# Umstieg auf HTML5

1

### In diesem Kapitel ...

- Die Geschichte von HTML
- Was ist HTML5?
- Unterstützte Browserfunktionen testen
- Auswahl eines passenden Browsers
- Die unterstützten Funktionen im IE mit Chrome Frame erweitern

*HTML5* ist die neueste Version der HTML-Sprachfamilie. HTML (HyperText Markup Language) ist einer der Hauptgründe für die Leistungsfähigkeit und Nützlichkeit des Webs. Dabei handelt es sich um ein recht einfaches System von Codes in Textdateien, die für die Struktur aller Webseiten im Internet sorgen.

In diesem Kapitel werden wir uns kurz ansehen, wie sich HTML5 in die Geschichte des Webs eingliedert und ein paar Werkzeuge zusammenstellen, die Sie benötigen, um loslegen zu können.

Laden Sie sich den Beispielcode unter der Adresse `www.downloads.fuer-dummies.de` herunter, um ihn lokal ausführen zu können.

## Die Geschichte von HTML in aller Kürze

HTML ist ein wesentlicher Bestandteil des Internets und kann bereits auf eine kurze, aber höchst abwechslungsreiche Geschichte zurückblicken. Um verstehen zu können, worum es bei HTML5 geht, ist es nützlich, einen Blick auf seine Herkunft zu werfen. Der Wandel des Internets (und insbesondere des Webs) hat sich mit ungeheurer Geschwindigkeit vollzogen. Und HTML versucht dabei mitzuhalten.

Im ersten Entwurf umfasste HTML eine Handvoll Tags und legte wirklich kaum mehr als das Aussehen der Seite fest. Mit der fortschreitenden Entwicklung des Webs kamen viele Funktionen hinzu. Im modernen Internet geht es zwar weiterhin um Dokumente, aber auch um Anwendungen. Heutige Websites sind dynamische, interaktive *Apps*.

Und auch die Art der Geräte, mit der wir das Internet nutzen, hat sich verändert. Früher konnten wir das Web nur mit Desktoprechnern nutzen. Heute spielen Mobiltelefone und andere Mobilgeräte eine wichtige Rolle bei der Webnutzung. Sie erfordern andere Denkansätze als die alten, an den Schreibtisch gebundenen riesigen Geräte, die noch vor ein paar Jahren vorwiegend für diese Zwecke genutzt wurden.

Damit wird es Zeit für einen frischen Satz von Standards, die der heutigen Art der Internetnutzung gerecht werden und diese unterstützen. HTML5 stellt diesen Satz von Standards zur Verfügung.

## Ein wenig Frühgeschichtliches

Im fernen Nebel der Vergangenheit (1989) entwickelte Tim Berners-Lee ein neues System, mit dem elektronische Dokumente miteinander verknüpft werden konnten. Er entwarf eine einfache Sprache, mit der Autoren verschiedene Dokumente mithilfe einer beschränkten Menge von Formatierungsoptionen miteinander verknüpfen (verlinken) konnten. Diese Sprache wurde HTML genannt.

Zu der Zeit gab es das Internet zwar bereits, der Zugriff darauf erfolgte aber noch vorwiegend über einfache Befehlszeilenprogramme. Daher war es auch nicht gerade leicht nutzbar. Bei der Entwicklung von HTML (und einigen anderen zugrunde liegenden Technologien) stand von Anfang an die einfache Nutzung im Vordergrund und die erstellten Dokumente sollten von den Benutzern leicht gewartet und verwaltet werden können. Das HTML-Design wurde absichtlich einfach gehalten, damit sich möglichst viele Leute am Prozess der Erstellung von Dokumenten in diesem neuen Format beteiligen konnten.

Bekanntermaßen begann damit der enorme Siegeszug des Webs, das schon bald allgegenwärtig war. Schon bald wurde aber auch klar, dass die einfachen Funktionen aus dem HTML-Basisbefehlssatz nicht ausreichen würden, um allen Anforderungen der vielen Anwender gerecht werden zu können, die mittlerweile Webseiten erstellten.

## Und der erste Browserkrieg brach aus ...

Als nach und nach verschiedene Unternehmen mit der Erstellung von Webbrowsern und damit Programmen begannen, die HTML-Dateien lesen und anzeigen konnten, fügten sie neue HTML-Funktionen hinzu, um sich Vorteile vor den Konkurrenten zu verschaffen. 1993 konnten dann mit dem Mosaic-Browser Bil-

der auf Webseiten angezeigt werden (das hatte die ursprüngliche Spezifikation nicht vorgesehen). Überall auf der Welt wurden Browser von kleinen Teams programmiert, die ihnen jeweils einen eigenen Satz neuer Funktionen verpassten.

1994 kristallisierte sich eine dominante Browserplattform heraus. Der Netscape Navigator feierte als Browser immense Erfolge. Gleichzeitig bildeten sich Arbeitsgruppen, um sich mit dem Problem der fehlenden Standards im Webbrowserbereich zu befassen. Die wichtigste dieser Gruppen war das W3C (World Wide Web Consortium), das von Tim Berners-Lee und damit eben jenem Typ geleitet wurde, mit dem das ganze Durcheinander im Web begonnen hatte. Netscape war jedoch derart dominant, dass dessen Repräsentanten den Tagungen zur Standardisierung häufig fernblieben. Netscape verwirklichte seine Wunschfunktionen einfach auf eigene Faust.

