

---

Sich mit HTML und CSS befassen

---

Die Rolle von JavaScript und AJAX untersuchen

---

Herausfinden, was JavaScript und AJAX für Webseiten bringen

---

Einen Editor wählen

---

Werkzeuge für den Browser zusammenstellen

---

## Kapitel 1

# Das Web auf die nächste Stufe befördern

**D**as Web hat schon seinen Reiz. Webseiten zusammenzustellen, ist überaus interessant, und kaum ein Unternehmen kann heute auf sie verzichten. Und mit dem Wachstum und den Änderungen des Webs haben sich auch die Erwartungen an Inhalt und Funktionsweise von Webseiten gewandelt. Wenn Sie bereits HTML oder XHTML (Extensible Hypertext Markup Language) kennen, ist das Erstellen von Webdokumenten für Sie nichts Neues.

Und so, wie sich das Web weiterentwickelt, entwickeln sich auch die Werkzeuge weiter, mit denen Webseiten und Webdokumente erstellt werden. JavaScript und AJAX sind dabei leistungsstarke Werkzeuge, mit denen Sie dynamische Webdokumente anlegen können. Dieses Kapitel gibt Ihnen einen ersten Überblick über einige der grundlegenden Technologien für die Erstellung von Webseiten.

## Etwas wirklich Cooles zusammenbauen

In diesem Buch geht es darum, Webseiten um Funktionen zu erweitern, die Sie mit einfachem HTML und CSS *nicht* zustande bringen.

Lassen Sie sich nicht täuschen, wir sprechen hier von Programmierung – und Programmierung ist nun einmal härter als die gute alte Webentwicklung. Aber es lohnt sich, wie das Beispiel aus Abbildung 1.1 zeigt.



**Abbildung 1.1:** Diese im Internet-Browser angezeigte Seite besitzt einige interessante Funktionen.



Damit dieses Beispiel nicht zu kompliziert wird, habe ich darin externe Bibliotheken verwendet. Diese werden in Teil IV dieses Buches erklärt, weshalb Sie im Moment einfach akzeptieren sollten, dass hier etwas Aufregendes geschieht.

Dieses Programm setzt eine aktive Internetverbindung voraus, um wie gewünscht arbeiten zu können. In Kapitel 10 finden Sie Alternativen, um sich mit externen Bibliotheken zu verbinden.



Wenn Sie dieses Beispiel in Aktion sehen wollen (was sich lohnt), gehen Sie über <http://www.wiley-vch.de/publish/dt/books/ISBN3-527-71444-8> zur Website dieses Buches. Dort finden Sie dieses und jedes andere Programm und Beispiel aus diesem Buch zum Herunterladen. Dann müssen Sie die Datei `demo.html` nur noch in Ihren Internet-Browser laden. Im Lesezeichen-Eingabefeld steht dann eine Adresse mit diesen Angaben. `file:///.../demo.html`. Die drei Punkte werden durch die Position der in den Browser geladenen Datei ersetzt.

Auf den ersten Blick sieht die Webseite recht einfach aus, aber wenn Sie sie in einem Browser öffnen (was Sie machen sollten) und mit ihr herumspielen, werden Sie schnell entdecken, dass sie viele Überraschungen enthält. Diese sehr einfache Seite zeigt einige der Gründe auf, warum Sie sich mit JavaScript und AJAX befassen sollten.

- ✓ **Die Schaltflächen bewirken etwas.** Vielleicht können Sie ja mit reinem HTML Formularbestandteile (wie Schaltflächen und Textfelder) erstellen, aber HTML kann mit diesen Schaltflächen und Textfeldern nichts anfangen. Dafür benötigen Sie eine Programmiersprache.

Damit etwas Interessantes geschieht, verwenden die Schaltflächen JavaScript, um etwas auszuführen. Außerdem ist die Tatsache, dass die Seite interaktiv geworden ist, eine große Änderung. Mit JavaScript können Sie *Anwendungen* und nicht nur Seiten anlegen.

- ✓ **Die Schaltfläche ZÄHLEN ändert die Seite dynamisch.** Wenn Sie auf die Schaltfläche ZÄHLEN klicken, wird der Seite automatisch ein neuer Inhalt hinzugefügt. Ein Programm zählt, wie oft die Schaltfläche ZÄHLEN angeklickt worden ist, und fügt dem Bereich mit der Beschriftung *Beobachten Sie diesen Bereich* eine Zeile Text hinzu. Wenn der Benutzer mit der Seite in eine Wechselwirkung (auch *Interaktion* genannt) tritt, verwendet die Seite Material, das sich ursprünglich nicht auf dem Server befand.

Klar, dieses Beispiel ist einfach gehalten, aber Sie können jedem Webelement dynamisch jeden Text hinzufügen. Damit steht Ihnen eine sehr leistungsstarke Möglichkeit zur Ver-

fügung. Abbildung 1.2 zeigt, wie die Seite aussieht, nachdem ich mehrfach auf die Schaltfläche ZÄHLEN geklickt habe.



**Abbildung 1.2:** Die Schaltfläche ZÄHLEN ändert in einem Bereich der Seite den Text.

- ✓ **Die Schaltfläche SICHTBARKEIT EIN/AUS lässt Dinge erscheinen und verschwinden.** Sie können mit HTML nicht wirklich dafür sorgen, dass Elemente auf dem Bildschirm erscheinen oder verschwinden. Das lässt sich zwar bis zu einem gewissen Grad mit CSS erreichen, aber JavaScript gibt Ihnen einen sehr leistungsstarken Werkzeugsatz an die Hand, mit dem Sie festlegen können, welche Teile einer Seite für den Benutzer sichtbar sind. Abbildung 1.3 zeigt die Seite, auf der nun das Element für die Textausgabe ausgeblendet ist.

Ich habe auf die Schaltfläche SICHTBARKEIT EIN/AUS geklickt, und die Textausgabe ist verschwunden.



**Abbildung 1.3:** Klicken Sie auf die Schaltfläche SICHTBARKEIT EIN/AUS, damit das Element mit der Textausgabe wieder erscheint.

- ✓ **Die Schaltfläche STIL ÄNDERN ändert sofort das Erscheinungsbild eines Teils der Seite.** Sie können JavaScript verwenden, um die Inhalte eines beliebigen HTML-Elements auf der Seite zu ändern – und Sie können JavaScript *auch* verwenden, um das Erscheinungsbild (das CSS) in Echtzeit zu modifizieren. Ich habe für unser Beispiel eine CSS-Klasse erstellt, die `funky` heißt und dem Ausgabefeld hinzugefügt oder von dort entfernt wird, wenn ein Benutzer auf die Schaltfläche klickt. Diese Vorgehensweise funktioniert mit jeder CSS-Klasse. (Super, oder?) Abbildung 1.4 zeigt die neu geladene Seite, nachdem ihr die Klasse `funky` zugewiesen worden ist.



**Abbildung 1.4:** Sie können das Erscheinungsbild eines jeden Seitenelements dynamisch ändern.



Ich habe in diesem Kapitel einige der Abbildungen mit Beschriftungen versehen, die die Geschehnisse beschreiben. Die hier gezeigten Abbildungen reichen aber nicht immer aus, um die Vorgänge verstehen zu können. Sie finden das HTML-Programm unter <http://www.wiley-vch.de/publish/dt/books/ISBN3-527-71444-8>. Laden Sie es herunter, und schauen Sie es sich an.

- ✓ **Die Schaltfläche ANIMIEREN** führt schrittweise eine Reihe von Manipulationen am Ausgabefeld durch, indem im Verlauf einer vordefinierten Zeitspanne seine Größe, seine Form und sein Aussehen geändert werden. (Das müssen Sie selbst ausprobieren, da sich diese Funktion nicht mit einer einfachen Abbildung darstellen lässt.)
- ✓ **Die Schaltfläche CODE ANZEIGEN** zeigt den Text einer externen Datei an. Diese Schaltfläche verwendet eine einfache Form von AJAX, um eine externe Datei in Echtzeit in die Seite zu laden. Mit dieser einfachen Möglichkeit lassen sich Seiten modular aufbauen. In unserem Fall nehme ich mir einfach eine Kopie des JavaScript-Codes, damit Sie sehen können, wie diese Funktion arbeitet. Machen Sie sich nichts daraus, wenn Sie den Code noch nicht verstehen. Aus diesem Grund, gibt es dieses Buch schließlich. Abbildung 1.5 zeigt das Ergebnis des Datei-Imports.
- ✓ **Lassen Sie den Benutzer Inhalte verschieben.** Ein Benutzer kann Seitenelemente mit der Maus aufnehmen und innerhalb der Seite verschieben. Dies ist eine weitere Funktion, die es so auf normalen Webseiten nicht gibt.



Natürlich veröffentlichen Sie normalerweise Ihren Code nicht. Ich möchte damit nur zeigen, wie einfach es prinzipiell ist, vom Server einen beliebigen Text zu laden. Ihnen kann ich hier ruhig auch den Inhalt einer Textdatei als Vorschau auf den Code zeigen.

Klicken Sie hier, um den Code aus einer Datei angezeigt zu bekommen.

