt. Nach den Vorgaben des World Wide Web Consortium ist nicht alles, was offenbar funktioniert, auch erlaubt. Die Browser tolerieren viele Fehler – aber darauf können Sie sich nicht verlassen.

Wenn Sie etwa notwendige Attribute vergessen oder Elemente in eine Reihenfolge bringen, die das W3C so nicht vorgesehen hat, korrigiert Ihr Browser im besten Fall die Fehler und versucht, das Dokument trotzdem korrekt darzustellen. Im schlechtesten Fall stellt der Browser die Seite aber doch nicht richtig dar. Auch wenn Sie Ihr Dokument mit der Debugging-Funktion Ihres HTML-Editors kontrolliert haben, ist das leider nicht immer ausreichend, weil der Editor sich vielleicht noch nach älteren HTML-Versionen richtet.

Das W3C bietet unter [bnfr.de/ql408](http://bnfr.de/ql408) einen komfortablen und kostenlosen *Validierungsdienst* an. Geben Sie einfach die URL Ihres HTML-Dokuments an oder laden Sie es hoch, und Sie bekommen recht detailliert (auf Englisch) Rückmeldung, ob und wo Sie in Ihrem HTML Standards verletzen.

Um noch einmal die Parallele zum deutschen Duden aufzugreifen: Sicher verstehen viele Mitmenschen Sie auch, wenn Sie auf korrekte Rechtschreibung und Grammatik verzichten. Aber die größtmögliche Verständlichkeit erreichen Sie, wenn Sie sich an die Regeln halten.

## 4.7 Fazit

In diesem Kapitel haben Sie eine ganze Menge gelernt:

- ▶ dass Sie HTML zur Strukturierung von Webseiten benutzen
- ▶ dass HTML aus Elementen besteht, die wiederum aus Start- und End-Tags sowie aus Inhalt bestehen
- ▶ dass Sie CSS zur Gestaltung von HTML-Seiten benutzen, indem Sie den HTML-Elementen mithilfe von Selektoren, Eigenschaften und Werten Formate zuweisen

Sie wissen jetzt, was HTML und CSS ist, wie man es grundsätzlich benutzt und mit welchen Werkzeugen Sie arbeiten können. Nun sind Sie in der Lage, Ihre erste komplett selbst geschriebene Website zu erstellen – was wir im nächsten Kapitel mit einem neuen Beispielprojekt tun werden.

#### Beispieldateien zum Download

Die in diesem Kapitel erstellten HTML- und CSS-Dateien finden Sie auf der Website zum Buch unter [bnfr.de/ql401](http://bnfr.de/ql401).