Microsoft spielte im Browserbereich bis 1995 keine Rolle. Der Internet Explorer (IE) wurde als direkter Konkurrent zum Netscape-Browser konzipiert. Eine Weile (während der Zeit, die auch als »erster Browserkrieg« bezeichnet wird) lieferten sich Netscape und Microsoft ein Kopf-an-Kopf-Rennen und versuchten jeweils exklusive Funktionen anzubieten, um Entwickler von der eigenen Vision des Webs zu überzeugen.

Es gab zwar ein Gremium zur Festlegung von Standards, praktisch ergänzten Netscape und Microsoft aber einfach ihre gewünschten Funktionen und ignorierten dabei im Prinzip das W3C. Hinsichtlich der Webstandards gab es einige kleine Fortschritte. HTML 2 wurde 1994/1995 als Standard verabschiedet (aber von keinem der Hersteller umfassend umgesetzt). HTML 3.2 wurde 1997 veröffentlicht und im Frühjahr 1998 folgte HTML 4.

Etwa zur selben Zeit, als HTML 4 Fuß fassen konnte, wurde die Dominanz von Microsoft im Browserbereich deutlich. 2002 benutzten etwa 95 Prozent der Internetnutzer den Internet Explorer. Angesichts dieser Dominanz lag die Zukunft von HTML nahezu völlig in Microsoft-Händen. Die Bemühungen der Standardisierungsgremien wurden damit weitgehend irrelevant. Microsoft hatte den ersten Browserkrieg in jeder Hinsicht gewonnen. Der (vorrangig auf HTML 4 aufbauende) Internet Explorer 6 war der einzige Browser, der noch eine echte Rolle spielte. Daraufhin gab es über etliche Jahre hinweg kaum noch Innovationen.

## Ein neuer Herausforderer erhebt sich aus der Asche

Einige neue Browser stellten die Microsoft-Dominanz jedoch auf die Probe. Insbesondere der erstmals 2004 erschienene Firefox-Browser muss hier erwähnt werden, da er mit einer Reihe innovativer Funktionen aufwarten konnte und

sich an die meisten Standards der W3C-Arbeitsgruppe hielt. Firefox (und in geringerem Ausmaß andere Browser wie Apples Safari, Opera und schließlich Google Chrome) rüttelten das Web wach. Diese Browser legten tendenziell wesentlich mehr Wert auf die Einhaltung der Standards als der IE und führten nach einer längeren Zeitspanne der Stagnation dazu, dass auch wieder neue Versionen des Internet Explorer entstanden. Zumindest in Lippenbekenntnissen bekannte sich jetzt auch Microsoft zu Standards und versprach, diese in den kommenden IE-Versionen verstärkt einzuhalten. Teilweise wird dies als der Beginn des zweiten Browserkriegs betrachtet, in dem es um den Kampf verschiedener Anbieter um Anteile des Browsermarkts geht.

Diesmal gibt es jedoch einen entscheidenden Unterschied. Das Web ist mittlerweile nicht mehr neu, sondern zu einem wesentlichen Bestandteil der Geschäftswelt und Gesellschaft geworden. Dokumente im Web werden nun nach ähnlichen Kriterien wie gedruckte Dokumente beurteilt und mit HTML 4 lassen sich die entsprechenden Standards nicht leicht erfüllen. Tatsächlich steht die gesamte Idee des Webs als Ansammlung von Dokumenten gewissermaßen auf dem Prüfstand. Web*seiten* werden durch Web*anwendungen* (Applikationen oder Apps) ersetzt. Viele der heutigen Aktivitäten im Internet haben nicht mehr viel mit dem Lesen von Dokumenten zu tun. Und Entwickler benutzen heute das Web selbst als eine Programmierschnittstelle.

## HTML 4 wurde langsam alt

Durch den Wandel des Webs wurde es notwendig, die Standards für Dokumente zu überdenken. Es wurde klar, dass HTML 4 die moderne Webentwicklung nicht mehr hinreichend unterstützen konnte. Die verschiedenen, über die Jahre hinweg ergänzten proprietären Tags hatten zwar ein wenig mehr Flexibilität gebracht, aber das reichte nicht annähernd aus. Es fehlten befriedigende Lösungen für das Seitenlayout oder den Umgang mit Schriftarten. Es gab zwar eine Reihe von Funktionen für die Eingabe von Formulardaten, der war aber zu beschränkt und in der Handhabung zu sperrig. Die meisten Browser enthielten zwar Varianten der Programmiersprache JavaScript, dessen Implementierung fiel aber höchst unterschiedlich aus. Mit Webtechnologien echte Anwendungen zu erstellen, wurde daher zu einem recht riskanten Vorhaben.

Um einige dieser Probleme zu beheben, stellte das W3C 2002 XHTML vor. Als neue HTML-Version sollte sich XHTML an den strengeren Standards der Auszeichnungssprache XML (eXtensible Markup Language) orientieren. XHTML ist weit weniger fehlertolerant als HTML. Wenn daher eine Seite die stringenten Anforderungen des Standards erfüllt, sollte sie (vermutlich) allgemein keine

Probleme bereiten und voraussagbare Resultate liefern. Praktisch wurde das idealistische Projekt der XHTML-Bewegung leider nie verwirklicht. Die Erstellung valider XHTML-Dokumente erwies sich als derart schwierig, dass sich nur sehr wenige Entwickler überhaupt daran versuchten. Von den Browsern wurde selbst mangelhafter XHTML-Code noch halbwegs vernünftig (wenn nicht sogar perfekt) dargestellt. Tatsächlich arbeiteten die meisten Browser eigentlich nicht einmal mit XHTML, sondern wandelten es insgeheim irgendwie in HTML um. Daher gab es für Entwickler auch kaum Anreize, sich an die XHTML-Standards zu halten.