**Animierte Demo einer Seite**

Zählen   Sichtbarkeit ein/aus   Stil ändern   Animieren   Code anzeigen

```
//demo.js
//zusammen mit demo.html verwenden
//Andy Harris, für JavaScript und AJAX für Dummies

//globale Variable used for counting
var i = 0;

function doCount(){
  //i um 1 erhöhen
  i++;

  //den aktuellen Text aus dem Ausgabefeld holen
  var currentText = $("#output").html();

  //den Text um den neuen Zählerstand erweitern
  currentText += "Zählerstand: " + i + "
";

  //re-write current text to div
  $("#output").html(currentText);
} // end doCount

function toggleVis(){
  //make the output div appear or disappear
  $("#output").toggle();
} // end toggleVis
```

Beobachten Sie diesen Bereich

**Abbildung 1.5:** Durch die Schaltfläche CODE ANZEIGEN wird Code aus einer Datei geladen und innerhalb der Seite dargestellt.

Diese Möglichkeiten sollten ein wenig prahlerisch überzeugen, sind aber nur der Anfang. Wenn Sie zu programmieren lernen, können Sie Ihre Webseiten nach und nach von statischen Dokumenten in lebendige Anwendungen verwandeln, die auf neuen Wegen mit ihren Benutzern interagieren können.



Nehmen Sie sich die Zeit, um sich näher mit dem Code dieses Projektes zu beschäftigen. Er befindet sich in drei Dateien: `demo.html`, `demo.css` und `demo.js`. Diese befinden sich in gepackter Form auf der Webseite des Verlags zu diesem Buch. Benutzen Sie im Browser eine Funktion zum Ansehen des Quellcodes, wie beispielsweise `TOOLS|QUELLTEXT ANZEIGEN` (Chromium-Webbrowser) oder auch `WEB-ENTWICKLER|SEITENQUELLTEXT ANZEIGEN` (Firefox), um den Inhalt der HTML-Datei im Fenster des Internet-Browsers darstellen zu lassen.

Der Code enthält sehr viele Kommentare, die ich eingefügt habe, damit Sie die Vorgänge besser nachvollziehen können. Es ist aber auch völlig in Ordnung, wenn Sie sich momentan noch nicht mit dem dort vorhandenen, doch recht komplexen Quelltext auseinandersetzen. Sehen Sie diesen Code erst einmal als einen Ausblick auf das, was in diesem Buch noch auf Sie zukommt.

## Erste Schritte

Anscheinend macht es Spaß, Webseiten Tricks vorführen zu lassen, wie sie im vorigen Abschnitt beschrieben werden. Es gibt zwar noch sehr viel zu lernen, aber nur keine Panik: Ich werde Ihnen dabei helfen, alle Klippen zu umschiffen. Im ersten Schritt werden Sie sich erst

einmal ein wenig um die zentralen Technologien kümmern, auf denen JavaScript und AJAX basieren, damit Sie sehen, in welcher Beziehung diese beiden zu anderen (fortschrittlicheren) Technologien stehen, die Sie möglicherweise zukünftig benötigen werden.

## Ein Überblick über zentrale Technologien

JavaScript und AJAX sind zwar leistungsstark, aber funktionieren nicht für sich allein. Sie machen nur im Umfeld von Webseiten Sinn. Damit sind sie auf verschiedene weitere Webtechnologien angewiesen. Wenn Sie eine Anwendung mit JavaScript zusammenbauen wollen, benötigen Sie:

- ✓ **HTML:** HTML (Hypertext Markup Language) ist die grundlegende Auszeichnungssprache (*Markup Language*) zur Beschreibung von Webseiten. Es handelt sich dabei um eine ziemlich einfache Methode, mit der Webseiten erstellt werden können, bei der man nicht mehr als einen Texteditor benötigt.
- ✓ **XHTML:** XHTML ist der Nachfolger von HTML 4. Da es bestimmte Tags nicht zulässt, ist XHTML eigentlich weniger leistungsfähig und leider in einigen Belangen umständlicher.
- ✓ **HTML5:** Die fünfte Fassung der HTML-Spezifikation ersetzt die bisherigen HTML-Standards. HTML5 steht mit CSS eine Sprache zur Seite, über die das Aussehen der dargestellten Elemente gestaltet werden kann. Mit HTML5 wird das Grundgerüst eines HTML-Dokuments wiederhergestellt, wodurch die sperrigen Formalien der HTML-Vorgängerversionen auch »offiziell« entfallen können:

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

<body>
Inhalt des Dokuments...
</body>

</html>
```

Wenn Sie dabei die Standardvorgaben wie UTF-8 für die Zeichencodierung verwenden wollen, können Sie auch die entsprechende Zeile noch weglassen. Die meisten Browser sind zudem äußerst nachsichtig, wenn das eine oder andere formale Element in HTML-Dokumenten fehlt.

Als wirkliche Neuerungen und Erweiterungen kamen mit HTML5 Elemente für Grafik und Multimedia hinzu (<svg>, <canvas>, <audio> und <video>).

- ✓ **CSS:** CSS (Cascading Style Sheets – gestufte Gestaltungsbögen) bieten eine Möglichkeit, HTML- oder XHTML-Seiten Gestaltungsvorlagen hinzuzufügen. HTML stellt die allgemeinen Rahmenbedingungen bereit, während CSS beispielsweise Farbe und Formatierung beschreibt.

- ✓ **JavaScript:** JavaScript ist eine Programmiersprache, die in allen modernen Webbrowsern eingebettet ist. Sie wurde für die Zusammenarbeit mit Webseiten entwickelt. Sie können sie benutzen, um Informationen aus Teilen einer Seite herauszuziehen und die Seite in Echtzeit zu ändern.
- ✓ **AJAX:** AJAX (Asynchronous JavaScript And XML) ist eine Technik, mit der JavaScript direkter mit dem Webserver kommunizieren kann. Es erzeugt zwischen dem Browser und dem Webserver eine interessante neue Beziehung. Ungefähr die Hälfte dieses Buches hat mit AJAX zu tun.
- ✓ **PHP:** PHP (rekursives Akronym für *PHP: Hypertext Preprocessor*, wobei hier *PHP Personal Home Page Tools* bedeutet) ist eine von mehreren wichtigen Sprachen für die Arbeit auf einem Webserver. PHP spielt in diesem Buch zwar keine Hauptrolle, beherrscht aber Dinge, die JavaScript nicht kann. AJAX wird häufig verwendet, um JavaScript-Anwendungen mit PHP-Programmen zu verbinden. Sie erhalten in Kapitel 14 eine kurze Einführung in PHP.
- ✓ **Java:** Java ist eine Sprache, die sich (trotz des ähnlichen Namens) völlig von JavaScript unterscheidet. Obwohl Java in einer Client-Server-Beziehung sowohl auf der Seite des Clients als auch auf der des Servers sinnvoll eingesetzt werden kann, liegt der Schwerpunkt dieses Buches nicht auf dieser Sprache.



Wenn Sie eine Online-Hilfe für JavaScript suchen, achten Sie darauf, dass Sie mit *JavaScript*-Experten und nicht mit *Java*-Programmierern kommunizieren. Auch wenn diese Sprachen ähnliche Namen haben, so unterscheiden sie sich doch vollständig voneinander. Java-Programmierer halten sich manchmal gerne für etwas Besseres, und Sie erhalten als Antwort nur wenig Sinnvolles, wenn Sie in einem Java-Forum eine Frage zu JavaScript stellen. Sprechen Sie mich im Zweifelsfall auf meiner Website ([www.aharrisbooks.net](http://www.aharrisbooks.net)) – bitte auf Englisch – an. Ich kann Ihnen bei beiden Sprachen helfen, und ich werde mich nicht über Sie lustig machen, wenn Sie dabei ein wenig durcheinanderkommen.



Wenn es um Referenzen oder Wikis rund um die Webprogrammierung geht, lohnt es sich meist, bei SELFHTML vorbeizuschauen. Dort finden Sie auch einen Schnellindex für JavaScript (<https://wiki.selfhtml.org/wiki/Schnell-Index/JavaScript>). Stöbern Sie dort am besten selbst ein wenig herum.)

## Die Auswahl eines Computers

Natürlich benötigen Sie einen Computer. Glücklicherweise gibt es dabei fast keine Vorgaben. Beim Erstellen von Webseiten können Sie eigentlich jeden Computer benutzen, mit dem Sie die Seiten auch betrachten können. Bei den Betriebssystemen eignet sich jedes der größeren (Windows, Mac oder Linux). Ich selbst benutze meist eine Kombination aus Linux (Fedora Core) und einer aktuelleren Windows-Version, die noch mit Aktualisierungen versorgt wird. Die in diesem Buch vorgestellten Programme sollten – bis auf eher unerhebliche Kleinigkeiten – auf jedem vernünftigen Computer funktionieren.

Es gibt sicherlich Gründe dafür, Ihre Webseiten im Internet bereitzustellen. Heute gibt es die Alternative, auf Ihrem Computer einen lokalen Webserver einzurichten; Sie können ein NAS-Laufwerk nehmen und es ins Internet einbinden oder einen Online-Hostingdienst in An-

spruch nehmen. Mittlerweile lässt sich angesichts der Vielzahl von Lösungsmöglichkeiten zwischen lokalem Netzwerk, dem WWW, aber auch den verschiedensten Cloud-Varianten kaum noch der Überblick behalten. Selbst Online-Hosting bekommen Sie häufig recht preisgünstig oder sogar kostenlos. Wenn Sie wollen, dass zu Ihrer Website auch ein bestimmter Name gehört (zum Beispiel `www.meineSite.de`), müssen Sie für das Registrieren einer Domäne eine jährliche Gebühr bezahlen. Online-Dienstleister setzen häufig Linux ein, aber in der Regel verwenden sie eine Oberfläche, die diese Einzelheiten vor Ihnen verbirgt.



Denken Sie bei öffentlich zugänglichen Internet-Angeboten an die juristischen Fallstricke!

Die richtigen Werkzeuge erleichtern jede Arbeit, und viele der tollen Software-Werkzeuge sind wirklich kostenlos erhältlich. Da diese Werkzeuge Open Source sind (und damit eine Lizenz besitzen, die das Weitergeben erlaubt), ist es legal, sie zu benutzen, ohne dafür etwas zahlen zu müssen – solange Sie diese Werkzeuge nicht kommerziell einsetzen.

Sie können mit einem Texteditor und einem Browser auf jedem Computer grundlegende Webentwicklungen betreiben. Sobald sich Ihre Kenntnisse der Webentwicklung weiterentwickeln, glauben Sie vielleicht, leistungsfähigere Werkzeuge kaufen zu müssen. Wenn Sie weiterlesen, werden Sie aber herausfinden, dass es auch hervorragende Werkzeuge gibt, die nichts kosten. Oft stören eher die unterschiedliche Behandlung der Zugriffsrechte und/oder der Umlaute oder das andersartige Dateisystem, wenn beispielsweise Linux-Systeme eingesetzt werden.

## Einen Editor auswählen

Webseiten, JavaScript, HTML und CSS bestehen aus reinem Text. Um so etwas zu schreiben, benötigen Sie kein bestimmtes Programm. Natürlich erleichtert es das Leben, wenn man das richtige Werkzeug besitzt. Und da Sie ziemlich viel Zeit zusammen mit Ihren Webwerkzeugen verbringen werden, sollten Sie deren Möglichkeiten kennen.

## Vermeiden Sie problematische Werkzeuge

Mit der Verwendung der falschen Werkzeuge können Sie sich Ihr Leben ganz schön schwer machen. Daher erwähne ich hier einige Werkzeuge, mit denen Sie Ihren Job als Webentwickler kaum vernünftig erledigen können:

- ✓ **Microsoft Word:** Textverarbeitungssysteme haben wirklich ihre Stärken (ich selbst habe eines beim Verfassen dieses Buches benutzt), aber sie wurden eigentlich nicht für das Erstellen von Webseiten entwickelt. Word (und alle anderen Textverarbeitungen) legen in ihren Dateien zusätzlich zum reinen Text viele Informationen ab. Dieses ganze Formatierungszeugs wird für Dokumente benötigt, die nicht für das Web bestimmt sind. Aber HTML und CSS gehen bei der Verwaltung dieser Daten eigene Wege, wodurch dieser Kram jetzt nur noch im Weg ist. Selbst die Option, etwas als HTML zu speichern, führt oft zu Problemen. Selbst wenn Word die Seite in HTML-Form ablegt, lässt sich mit den sperrigen Formatierungen von Word nur sehr schwer arbeiten. Zudem können die so erzeugten Ergebnisse nicht für JavaScript verwendet werden.