Um in HTML fehlende Funktionen nachzurüsten, nutzten viele Entwickler Plug-in-Technologien wie eingebettete Java- und Flash-Applets. Java bleibt zwar in anderen Bereichen äußerst wichtig, konnte aber als clientseitige Umgebung nie richtig Fuß fassen. Flash hingegen war eine Weile recht beliebt. Leider verursacht es selbst wieder Probleme. Der Inhalt von Flash-Applets lässt sich nur mit Flash-Editoren ändern und ist für Suchmaschinen nicht oder zumindest nicht leicht lesbar. Viele der neuen HTML5-Funktionen (insbesondere die Unterstützung von Schriftarten und das `canvas`-Tag) kann man als direkte Antwort auf Flash betrachten.

Das W3C setzte seine Bemühungen zwar fort und erstellte mit XHTML 2.0 eine neue XHTML-Variante, aber in der Zwischenzeit begann eine zweite Gruppe namens *WHATWG (Web Hypertext Application Technology Working Group)* mit der Arbeit an einem eigenen konkurrierenden Standard, der unter der Bezeichnung *HTML5* bekannt wurde. Der Hauptgrund für diese konkurrierenden Standards besteht darin, dass der Eindruck entstand, XHTML sei zu starr und würde sich weiterhin zu sehr auf den Gedanken von HTML als Seitenbeschreibungssprache konzentrieren. Die Entwicklung von HTML5 wurde teilweise dadurch angeregt, dass ein Rahmen (ein Framework) für die Erstellung von Webanwendungen entstehen sollte, den Entwickler auch wirklich nutzen würden. Schließlich ließ das W3C XHTML 2 fallen und unterstützt nun die WHATWG-Vorschläge, weshalb es sich bei HTML5 wohl um den nächsten kommenden Standard handeln dürfte.

## Das eigentliche HTML5 kennenlernen

Die WHATWG-Gruppe scheint aus der Geschichte einige Lektionen gelernt zu haben. Das Design von HTML5 weist auf die folgenden Schwerpunkte hin:

- ✔ **Die Kernsprache sollte einfach sein.** In HTML5 geht es ein wenig aufgeräumter zu als in XHTML. Insbesondere beim Dokumenttyp kam ein wenig

frischer Wind rein, wenn man ihn mit dem Unsinn vergleicht, den man am Anfang einer XHTML-Seite schreiben muss. Alle Tags beschreiben irgendein Merkmal der Seite. Bei den meisten Tags handelt es sich um englische Ausdrücke, die von ein paar Abkürzungen ergänzt werden.

- ✔ **Die Auszeichnungen basieren auf der Semantik.** Einer der ursprünglichen HTML-Ansätze bestand in Auszeichnungen, die auf der *Bedeutung* (der Funktion) und nicht auf *Details* basieren sollten. Beispielsweise wird eine Überschrift (headline) einfach mit `<h1>` ausgezeichnet, ohne dass eine bestimmte Schriftgröße oder Schriftart angegeben wird. HTML5 kehrt zu dieser Tradition zurück und fügt eine Reihe neuer Tags zur Beschreibung gängiger Teile von Seiten hinzu.
- ✔ **CSS wird für die Details der Formatierung verwendet.** Wie XHTML stützt sich auch HTML5 beim Umgang mit Details zum Aussehen der jeweiligen Elemente stark auf eine andere Sprache, die den Namen *CSS (Cascading Style Sheets)* trägt. Letztlich beschreibt HTML, um *was* es sich bei einem Element handelt. CSS beschreibt hingegen, *wie* die jeweiligen Elemente aussehen sollten. HTML5 kennt keine Tags wie `<font>` oder `<center>`, weil CSS diese Merkmale flexibler handhabt.
- ✔ **Viele Webseiten sind Apps.** *Formulare* (jene Elemente, über die Benutzer Daten über Webseiten eingeben können) sind zwar von Anfang an Bestandteil von HTML, wurden aber über die Jahre hinweg kaum weiterentwickelt. Mit HTML5 kommen einige recht interessante neue Formularelemente hinzu, durch die sich HTML viel besser für Interaktionen mit den Nutzern eignet.
- ✔ **JavaScript steht im Mittelpunkt.** In den meisten Webbrowsern ist bereits seit Jahren eine Variante der Programmiersprache JavaScript (JS) integriert. Wegen etlicher Beschränkungen konnte man JavaScript bisher allerdings nicht wirklich ernst nehmen. Einige Beschränkungen bestanden wegen berechtigter Sicherheitsbedenken, während es sich bei anderen einfach um schlechte oder inkompatible Implementierungen handelte. Mit dem Entstehen neuer, leistungsfähiger JavaScript-Engines und einem neuen Paradigma namens *AJAX (Asynchronous JavaScript and XML)* ist JavaScript als leistungsstarke und wichtige Programmierumgebung auferstanden. Bei vielen der interessantesten HTML5-Funktionen (wie dem `canvas`-Tag) handelt es sich vorwiegend um Entwicklungen der JavaScript-Sprache. (Das `canvas`-Tag ist zwar ein HTML-Tag, mit dem sich ohne JavaScript aber nichts Interessantes anstellen lässt.)

# HTML5 ist mehr als HTML!

Irgendwie ist es ein wenig unglücklich, dass diese Technologie HTML5 genannt wurde, da die Sprache HTML eigentlich nur ein Rad in einem viel größeren Getriebe ist. Tatsächlich handelt es sich bei dem, was wir HTML5 nennen, um die Integration etlicher verschiedener Technologien (HTML, CSS, JavaScript und serverbasierte Technologien), deren Rollen nachfolgend beschrieben werden sollen.

## HTML

Natürlich hat es Änderungen an der Sprache HTML selbst gegeben. Einige kamen zum HTML-4-Standard hinzu und ein paar wurden entfernt. HTML5 bleibt jedoch zu HTML 4 abwärtskompatibel, weshalb es nicht zwingend notwendig ist, dass Ihr Code dem HTML5-Standard entspricht. Die Anpassung von HTML 4 an HTML5 ist wahrscheinlich die leichteste Übung bei der Umstellung auf die umfassende HTML-Sichtweise.

Dies sind die wichtigsten HTML-Eigenschaften:

- ✔ **Semantische Auszeichnung:** HTML5 beinhaltet neue Tags, die Teile von Dokumenten beschreiben. Nun gibt es spezielle Tags für Navigationselemente, Artikel, Abschnitte, Kopf- und Fußzeilen.
- ✔ **Neue Formularelemente:** HTML5-Formulare haben einige wesentliche Aktualisierungen erfahren. Es gibt verschiedene neue Varianten des `<input>`-Elements, über die Farben, Zahlen, E-Mail-Adressen und Datumsangaben von Benutzern leicht ausgewählt werden können.
- ✔ **Mediale Elemente:** Über Tags, die dem `<img>`-Tag ähneln, unterstützt HTML5 endlich auch Audio und Video nativ.
- ✔ **Das `canvas`-Tag:** Mit dem `canvas`-Tag können Programmierer Grafiken interaktiv erstellen. Mit dieser Fähigkeit lassen sich ziemlich faszinierende Dinge wie maßgeschneiderte Spiele und Schnittstellenelemente realisieren.

## CSS

Für die größten Umstellungsprobleme bei den HTML-4-Gewohnten dürfte wohl wirklich nicht HTML selbst, sondern die geänderte Beziehung zwischen HTML und CSS sorgen. In HTML5 (wie auch in XHTML) beschreibt die Auszeichnungssprache nur die Bedeutung der verschiedenen Elemente. CSS wird also wirklich

als Seitenbeschreibungssprache benutzt. Wenn Sie komplett auf HTML5 umsteigen wollen, dann müssen Sie sich von der Nutzung von Tags wie `<font>` und `<center>` verabschieden, die sich lediglich auf punktuelle Details beziehen. Bei HTML 4 lässt sich CSS noch als optionale Erweiterung betrachten, beim Umgang mit HTML5 spielt es aber eine zentrale Rolle. Wenn Sie sich bisher noch nicht mit CSS vertraut gemacht haben, wird es nun definitiv Zeit. Die Nutzung von CSS setzt zwar eine etwas andere Denkweise voraus, die dafür aber ungeheuer leistungsfähig und flexibel ist. Mit dem HTML5-Standard geht ein neuer CSS-Standard einher, der *CSS3* getauft wurde. Aufgrund der engen Wechselwirkung ist es beinahe unmöglich, über HTML5 zu sprechen, ohne dabei nicht auch CSS3 einzubeziehen. Dies sind die wesentlichen neuen CSS3-Funktionen:

- ✔ **Integrierte Unterstützung von Schriftarten:** Mit diesem lange erwarteten Werkzeug können Sie Schriftarten (Fonts) auf Webseiten ablegen und zu deren Darstellung nutzen. Die Fonts müssen dabei nicht mehr vom Anwender unter dem benutzten Betriebssystem installiert werden.
- ✔ **Neue Selektoren:** Selektoren werden zur Beschreibung von zu ändernden Codesegmenten verwendet. CSS3 unterstützt nun neue Selektoren, mit denen Sie jedes zweite Element und auch spezifische Unterelemente (beispielsweise verschiedene Arten von `input`-Tags) auswählen können.
- ✔ **Spalten:** Spalten wurden in HTML bisher nicht sinnvoll unterstützt. Das hat zu einer Menge mehr oder minder cleveren Lösungen geführt, mit denen versucht wurde, diesen Mangel zu überwinden. Endlich lassen sich mit CSS Elemente leicht in mehrere Spalte aufteilen.
- ✔ **Optische Verbesserungen:** CSS bietet etliche interessante neue Möglichkeiten wie etwa Transparenz, Schatten, abgerundete Ecken, Animationen, Farbverläufe (Gradienten) und Transformationen. Damit haben Sie vielfältige neuartige Einflussmöglichkeiten auf das Aussehen von Webseiten.

## JavaScript

Wenn HTML die Funktion der Teile eines Dokuments und CSS deren Aussehen beschreibt, dann wird JavaScript benutzt, um festzulegen, wie sich die Elemente verhalten. JavaScript ist eine ausgewachsene Programmiersprache, für deren umfassende Behandlung wieder eigene Bücher erforderlich sind (wie beispielsweise *JavaScript für Dummies*). Hier kann ich JavaScript nicht umfassend behandeln. Es stellt aber eine äußerst kritische Komponente des HTML5-Kon-

zepts dar. Einige der interessantesten HTML5-Funktionen (beispielsweise das `canvas`-Tag, Geolokation und lokale Datenspeicherung) werden erst über JavaScript nutzbar. Die folgenden Funktionen werde ich in diesem Buch beschreiben:

- ✔ **Unterstützung von Vektorgrafiken:** Vektorgrafiken stellen eine interessante Alternative zu den traditionellen Pixelgrafiken dar, weil sie im laufenden Betrieb codegesteuert erzeugt werden können. HTML5 bietet dazu zwei Möglichkeiten an: über SVG (Scalable Vector Graphics) und das `canvas`-Tag.
- ✔ **Neue Selektoren:** Am Anfang der JavaScript-Programmierung steht meist die Auswahl eines Elements über dessen ID-Wert. Mit HTML5 können Sie nun Elemente über deren Tag-Namen oder mit denselben Verfahren wie in CSS auswählen.
- ✔ **Lokale Speichermechanismen:** Die bisherigen HTML-Versionen boten nur sehr beschränkte Möglichkeiten zur clientseitigen Speicherung von Daten. HTML5 bietet Entwicklern nun entsprechende Möglichkeiten. Es steht jetzt sogar ein integrierter Datenbankmanager bereit, der SQL-Befehle versteht.
- ✔ **Geolokation:** Diese interessante Funktion verwendet verschiedene Verfahren, um den jeweils aktuellen geografischen Standort von Nutzern zu ermitteln.

## Servertechnologien

Bei der modernen Webentwicklung geht es um Kommunikation. Alle HTML5 ausmachenden Technologien sind im Webbrowser enthalten, der ein wichtiger Teil des Webs ist. Ein gleichermaßen wichtiger Bestandteil der Webentwicklung ist jedoch die Flut an Technologien, die in den Webserver integriert sind. Viele der heute interessantesten Projekte verwenden Technologien wie PHP oder ASP zur Ausführung von Programmen, die Webseiten erzeugen. Viele interessante Anwendungen nutzen zur Verwaltung größerer Datenmengen auch Datenbankprogramme wie Oracle oder MySQL. Seit AJAX seinen Einzug gehalten hat, ist die Integration derartiger Technologien und des Browsers sehr viel einfacher geworden. So interessant diese Werkzeuge aber auch sein mögen, so stehen sie im vorliegenden Schnelleinstieg doch nicht im Mittelpunkt. Wenn Sie sich dafür interessieren, sollten Sie zu einem Buch mit entsprechendem Themenschwerpunkt greifen.

## Ein Blick auf die Browserfunktionen

Wie sich der Geschichte von HTML entnehmen lässt, reicht es nicht aus, einen Standard zu verkünden, um ihn auch zu einem solchen zu machen. Offiziell wurde HTML5 bisher noch nicht als Standard verabschiedet und es gibt noch keinen einzigen verbreiteten Browser, in dem alle seine Funktionen implementiert wären. Angesichts dessen werden Sie sich vielleicht fragen, ob es dann sinnvoll ist, sich mit dieser Technologie bereits zu befassen. Aus folgenden Gründen denke ich das schon:

- ✔ **Die meisten Ansätze wurden akzeptiert.** HTML5 selbst wurde bisher als Standard zwar formal noch nicht verabschiedet, die meisten kritischen Aspekte wurden aber bereits umgesetzt. Moderne Webbrowser kommen selbst dann mit HTML5 klar, wenn sie nicht alle dort vorgesehenen tollen Möglichkeiten unterstützen.
- ✔ **Es besteht kaum ein Zweifel daran, dass HTML5 der kommende Standard sein wird.** Das W3C hat im Wesentlichen eingestanden, dass XHTML 2.0 keine brauchbare Lösung ist, und HTML5 damit klar zum Sieger im Rennen um den neuen Standard erklärt. Wenn es überhaupt einen Standard geben wird, dann handelt es sich dabei jedenfalls um HTML5 (mit den verwandten Funktionen in CSS und JS).
- ✔ **Die Einhaltung von Standards ist mittlerweile erwünscht.** Im ersten Browserkrieg fügten die konkurrierenden Hersteller ohne Rücksicht auf Standards neue Funktionen hinzu. Heute werden Browser danach beurteilt, wie gut sie sich an anerkannte Webstandards halten. Selbst Microsoft spielt hier mit und behauptet, dass der IE9 die Mehrzahl der HTML5-Funktionen unterstützt.
- ✔ **HTML5 fördert gute Codepraktiken.** Die Trennung des Inhalts vom Layout bildet im Rahmen der modernen Webentwicklung einen kritischen Aspekt. Wenn Sie bisher mit XHTML gearbeitet haben, ist Ihnen diese Situation bereits vertraut. Wenn Sie sich besser mit HTML 4 auskennen, handelt es sich um einen unvermeidlichen neuen Ansatz.

Offiziell steht nicht zu erwarten, dass HTML5 vor 2022 komplett als Standard verabschiedet wird. Im Webzeitalter ist das eine Ewigkeit. Teile des Standards (wie das `canvas`-Tag) werden aber bereits jetzt umfassend unterstützt. Und es lohnt sich, die Möglichkeiten sofort zu untersuchen. Andere Teile (wie die meisten der Formularelemente und der Tags zur semantischen Auszeichnung) bieten geeignete Rückzugslösungen für den Fall, dass der Browser die anspruchsvolle-

ren Funktionen nicht unterstützt. Andere (wie Drag-and-Drop) eignen sich einfach noch nicht für den praktischen Einsatz. Einige (wie die lokale Daten unterstützenden Mechanismen) werden zwar heiß diskutiert, aber noch ist nicht klar, in welcher Form die Technologie in den Standard Einzug halten wird. Bei der Besprechung der jeweiligen Themen in diesem Buch werde ich Sie darüber informieren, ob die Funktionen bereits nutzbar sind und von welchen Browsern sie unterstützt werden.