- ✓ **Editor/Notepad:** Hierbei handelt es sich um das klassische Werkzeug, das in den meisten Windows-Versionen existiert. Je nach Version heißt es Editor oder Notepad. Die Anwendung speichert Seiten in reinem Textformat ab, weshalb es sich für die Webentwicklung besser als Word eignet. Allerdings ist Editor/Notepad für die ernsthafte Arbeit viel zu einfach gestrickt. Dem Programm fehlen bereits grundlegende Funktionen wie eine Zeilennummerierung oder die Möglichkeit zum gleichzeitigen Arbeiten mit mehreren Dokumenten.
- ✓ **TextEdit:** Der standardmäßige Texteditor auf dem Mac ist ein sehr leistungsstarkes Werkzeug, aber die Anwendung ähnelt mehr einer Textverarbeitung als dem, was ich als echten Texteditor bezeichne. Wenn Sie in TextEdit eine HTML-Datei speichern, wird sie normalerweise nicht so abgelegt, wie Sie sie benötigen: Sie sehen dann nicht das Ergebnis des Codes, sondern den Code selbst. Wenn Sie TextEdit für HTML oder JavaScript verwenden wollen, müssen Sie vor dem Speichern im Formatierungsmenü festlegen, dass reiner Text erzeugt werden soll.

## Einen WYSIWYG-Editor verwenden

Die Versprechungen von WYSIWYG-Editoren (WYSIWYG: *What You See Is What You Get* oder *Was Sie sehen, erhalten Sie auch*) sind sehr verführerisch. Textverarbeitungen besitzen diese Fähigkeit seit Jahren. Während der Bearbeitung eines Dokuments auf dem Bildschirm können Sie dessen Aussehen in Echtzeit in einer Vorschau für das Papier beobachten. Viele Werkzeuge versprechen diese Funktion auch für die Entwicklung von Webseiten: Das bekannteste ist vielleicht Adobe Dreamweaver, gefolgt von Microsoft Expression Web. Auch wenn diese Werkzeuge im Bereich der herkömmlichen Webentwicklung sehr beliebt sind, so haben sie doch einige Nachteile, wenn es um die Art interaktiver Arbeit geht, mit der ich mich in diesem Buch befasse:

- ✓ **WYSIWYG ist eine Lüge.** WYSIWYG funktioniert dann sehr gut, wenn die Ausgabe in Form eines *Papierdokuments über einen Drucker* erfolgt. Sie können das Aussehen der Ausgabe vorhersagen. Webseiten verhalten sich anders, weil die Ausgabe auf einem Bildschirm erfolgt, der irgendjemandem gehört. Sie wissen nicht, welche Größe der Bildschirm hat, und vielleicht nicht einmal, welche Farben er unterstützt und welche Schriftarten installiert sind. Außerdem wissen Sie nicht, mit welchem Browser Benutzer die Seiten betrachten, was für erhebliche Unterschiede zur Ausgabe auf Papier sorgt.
- ✓ **Der Editor verbirgt benötigte Einzelheiten.** Ein grafischer Editor versucht, einige Einzelheiten der Webentwicklung vor Ihnen zu verbergen. Das ist zunächst einmal in Ordnung, aber irgendwann kommen Sie an den Punkt, an dem genau diese Kontrolle nötig ist. Die meisten mit Dreamweaver arbeitenden Profis verbringen die meiste Zeit mit der Ansicht CODE und ignorieren die Vorteile eines grafischen Editors. Warum wollen Sie für Funktionen bezahlen, die Sie dann doch ignorieren?
- ✓ **Grafische Editoren gehen von statischen Dokumenten aus.** Ein grafischer Editor basiert auf der Idee, dass es sich bei einer Webseite um ein ganz normales Dokument handelt. Aber die Seiten, die in diesem Buch erstellt werden, sind viel mehr als das. Sie schreiben (zum Beispiel) Code, der Webdokumente auf die Schnelle erstellt und modifiziert. Sie müssen wissen, wie Webdokumente manuell erstellt werden, damit Sie den Code schreiben können, der sie anlegt und dynamisch ändert.

## Die Editoren der Programmierer

Es gibt nun eine Reihe von Editoren, die versuchen, die Lücke zwischen reinen Texteditoren und WYSIWYG-Werkzeugen zu füllen. Diese Editoren schreiben reinen Text, unterstützen aber zusätzliche Funktionen für Programmierer wie:

- ✓ **Spracherkennung:** Nein, nicht, was Sie jetzt vielleicht denken. Editoren für Programmierer erkennen oft, welche Programmiersprachen in Dokumenten verwendet werden und können sich so anpassen, dass sie Ihnen helfen, wenn Sie HTML-, JavaScript- oder CSS-Code schreiben. Die meisten der Editoren für Programmierer geben sich bei den erwähnten Sprachen keine Blößen und können mit ihnen umgehen.
- ✓ **Syntaxhervorhebung:** Verschiedene Sprachelemente werden in unterschiedlichen Farben wiedergegeben, damit Sie leichter erkennen können, ob es sich um reinen Text handelt, ob es etwa ein HTML-Tag ist und so weiter. Diese einfache Funktion erleichtert es, Probleme wie fehlende Auszeichnungen und die allgemeine Seitenstruktur zu erkennen.
- ✓ **Syntaxunterstützung:** Editoren für Programmierer sorgen häufig für eine Art Hilfe, damit Sie sich an die Syntax der Sprache erinnern. Diese Hilfen können Schaltflächen und Makros für allgemein üblichen Code sein. Es kann sich aber auch um vorgefertigte Vorlagen für standardmäßige Layouts, Muster und vielleicht sogar automatische Syntaxvervollständigungen handeln. Die letzte Funktion beobachtet, was Sie schreiben, und unterbreitet Ihnen auf der Basis der verwendeten Sprache Vorschläge.
- ✓ **Es werden mehrere Dokumente unterstützt:** Fortgeschrittenere Webanwendungen verlangen oft, dass gleichzeitig mehrere unterschiedliche Dokumente bearbeitet werden. So können Sie beispielsweise ein Dutzend Webseiten mit einigen CSS-Gestaltungsvorlagen und einer oder zwei externen JavaScript-Dateien geöffnet haben. Mit einem Editor für Programmierer können Sie diese Dateien parallel betrachten und bearbeiten. Bei vielen dieser Editoren können Sie auch übergeordnete *Projektdateien* anlegen, damit Sie alle Dateien, die zusammengehören, automatisch als Stapel speichern und laden können.
- ✓ **Makros:** Programmierung bedeutet oft, sich wiederholende Schritte zu schreiben. Das Vorhandensein einer Funktion, die Folgen von Tastatureingaben als *Makros* aufzeichnet und abspielt, kann unglaublich hilfreich sein.
- ✓ **Fehlersuche und Vorschau:** Die meisten Editoren für Programmierer enthalten Möglichkeiten, eine Vorschau der Ergebnisse im Browser (oder direkt im Editor) anzuzeigen. Die Editoren enthalten häufig auch Werkzeuge, die bestimmte Fehler vorhersagen oder auf auftretende Fehler reagieren können. Sie müssen zumindest direkt zu einer bestimmten Zeile oder einem Codeabschnitt springen können. Die Suche nach Fehlern wird auch *Debuggen* genannt.
- ✓ **Unterstützung von Einrückungen:** Die meisten Programmierer verwenden Einrückungen als leistungsstarkes Werkzeug und Hilfe, um die Struktur der Webdokumente leichter verstehen zu können, die sie erstellen. Ein guter Editor kann Ihnen bei diesen Einrückungen helfen, damit Sie sofort erkennen, wo Sie bei der Struktur Ihres Dokuments Fehler gemacht haben.