## Die Browserfähigkeiten ermitteln

Mit HTML5 gehen eine Menge verschiedener Technologien einher und in die verschiedenen Browser wurden bisher unterschiedliche Teile des Standards integriert. Festzustellen, welche Funktionen nutzbar sind, kann recht problematisch werden. Es existieren aber einige gute Lösungen für dieses Problem. Auf einer Reihe von Websites finden Sie Aufstellungen mit Angaben darüber, welche Funktionen von welchem Browser unterstützt werden. Mir persönlich gefallen insbesondere die Zusammenstellungen unter `http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML5)` und unter `http://caniuse.com`. Mit diesen Werkzeugen können Sie einfacher feststellen, welche Funktionen aktuell von den verbreiteten Browsern unterstützt werden. Das ist insbesondere praktisch, wenn es um Browser geht, die Sie nicht auf Ihrem eigenen Rechner installiert haben.

Allerdings ändert sich die Unterstützung der HTML5-Funktionen in den verschiedenen Browsern buchstäblich täglich. Laufend werden neue Versionen der verbreiteten Browser veröffentlicht, weshalb man über den aktuellen Stand kaum wirklich auf dem Laufenden bleiben kann. Deshalb habe ich ein Programm erstellt, mit dem Sie prüfen können, welche HTML5-Funktionen aktuell von Ihrem Browser unterstützt werden. Abbildung 1.1 zeigt `detect.html` in Aktion.

`detect.html` finden Sie unter `www.downloads.fuer-dummies.de`. Nutzen Sie das Programm in einem beliebigen Browser, um die von diesem Browser unterstützten HTML5-Funktionen direkt zu analysieren.

Das Programm nutzt ein Skript namens Modernizr, mit dem sich die Überprüfung verschiedener Browserfunktionen automatisieren lässt. Modernizr erhalten Sie kostenlos unter `www.modernizr.com`.

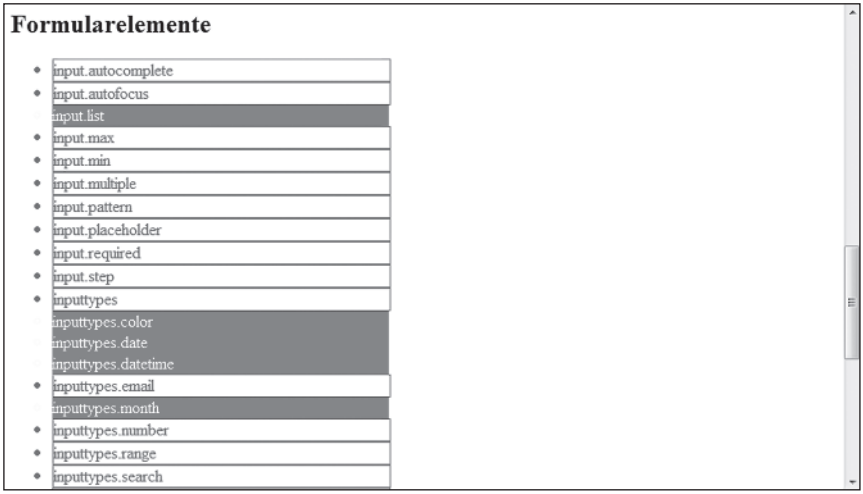


*Abbildung 1.1: `detect.html` in Aktion*

## Verfügbare Funktionen selbst prüfen

Sie können das Modernizr-Skript auch in Ihrem eigenen Code nutzen. Im Wesentlichen erzeugt Modernizr einen booleschen Wert (true/false) für die jeweiligen HTML-Funktionen. Wenn Sie feststellen wollen, ob Ihr aktueller Browser eine bestimmte Funktion unterstützt, können Sie dazu eine entsprechende Variable prüfen. Bei positivem Testverlauf können Sie die Funktion anschließend implementieren. Fällt der Test negativ aus, werden Sie zumeist irgendeine Alternativlösung implementieren. So gehen Sie bei der Prüfung vor:

1. **Laden Sie das Modernizr-Skript aus dem Web herunter.** Es ist kostenlos über `www.modernizr.com` erhältlich. Installieren Sie das Skript im selben Verzeichnis wie Ihre Webseite. (Wenn Sie Ihre Seite auf einen Server verschieben, müssen Sie dafür sorgen, dass auch dort eine Kopie des Skripts vorhanden ist.)

2. **Binden Sie einen Verweis in das Skript ein.** Verwenden Sie das `<script>`-Tag, um im Header (vor irgendwelchem anderen JavaScript-Code) eine Referenz auf das Skript zu erstellen:

```
<script type = "text/javascript"
        src = "modernizr-1.6.min.js"></script>
```

3. **Fügen Sie eine spezielle Klasse zum `html`-Tag hinzu.** Für das Modernizr-Skript muss ein spezielles Tag verfügbar sein, damit es weiß, was es machen soll. Fügen Sie daher die Klasse `"no-js"` zum `html`-Tag hinzu:

   ```
   <html lang = "de-DE"
         class = "no-js">
   ```

4. **Schreiben Sie eine neue JavaScript-Funktion, die die eigentlichen Tests durchführt.** Spezifische Beispiele finden Sie in den Quelltexten weiter hinten in diesem Kapitel.

5. **Verwenden Sie die passende boolesche Eigenschaft, um die Unterstützung einer bestimmten Funktion zu prüfen.** Zu allen von Modernizr unterstützten HTML5-Funktionen existiert eine entsprechende Variable. (Auf der Modernizr-Website können Sie nachsehen, welche Variablen es gibt. Alternativ können Sie dazu aber auch einen Blick in mein Skript `detect.html` werfen, das sie alle verwendet.)