## Sich mit Editoren vertraut machen

Mir fallen sofort einige vielfältig einsetzbare Editoren für Programmierer ein (die teilweise leider nur in Englisch zur Verfügung stehen, wie die Abbildungen 1.6 bis 1.10 zeigen). Sie sollten darüber nachdenken, eine oder mehrere dieser kostenlosen Anwendungen zu installieren:

- ✓ **vi, vim und Emacs:** Bei diesen handelt es sich um die Großväter aller Texteditoren oder deren Abkömmlinge. Sie sind in Unix-/Linux-Umgebungen sehr beliebt, und es gibt auch Versionen für Windows und Mac. Obwohl die Editoren sehr leistungsfähig sind, wurden sie doch zu einer Zeit entwickelt, in der sich die Vorstellungen der Benutzer stark von den heutigen unterscheiden. Wenn Sie bisher mit einem dieser Werkzeuge gearbeitet haben, sollten Sie es sich gut überlegen, ob Sie sie gegen eine modernere Variante austauschen wollen. (Gut, ich gebe es zu: Mein wichtigster Texteditor ist immer noch Emacs, wobei ich heute bei all den einfacher zu bedienenden Möglichkeiten mit diesem Programm erst gar nicht mehr angefangen hätte.) Abbildung 1.6 zeigt eine JavaScript-Datei, die mit Emacs bearbeitet wird.

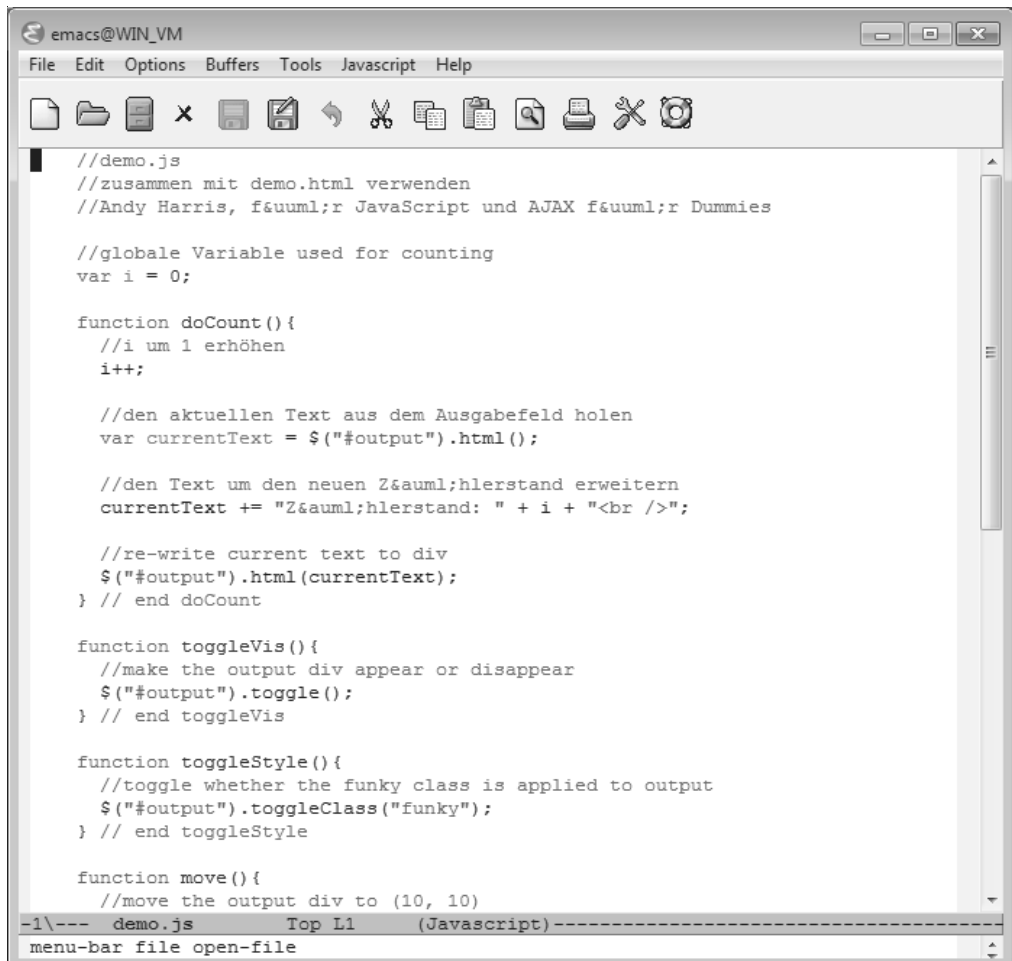


Abbildung 1.6: Emacs zeichnet sich nicht durch besondere Schönheit aus, ist aber sehr leistungsfähig.

- ✓ **Notepad++:** Hierbei handelt es sich um das Programm, das Editor/Notepad für Windows eigentlich sein sollte. Das Programm startet mit der Geschwindigkeit und Einfachheit von Notepad, enthält aber unzählige Funktionen für Programmierer. Ich liebe an diesem Programm ganz besonders die eingebaute Unterstützung der Gültigkeitsprüfung (Validierung) einer Seite. Notepad++ ist eine der wenigen Anwendungen, die eine dauerhafte Desktopverknüpfung verdient haben. Unglücklicherweise läuft es nur unter Windows. Abbildung 1.7 zeigt die Datei aus Abbildung 1.6, die hier in Notepad++ bearbeitet wird.



**Abbildung 1.7:** Sie werden feststellen, dass Notepad++ eine sehr leistungsfähige Alternative zum Editor von Windows ist.

- ✓ **Bluefish:** Der Texteditor Bluefish hat sich schnell zu einem beliebten Werkzeug von Webentwicklern entwickelt. Er ist nicht nur schnell, sondern auch leistungsstark und besitzt viele großartige Funktionen für Webentwickler. Eine sehr mächtige Funktion ist der »CSS-Baukasten«, der Ihnen dabei hilft, Stylesheets über ein Menüsystem zu erstellen, wodurch Sie sich die Syntax zusammenklicken können. Außerdem gibt es einen ausgezeichneten Generator für Standardvorlagen, der das Erstellen echter HTML-Webseiten stark vereinfacht. Bluefish gibt es für alle größeren Betriebssystemplattformen. Abbildung 1.8 zeigt Bluefish bei der Arbeit.

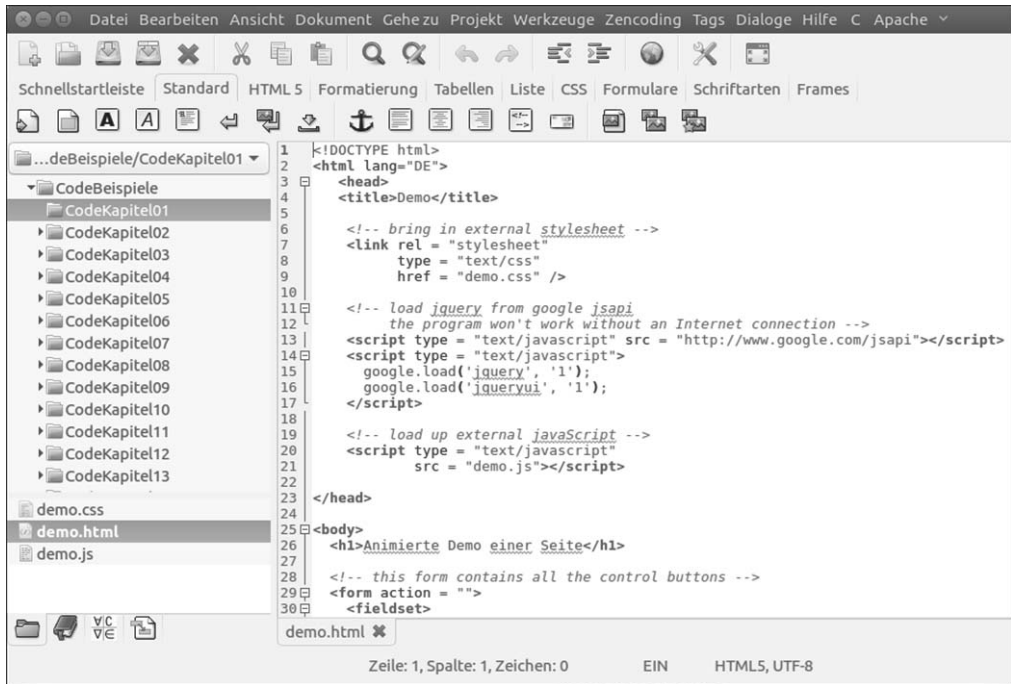


Abbildung 1.8: Bluefish ist ein guter Editor für HTML und JavaScript.

- ✓ **jEdit:** Dieser leistungsstarke Editor ist in Java geschrieben, weshalb er auch für so gut wie jede Betriebssystemplattform zur Verfügung steht. Er ist schon in seiner Grundform sehr stark, aber seine beste Funktion ist eine umfangreiche Plugin-Bibliothek, über die Sie den Editor an Ihre Bedürfnisse anpassen können. Wenn Sie die XML-Bibliothek installieren, bietet jEdit eine unglaubliche HTML-Unterstützung.

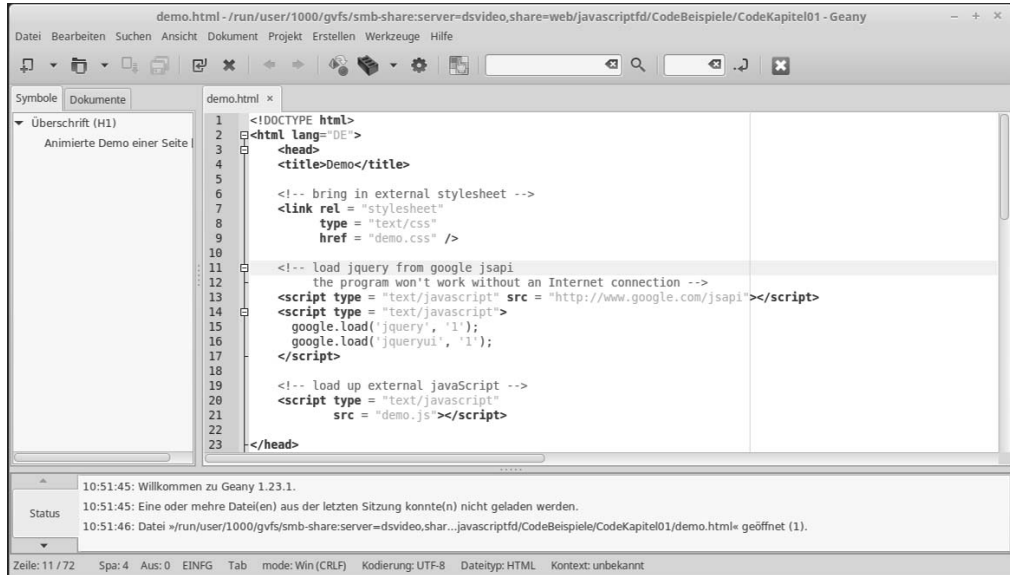
Vielleicht sind bei Ihnen aber ja auch ganz andere Editoren im Werkzeugkasten gelandet. Bei mir ist das beispielsweise Scite und UltraEdit gelungen. Und natürlich landen Sie bei der allgemeinen Suche im Internet fast unweigerlich auf Bestenlisten, die Ihnen weitere Vorschläge machen können.

## Geany

Den Texteditor Geany gibt es bereits seit etlichen Jahren. Er verfügt über die Funktionen einer integrierten Entwicklungsumgebung, und es gibt ihn in Varianten für Linux, OS X und Windows. Da er bereits einige Jahre erhältlich ist, hat er den Vorteil, dass er sich bei vielen Linux-Versionen entweder direkt in den Paketquellen befindet, oder sich zumindest über einschlägige Ressourcen leicht nachinstallieren lässt.

Die schnelle und leichtgewichtige integrierte Entwicklungsumgebung verfügt über grundlegende Funktionen wie Syntaxhervorhebung sowie Code-Vervollständigung und kann mehrere Dateien gleichzeitig in Registerkarten öffnen. Eingehender informieren können Sie sich beispielsweise auf [www.geany.org](http://www.geany.org) oder auch bei Wikipedia. Geany unterstützt neben HTML,

PHP und C einige Dutzend Programmiersprachen, und auch die Oberfläche der IDE wurde in verschiedene Sprachen übersetzt. Abbildung 1.9 zeigt Geany mit einer geladenen Datei, über die unten in der Statuszeile einige interessante Informationen angezeigt werden.



**Abbildung 1.9:** Geany zeigt die Datei `demo.html` an (Linux Mint). Der Dateiname in der Titelleiste verrät eine Menge.

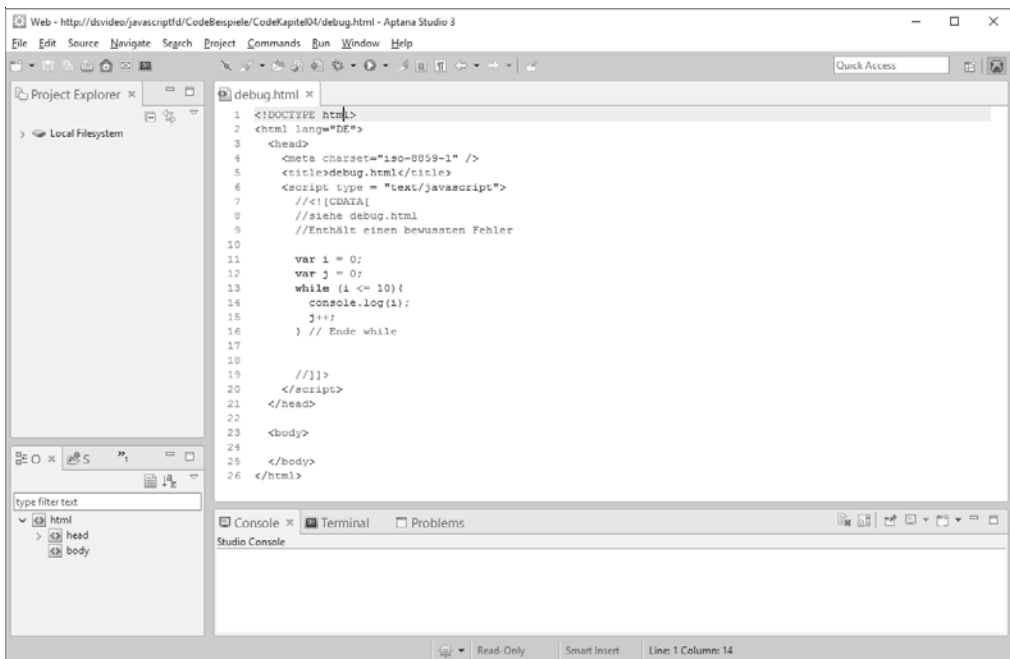
## Aptana

Aptana konnte als Editor für Programmierer die Welt der Webentwicklung erobern. Dieser vollständig ausgestattete Editor für Programmierer basiert auf dem leistungsstarken und bekannten Editor Eclipse (hauptsächlich eingesetzt bei der Java-Programmierung). Bei Aptana gibt es einiges, das empfehlenswert ist:

- ✓ **Umfangreiche, eingebaute Unterstützung für Websprachen:** Aptana unterstützt standardmäßig HTML/XHTML, CSS, JavaScript und AJAX.
- ✓ **Syntaxhervorhebung:** Die meisten Editoren für Programmierer unterstützen Syntaxhervorhebungen, aber Aptana ist hierbei besonders leistungsfähig. Es kann vorkommen, dass Sie dasselbe Dokument gleichzeitig in drei verschiedenen Sprachen geöffnet haben; dabei erkennt Aptana aus dem Kontext, ob Sie CSS-, XHTML- oder JavaScript-Code schreiben.
- ✓ **Code-Vervollständigung:** Dies ist eine der beeindruckendsten Funktionen von Aptana. Wenn Sie mit dem Schreiben einer Codezeile beginnen, lässt Aptana ein Menü mit Vorschlägen erscheinen. Dies hilft Ihnen, Fehler zu vermeiden, und Sie müssen nicht alle CSS-Attribute und JavaScript-Anweisungen im genauen Wortlaut auswendig kennen.

- ✓ **Entdecken von Fehlern:** Aptana kann sich Ihr Dokument anschauen, während Sie es erstellen, und in Echtzeit Bereiche hervorheben. Diese Funktion kann Ihnen dabei helfen, einen besseren Code zu schreiben, was wiederum dazu führt, dass sich Ihre Kenntnisse verbessern.
- ✓ **AJAX-Unterstützung:** AJAX ist eine Technologie, die nicht von allen Editoren direkt unterstützt wird. Aptana weist eine Reihe von Funktionen auf, die Ihnen bei AJAX helfen, zum Beispiel eine eingebaute Unterstützung aller größeren AJAX-Bibliotheken.

Aptana ist in der Standardversion vollständig kostenlos. Wie Aptana gestartet aussieht, zeigt Abbildung 1.10.



**Abbildung 1.10:** Aptana zählt zur Oberklasse der kostenlosen Editoren.



Leider hatte das Installationsprogramm in einer ganzen Reihe von Versionen einen Fehler. Es startete zwar unter Windows, meldete aber einen CRC-Fehler. Dieser tritt auf allen getesteten Windows-Rechnern auf und lässt sich umgehen, indem das Installationsprogramm über die mit der Option `ALS ADMINISTRATOR AUSFÜHREN` gestarteten Eingabeaufforderung mit der folgenden Option gestartet wird:

```
Aptana_Studio_3_Setup_x.x.x /passive /norestart
```

Ersetzen Sie den Dateinamen durch den der von Ihnen tatsächlich heruntergeladenen Datei.



Es gibt auch Versionen von Aptana mit deutschsprachiger Oberfläche. Da die Versionsnummer möglicherweise hinterherhinkt, müssen Sie selbst entscheiden, welche Version Sie nutzen wollen.

## Aufbau einer Browser-Sammlung

Webseiten leben in Webbrowsern. Jeder Browser interpretiert HTML und CSS ein wenig unterschiedlich, und diese Unterschiede werden deutlich, wenn Sie beginnen, über JavaScript und AJAX zu sprechen. Kleinere (und manchmal auch weniger kleine) Unterschiede in der Art der Unterstützung Ihres Codes durch den Browser können sehr wichtig werden.

### Einen Standard einrichten

Jeder Browser hat seine eigene Art, Webseiten anzuzeigen. Obwohl sich die dabei eingeschlagenen Wege ähneln, können die Unterschiede ab und an doch zu Schwierigkeiten führen. Und wenn Sie dann mit dem Schreiben von JavaScript-Code beginnen, finden Sie auch noch heraus, dass jeder Browser den Code anders interpretiert. Das kann zu einem ganz schönen Kuddelmuddel führen.

Glücklicherweise schreitet die Standardisierung seit einigen Jahren voran. Die Entwickler der verschiedenen Browser haben sich zusammengesetzt und darauf geeinigt, dass ein zentrales Team, das das *World Wide Web Consortium* (W3C) genannt wird, die notwendigen Standards erlässt. Wenn ein Browser Unterstützung für JavaScript enthält, akzeptiert er (zumindest theoretisch) ein Standardverhalten. Und solange Ihr Code denselben Standards folgt, können Sie davon ausgehen, dass alles sauber funktioniert.



Ich befolge in diesem Buch nur die JavaScript-Standards, wie sie von den meisten Entwicklern umgesetzt werden. Der hier vorgestellte Code wurde anfangs noch im Internet Explorer 9 für Windows und in Firefox 9 für Windows und Linux geschrieben. Leider kann der Code manchmal doch zu Problemen führen (die nicht unbedingt Fehler sind), auf die ich aber so weit wie möglich hinweise.



Auch Browser unterstützen nur bestimmte Betriebssystemversionen und werden für ältere Plattformen nicht mehr weiterentwickelt; somit erhalten Sie keine weitere technische Unterstützung mehr. Bis hin zu Windows Vista und Firefox sind davon bei den meisten Browsern alle Versionen unter Windows betroffen. Inwiefern das für Sie eine Rolle spielt, müssen Sie selbst entscheiden. Empfehlenswert sind die neuen Versionen insbesondere unter Windows aus Sicherheitsgründen.

### Sich für einen oder zwei Browser entscheiden

Wenn Sie für das Internet Seiten oder Programme entwickeln, sollten Sie ein paar Dinge über die gängigen Browser wissen:

- ✓ **Ältere Browser:** Im Internet sind weiter ältere Browser im Einsatz. Einfache Benutzer setzen weiter ihre alten Rechner ein, mit denen sie alle benötigten Aufgaben erledigen können, und sind schlicht bei dem Browser und den Programmen geblieben, die sich auf dem Computer befanden, als sie das gute Stück erhalten haben. Browser vor IE6 oder den ersten Firefox-Versionen werden heute aber zunehmend problematisch, weil immer weniger Webseiten ohne die modernen multimedialen Möglichkeiten auskommen. Diese



neueren Webstandards und JavaScript werden aber von älteren Browsern allenfalls bedingt unterstützt. Mit HTML5 und CSS sollten die eingesetzten Browser heute schon umgehen können. Und da es in diesem Buch um das Thema JavaScript geht, müssen schon etwas modernere Browser verwendet werden. Das könnten neben Microsoft Edge (oder dessen Vorgänger, dem Microsoft Internet Explorer 9) beispielsweise aktuelle Versionen von Mozilla Firefox, Apple Safari oder Google Chrome sein.



AJAX funktioniert in wirklich alten Browsern auf keinen Fall. Die Unterstützung von HTML5 begann in den Internet-Browsern etwa 2012. Neuer sollten die verwendeten Browser daher auf jeden Fall sein. Vergessen Sie dabei besser auch nicht die Android-Versionen für Tablets und Smartphones ([https://de.wikipedia.org/wiki/Liste\\_von\\_Android-Versionen](https://de.wikipedia.org/wiki/Liste_von_Android-Versionen)).

- ✓ **Microsoft Internet Explorer bis Version 6** : Der Microsoft Internet Explorer 6 war lange der führende Browser im Internet, sollte sich heute aber auf dem Altenteil befinden. Insbesondere errang dieser Browser dadurch Bekanntheit, dass er Funktionen besaß, die nicht den Standards der Community entsprachen. Mit HTML5 nahm die Verbreitung dieses Browsers stark ab. In Europa wurde das unter Windows vorinstallierte Programm danach neben Firefox oftmals zum Zweitbrowser.
- ✓ **Microsoft Internet Explorer ab Version 7**: Diese Versionen des Internet Explorers können als eine Art Anerkennung der Leistungen der Firefox-Entwickler betrachtet werden, weil sie viele Funktionen und Verbesserungen aufweisen, die schon Firefox hatte. Allerdings litt der Internet Explorer weiterhin unter der unvollständigen Unterstützung der Standards, auch wenn es merkliche Verbesserungen gab.
- ✓ Mit **Microsoft Edge** und Windows 10 scheint Microsoft vieles richtig gemacht und bei den Browsern wieder aufgeschlossen zu haben.
- ✓ **Firefox**: Der bereits erwähnte Browser Firefox von Mozilla eröffnete wieder die sogenannten »Browserkriege« und wurde weltweit schnell zu mehr als einer echten Konkurrenz für den Microsoft Internet Explorer. Für Entwickler war Firefox einer der ersten Browser, der ernsthaft den Webstandards entsprach. Firefox ist wegen seiner erweiterbaren Architektur gerade für Entwickler wichtig, weil es die Möglichkeit gibt, Firefox zu einem hochwertigen Entwicklungswerkzeug zu machen. Schauen Sie sich den nächsten Abschnitt dieses Kapitels an, in dem Sie einige großartige Vorschläge für Firefox-Erweiterungen finden.