6. **Benutzen Sie die Funktion oder eine Alternative.** Üblicherweise werden Sie Modernizr benutzen, um die Unterstützung einer Funktion zu überprüfen. Wenn diese vorhanden ist, werden Sie sie verwenden. Falls nicht, werden Sie irgendeine Alternativlösung implementieren.

Um Ihnen ein Beispiel zu liefern, verwendet das nachfolgend abgedruckte Skript `videoCheck.html` das Modernizr-Skript, um zu prüfen, ob der aktuelle Browser das HTML5-Tag `video` unterstützt. Ist das der Fall, wird auch noch geprüft, welche der beiden wichtigsten Videocodecs genutzt werden können.

```
<!DOCTYPE HTML>
<html lang = "de"
      class = "no-js">
<head>
  <title>checkVideo.html</title>
  <meta charset = "ISO-8859-1" />
  <meta http-equiv="X-UA-Compatible" content="chrome=1" />
    <script type = "text/javascript"
            src = "modernizr-1.6.min.js"></script>
    <script type = "text/javascript">
      function init(){
        var output = document.getElementById("output");
```

```
        if (Modernizr.video){
          output.innerHTML =
            "Dieser Browser unterstützt das HTML5-Tag
            'video'.<br /> ";
          if (Modernizr.video.h264){
            output.innerHTML += "Der H.264-Codec
            wird unterstützt.<br />";
          } // end if
          if (Modernizr.video.ogg){
            output.innerHTML += "Der Ogg-Theora-Codec wird
            unterstützt.<br />";
          } // end if
        } else {
          output.innerHTML =
            "Dieser Browser unterstützt das HTML5-Tag
            'video' nicht.";
        } // end else
      } // end init
  </script>
</head>

<body onload = "init()">
  <h1>Unterstützung des HTML5-Tags video</h1>
  <div id = "output">
    Prüfe Unterstützung ...
  </div>
</body>
</html>
```

Abbildung 1.2 zeigt das Skript `videoCheck.html` zur Prüfung der Unterstützung des `video`-Tags in Aktion.

Dieses Beispiel prüft einfach, ob das `video`-Element unterstützt wird. In einem anspruchsvolleren Beispiel würde man die zugehörigen Tags oder den Code in eine Webseite integrieren, um den Möglichkeiten entsprechend im Browser ein Video anzuzeigen.

Weitere Informationen zum `video`-Tag finden Sie in Kapitel 3.


*Abbildung 1.2: Prüfung der Unterstützung des `video`-Tags*

## Einen passenden Browser auswählen

Wenn Sie HTML5-Code schreiben wollen, werden Sie Ihre Seiten wahrscheinlich in einem Browser betrachten wollen, der HTML5 korrekt interpretiert. Das ist weniger einfach, als es sich anhört. Bei HTML5 handelt es sich eigentlich nicht um eine Spezifikation, sondern um eine Reihe verschiedener Standards. Die verschiedenen Browser bieten unterschiedlich gute Unterstützung. Es empfiehlt sich, möglichst viele Browser zur Auswahl zu haben, um feststellen zu können, welcher sich für die eigenen Zwecke am besten eignet. Die aktuell verfügbaren Browser bieten unterschiedliche HTML5-Unterstützung.

Beachten Sie dabei, dass zwar eine recht große Anzahl von Browsern zur Auswahl steht, dass die meisten aber auf einer kleineren Auswahl von Werkzeugen basieren, die *Rendering-Engines* genannt werden. Diese geben hinsichtlich der Unterstützung bestimmter Funktionen letztlich den Ausschlag. Hier eine Aufstellung der wichtigsten Engines, Browser, von denen sie verwendet werden, und Angaben zur Qualität der HTML5-Unterstützung:

- ✔ **Gecko (Firefox):** Die Gecko-Engine wird vorrangig von Firefox, Mozilla und einer Reihe weiterer verwandter Browser genutzt. Sie unterstützt zwar viele, aber nicht alle Funktionen. Zwar sollte bereits das in Firefox 4.0 enthaltene Gecko 2.0 die allermeisten HTML5-Funktionen unterstützen, bei einigen ist dies aber auch heute noch nicht (vollständig) der Fall. Und auch wenn Firefox ein in der Gemeinde der Webentwickler beliebter und geschätzter Browser ist, lässt seine HTML5-Unterstützung doch immer noch ein wenig zu wünschen übrig.

Nach dem Versionsübergang von 3.6 auf 4.0 bei Firefox hat sich auch die Politik bei der Benennung der Gecko-Versionen geändert. Die Gecko-Version entspricht seit Version 5 der Firefox-Version. Und seit der im März 2011 veröffentlichten Firefox-Version 4.0 fand somit bis zum April 2012 auch ein Sprung der Gecko-Versionen von 2.0 auf 12.0 statt.