- ✓ **Opera**: Opera ist ein wichtiger Browser, weil er einer der ersten war, der Webstandards aktiv unterstützte. Lange war er in gewissen Kreisen sehr beliebt, konnte sich aber nicht allgemein durchsetzen, zumal dann auch die Browser von Google zunehmend Verbreitung fanden. Da Opera die Webstandards unterstützt, sollte es allerdings mit jedem Code umgehen können, der für standardkonforme Browser geschrieben wird.
- ✓ **Safari**: Bei Apple Safari handelt es sich um das Webbrowser-Paket, das zusammen mit Mac OS ausgeliefert wird. Safari ist ein ausgezeichnete standardkonformer Browser. Zwischenzeitlich gab es auch Windows-Versionen von Safari. Die Webbrowser auf iPhone

oder iPad verwenden dieselbe Engine wie Safari, was Sie nicht vergessen dürfen, wenn Sie Anwendungen für Mobilgeräte erstellen.

- ✓ **Chrome:** Dieser Browser wurde von Google entwickelt. Er entspricht den Standards von allen Browsern wohl am besten und ist besonders leistungsstark beim Umgang mit JavaScript und AJAX. Dies überrascht nicht, wenn man bedenkt, dass Google eine der Firmen ist, die Pionierarbeiten bei der Verwendung von AJAX geleistet haben und dieses aktiv fördern. Chrome ist für seinen schnellen JavaScript-Interpreter bekannt. *Chromium* bildet die Basis des Chrome-Browsers, die von Google als Open-Source-Projekt zur Verfügung gestellt wird. Chromium ist daher auch für Linux erhältlich, muss da aber – aus lizenzrechtlichen Gründen – auf einige Bestandteile von Chrome verzichten.
- ✓ **Andere Browser:** Es gibt noch viele weitere Browser, die heutzutage eingesetzt werden, zu denen insbesondere auch Browser für spezielle Formen von Linux, Smartphones oder Tablets gehören. Es ist so gut wie unmöglich, alle vollständig zu unterstützen; glücklicherweise orientieren sie sich aber heute überwiegend an den Webstandards. Nennenswert ist vielleicht noch *Dolphin*, das sich aber auf zunehmend schwerem Stand gegenüber Googles Browser und/oder Firefox befindet. Auf einigen Erfolg kann auch der *UC Browser* zurückblicken, allerdings vorwiegend im Mobilbereich und dort insbesondere in China und Indien.



Eigentlich sollte ich bei den Browsern ausführlicher auf die Unterschiede zwischen den Browser-Engines (auch HTML-Renderer genannt) eingehen. Dann käme ich aber auf Themen wie mit Gecko, WebKit, Presto, Blink, Trident und EdgeHTML zu sprechen, was nicht wirklich hilfreich ist, wenn man einmal davon absieht, dass Sie ein wenig über die Verwandtschaft der Browser erfahren würden. Bei Interesse hilft eine Internet-Suche weiter.



Ich selbst habe die meisten meiner Tests mit Firefox durchgeführt, weil der über eine sehr gute Unterstützung der Standards und exzellente Werkzeuge für das Verbessern und Debuggen von Code verfügt. Danach überprüfe ich meinen Code üblicherweise mit Browsern wie dem Internet Explorer oder Edge und Chrome oder Chromium.

## Firefox zur Entwicklungsmaschine machen

Der Browser Firefox ist für Webentwickler ausgesprochen wichtig. Er besitzt eine Reihe attraktiver Funktionen, zu denen auch die hervorragende Unterstützung von HTML und JavaScript gehört. Aber der wichtigste Vorteil von Firefox als Werkzeug für Entwickler besteht wohl in der Unterstützung von Erweiterungen (Add-ons). Viele der gängigen Browser behandeln ihren Code wie einen geheimen Schatz, und sie lassen sich nur schlecht und schwer erweitern. Firefox wurde von Anfang an so entwickelt, dass seine Fähigkeiten erweiterbar sein sind. Diese Chance wird von vielen cleveren Programmierern auch genutzt. Und dabei haben sich einige der Erweiterungen derart gemausert, dass sie für Entwickler unentbehrlich geworden sind.

## Das Firefox-Add-on »Web Developer«

Die Erweiterung WEB DEVELOPER von Chris Pederick fügt unglaubliche Funktionen zu Firefox hinzu, die den Browser um eine ganze Reihe ausgesprochen nützlicher Möglichkeiten erweitert.

Sie können »Web Developer« als Firefox-Add-on installieren. Anschließend erreichen Sie die verschiedenen Menüs über die eingeblendete Leiste des Add-ons:

- ✓ **CSS bearbeiten:** Sie können ein schmales Fenster anzeigen lassen, um dort CSS-Code einzugeben. Das CSS-Änderung zeigt sich sofort auf dem Bildschirm, wodurch Sie unmittelbar erkennen können, was sie auf einer Seite bewirkt.
- ✓ **Lineal anzeigen:** Mit diesem ausgesprochen praktischen Werkzeug aus dem Menü SONSTIGES können Sie Bereiche um die Objekte einer Webseite herum aufziehen und dadurch pixelgenaue Angaben zur Größe der Objekte erhalten. Dieses Werkzeug ist besonders beim Debuggen von Weblayouts nützlich.
- ✓ **Tabelleninformationen anzeigen:** Dieses Werkzeug ist bei Layouts sinnvoll, die auf Tabellen basieren. Es bietet eine gute Möglichkeit zu sehen, wie ein komplexer, auf Tabellen basierender Entwurf erstellt wird.



Wenn sich das benötigte Layout anderweitig realisieren lässt, vermeiden Sie auf Tabellen basierende Entwürfe am besten. Wenn Sie sich Seiten anderer Entwickler ansehen, ist diese Funktion aber durchaus nützlich.

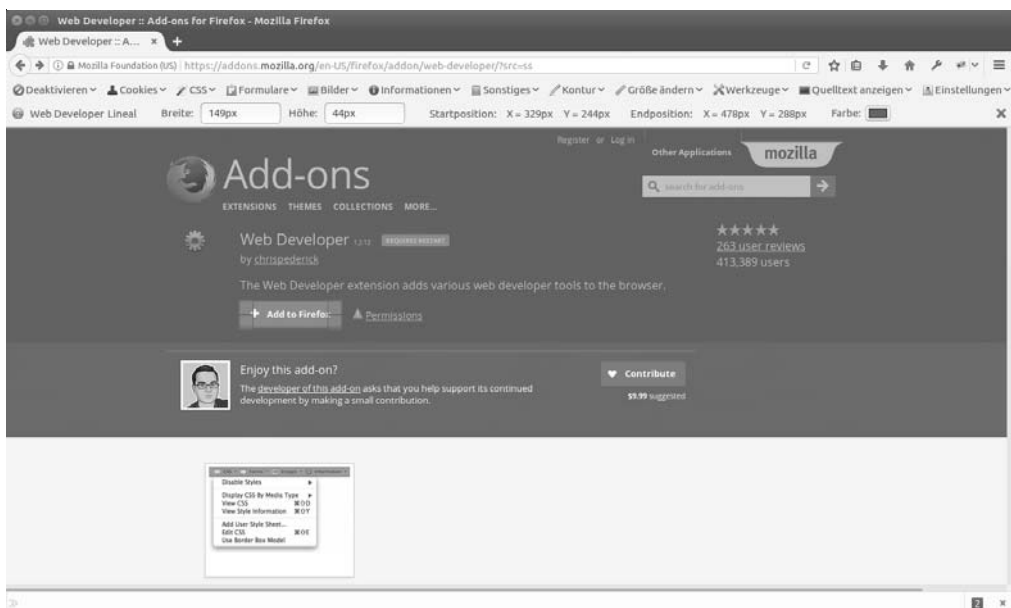


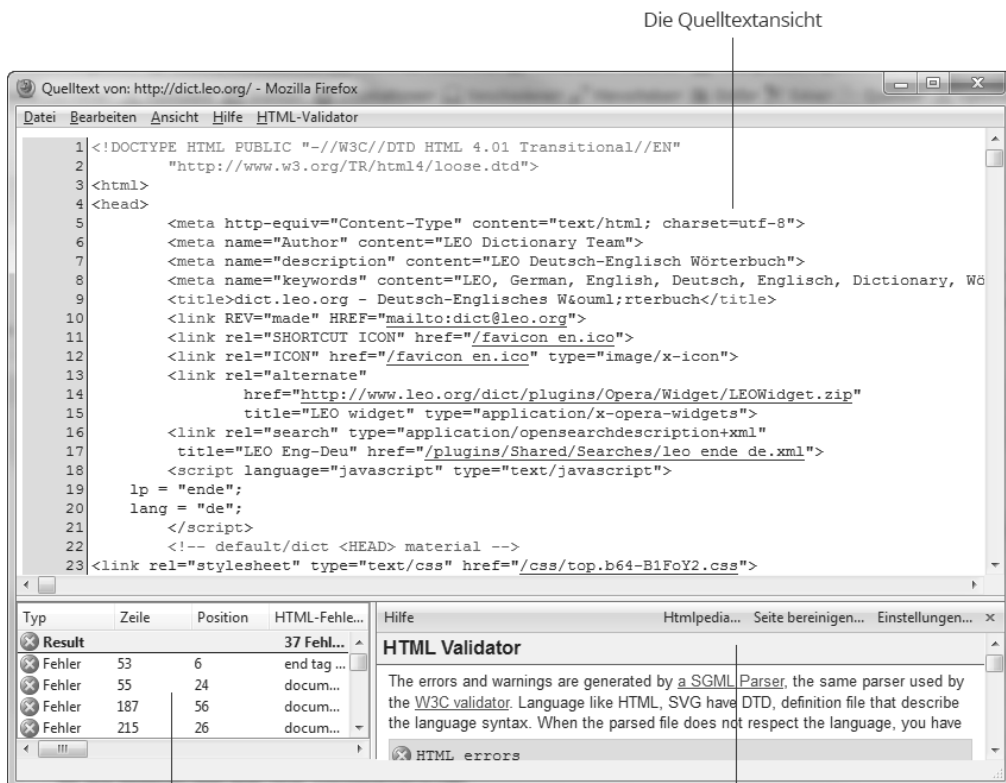
Abbildung 1.11: So ähnlich sieht es aus, wenn Sie »Web Developer« als Add-on installiert haben.

- ✓ **Größe ändern:** Über das Menü GRÖSSE ÄNDERN können Sie sich anzeigen lassen, wie Ihre Seite in einer anderen Größe aussieht. Beim Seitenentwurf kann dies sehr nützlich sein.
- ✓ **Validierungswerkzeuge:** Im Menü WERKZEUGE gibt es einige wirklich praktische Werkzeuge für die Validierung Ihrer Webseite. Hier gibt es unter anderem auch Optionen für die Validierung von HTML und CSS.

## HTML Validator

Diese Erweiterung holt dieselbe Validierungs\_Engine auf Ihren Browser, die der W3C verwendet hat. Sie liefert eine schnelle Rückmeldung über die Gültigkeit der einzelnen von Ihnen betrachteten Seiten. Außerdem erweitert der HTML-Validator die Quelltextanzeige von Firefox um zusätzliche Informationen, zu denen unter anderem Angaben über die gemachten Fehler gehören.

Die Hinweise zur Fehlerbehebung können wirklich hilfreich sein, und es gibt sogar ein Werkzeug, um den Code mit dem ausgezeichneten Programm *HTML Tidy* automatisch reparieren zu lassen.



In diesem Fenster werden die Validierungsfehler angezeigt.

Die Erweiterung Validator sorgt für nützliche Fehlermeldungen.

**Abbildung 1.12:** HTML-Validator ergänzt die Quelltextansicht um nützliche Funktionen.



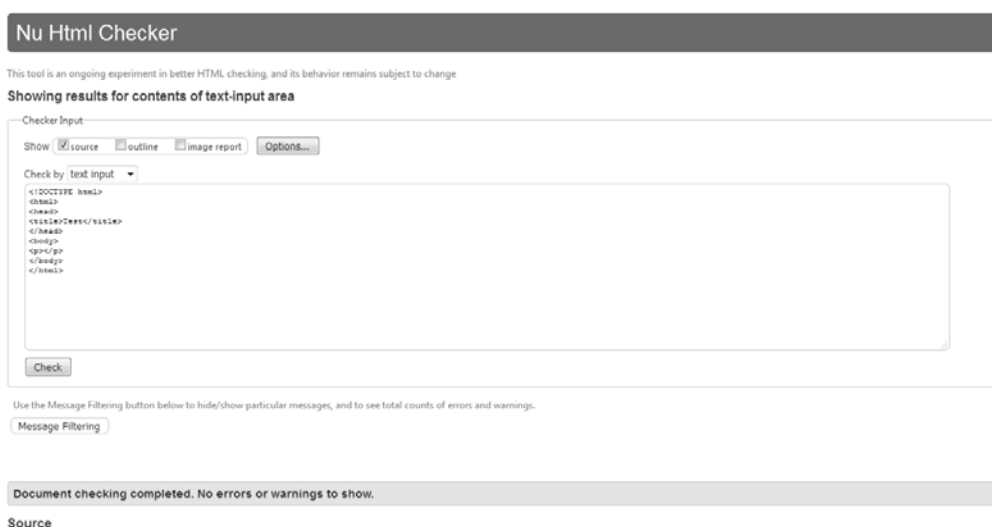
Leider eignet sich der HTML-Validator bisher immer noch nicht für den Sprung auf HTML5. Nach der Wahl von »HTML Tidy« als Überprüfungsalgorithmus wird zwar XHTML- und älterer HTML4-Code gut verarbeitet, ein HTML5-Header produziert aber reichlich Fehlermeldungen.

Auf HTML gründet sich Ihr JavaScript-Code, und ungültiges HTML sorgt für fehlerhafte Rahmenbedingungen. Abbildung 1.12 zeigt das verbesserte Anzeigefenster für den Quellcode in Firefox, das nun Bereiche für die Fehler und für eine Fehlerbeschreibung mit Einstellungsmöglichkeiten erhalten hat.

## Online-Validierung

Als Alternative zur lokalen Überprüfung von Quelltexten können über das Web bereitgestellte Möglichkeiten genutzt werden. Über die Adresse <https://validator.w3.org> können Sie Ihren HTML-Code beim *W3C Markup Validation Service* validieren lassen. Unter dieser Adresse müssen Sie selbst gültige Internet-Adresse der Dateien eingeben, die validiert werden sollen. Für ambitionierte Programmierer sollte das keine allzu große Einschränkung sein.

Erweitern Sie die angegebene Adresse zu <https://validator.w3.org/nu>, kommen Sie zu dem Werkzeug, das auch vom vorgestellten Firefox-Add-on *Web Developer* zur Überprüfung von HTML genutzt wird, dem *Nu Html Checker* (siehe Abbildung 1.13).



**Abbildung 1.13:** In den Nu Html Checker können Sie Dateien oder zu prüfenden Code hineinladen.

In der Abbildung 1.13 wird der präsentierte Code für das einfache Gerüst geprüft. Unter CHECK BY können Sie festlegen, woher der zu prüfende Code kommen soll. Sie können ihn auch einfach in den Eingabebereich auf dem Bildschirm kopieren.

## Firebug und Firefox-Debugger

Die Erweiterung Firebug ist eines der bekanntesten und über einen langen Zeitraum wichtigsten Werkzeuge für Webentwickler. Es machte aus Firefox ein vollständiges Werkzeug für die Fehlersuche (mit einem *Debugger*). Mit der Zeit wurden die ursprünglich von Firebug bereitgestellten Funktionen zunehmend in den Debugger von Firefox selbst integriert. Diesen finden Sie in den neueren Versionen von Firefox unter EXTRAS|WEB-ENTWICKLER|DEBUGGER in den Menüs.

Ob Sie Firebug als Add-on in Firefox installieren konnten, war abhängig von der eingesetzten Version. Ein installiertes Firebug verbirgt sich jedenfalls normalerweise im Menü EXTRAS von Firefox unter WEB-ENTWICKLER|FIREBUG und ist zudem über das Symbol einer Feuerwanne in der einen oder anderen Leiste zu erreichen.

Firebug hat die folgenden Eigenschaften:

- ✓ **Untersuchungsfenster:** In diesem Bereich auch Inspektor genannten Bereich können Sie die Maus über einem beliebiges Element auf Ihrer Seite bewegen, um den dazugehörigen Code und entsprechende Daten angezeigt zu bekommen. Dies bietet eine einfache Möglichkeit, um eine Seite zu analysieren und Fehler zu suchen. Außerdem sehen Sie sofort, welches Style auf ein Code-Element angewendet wird.
- ✓ **CSS-Ansicht und –Bearbeitung (Stilbearbeitung):** Sie können sich das CSS einer Seite in einem eigenständigen Fensterelement ansehen, erhalten eine Vorschau aller Farben und können die Werte bearbeiten. Die Ergebnisse werden in Echtzeit auf der Seite wiedergegeben.
- ✓ **Debugger:** Selbst Profis machen Fehler. Und JavaScript-Entwickler benötigen Werkzeuge für die Fehlersuche. Debugger verfügen über bessere Mechanismen für die Fehlersuche und können Sie bei der Fehlersuche unterstützen.
- ✓ **AJAX-Überwachung:** Die AJAX-Programmierung basiert auf einer Reihe von Anfragen vom und an den Server. Firebug konnte Ihnen dabei helfen, diesen Anfragen auf der Spur zu bleiben und das Verschieben der Daten zu beobachten.
- ✓ **Sofortige Code-Ansicht:** Die normale Quellcodeansicht der meisten Browser zeigt Ihnen den Code so, wie er ursprünglich vom Browser kommt. Beim Programmieren mit JavaScript ändern Sie oft den Code auf die Schnelle. Die Seite wird dann so angezeigt, wie sie wirklich aussieht, und das selbst dann, wenn sie von JavaScript geändert wird.

Abbildung 1.14 zeigt den *Firefox-Debugger* im Inspektionsmodus. Sobald Sie den Mauszeiger über ein Element der Webseite schweben lassen, wird der zugehörige Code grafisch hervorgehoben. Wie Sie sehen, entspricht er recht gut den Beschreibungen von Firebug (und auch den Debuggern von anderen Browsern).

Offiziell wurde die aktive Entwicklung von Firebug mittlerweile eingestellt, das Add-on lebt aber möglicherweise immer noch weiter. Auch was *Firebug Lite* angeht, bei dem es sich um eine abgespeckte Version von Firebug handelt, die in IE und anderen Browsern funktionierte,

muss man abwarten. Mit diesen Varianten konnten beispielsweise Chrome und Opera um die meisten Funktionen von Firebug erweitert werden.



**Abbildung 1.14:** Der Firefox-Debugger mit den Entwicklerwerkzeugen in einem eigenen Fenster

Wie dem auch sei, in jedem Fall ist es zu begrüßen, dass die Funktionen von Firebug in die integrierten Debugger der modernen Browser aufgenommen wurden.



An Firefox wurden für die Version 52 erhebliche interne Änderungen vorgenommen. Insbesondere über die Versionen 50 bis 52 wurde seither häufig berichtet, dass Add-ons gar nicht mehr oder nur teilweise funktionierten. Leider sind davon auch einige der hier im Buch vorgestellten Dinge betroffen.