- **Trident (Internet Explorer):** Die verschiedenen Varianten des Internet Explorer verwenden durchweg die Trident-Engine. Diese besaß lange die schwächste Unterstützung von HTML5-Funktionen unter den wichtigen Browsern. Mit IE9 wurde die HTML5-Unterstützung zwar deutlich vollständiger, aber selbst hier fehlen noch einige wichtige Funktionen, wie beispielsweise die Unterstützung anspruchsvollerer Formularelemente und Geolokation.
- **WebKit:** Die WebKit-Engine wurde ursprünglich von Apple entwickelt und basierte auf Code aus dem Open-Source-Projekt KHTML. Apple veröffentlichte den Code dann als Open Source, woraufhin er zur Basis einer Reihe von Browsern wurde. Die Safari-Browser auf Macs, iPhones und iPads verwenden durchweg die WebKit-Engine. WebKit dient auch als Grundlage des Google-Browsers Chrome, des namenlosen und des Dolphin-Browsers auf der mobilen Android-Plattform. WebKit ist zur Standard-Rendering-Engine mobiler Plattformen geworden. Wenn Sie sehen wollen, wie Ihre Seiten auf mobilen Plattformen aussehen, sollten Sie zu diesem Zweck einen auf WebKit basierenden Browser wie Chrome oder Safari verwenden. WebKit bietet die umfassendste Unterstützung von HTML5-Elementen, unterstützt aber auch noch nicht alles. Den meisten Code in diesem Buch habe ich anfangs in Google Chrome 6 mit seiner WebKit-Engine getestet.
- **Presto:** Die Opera-Browserfamilie baut auf der Presto-Engine auf. Opera genießt zwar seit Langem den Ruf eines technisch überlegenen Browsers, konnte aber nie die ihm eigentlich gebührenden Marktanteile für sich gewinnen. Auf Gaming-Plattformen und Mobilgeräten basieren einige Browser auf Presto (beispielsweise Wii Internet Channel, der Nintendo DS Browser und Opera Mobile sowie Opera Mini, die für zahlreiche Mobiltelefone und portable Geräte bereitstehen).

Browserspezifikationen ändern sich recht häufig mit den Versionen. Wenn Sie dieses Buch lesen, wurden wahrscheinlich bereits wieder einige neue Funktionen zusätzlich implementiert. Sie sollten Ihre Webseiten immer in möglichst vielen Browsern testen, um keine bösen Überraschungen zu erleben. Sie sollten in diesem Zusammenhang auch die bereits erwähnte Webseite `http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML5)` nutzen.

Auf dieser Wikipedia-Seite finden Sie meist die aktuellsten Informationen darüber, welche HTML-Funktionen von welchem Browser unterstützt werden.

## Die HTML5-Unterstützung im IE mit Chrome Frame verbessern

Es ist schon ganz schön deprimierend, wenn man feststellen muss, dass der Browser mit dem größten Marktanteil HTML5-Standards am wenigsten unterstützt. Hier gibt es aber Abhilfe. Mit *Chrome Frame* bietet Google eine spezielle Lösung an, bei der die Rendering-Engine von Chrome in den Internet Explorer (IE) eingebunden wird.

Passen Sie bei Angaben zu Browsermarktanteilen auf. Der Internet Explorer ist *weltweit* dominant und wird insbesondere in den USA bevorzugt genutzt, wobei die allermeisten Internetzugriffe weiterhin über Desktoprechner erfolgen. Im Mobilbereich hat Apples Safari-Browser momentan deutlich die Nase vorn, der auch ansonsten neben Google Chrome auf dem Vormarsch ist. Deutschland ist (oder war zumindest einige Jahre lang) aber so etwas wie ein Firefox-Land.

Damit das funktioniert, muss Chrome Frame selbst aber erst einmal installiert werden. Und das hat den Haken, dass Besucher der von Ihnen erstellten Webseiten gegebenenfalls dazu erst einmal aufgefordert werden müssen.

Das lässt sich beispielsweise mit dem folgenden Minibeispiel erledigen, das Sie beim Quellcode zum Buch in der Datei `ChromeFrame.html` finden und das bei installiertem Chrome Frame und in anderen Browsern als dem Internet Explorer rein gar nichts macht:

```
<html lang="de-DE">
 <head>
  <title>ChromeFrame.html</title>
  <meta charset="ISO-8859-1" />
  <script type="text/javascript" src=
   "http://ajax.googleapis.com/ajax/libs/chrome-frame/1/
    CFInstall.min.js">
  </script>
 </head>
 <body onload="CFInstall.check()">
 </body>
</html>
```

Ist Chrome Frame nicht installiert, wird der Nutzer auf eine entsprechende Webseite geführt, die ihm die Möglichkeit zum Herunterladen und zum Installieren von Google Chrome Frame anbietet (siehe Abbildung 1.3).


*Abbildung 1.3: Die Webseite für den Download von Google Chrome Frame*

Nach erfolgter Installation müssen Sie den Internet Explorer beenden und neu starten. Dann wird gefragt, ob das neu installierte Add-on aktiviert werden soll. (Bis es wirklich nutzbar ist, vergeht dann noch einmal eine Weile.)

Damit das installierte Chrome Frame nun auch genutzt wird, müssen Sie in Ihre HTML-Quelltexte jeweils ein zusätzliches `meta`-Tag einfügen, das so aussieht:

```
<meta http-equiv="X-UA-Compatible" content="chrome=1" />
```

Da diese Zeile im HTML-Code keine anderweitigen nachteiligen Wirkungen haben sollte, können Sie sie eigentlich in alle Ihre HTML5-Dateien aufnehmen. Sie finden sie beispielsweise in den Dateien `videoCheck.html` und `detectWCF.html` bei den Quelltexten zum Buch. `detectWCF.html` ist dabei – abgesehen vom zusätzlichen `meta`-Tag – mit `detect.html` identisch.

Abgesehen von diesem `meta`-Tag können Sie Ihren Code anschließend so schreiben, als ob beim Benutzer Chrome (mit seiner hervorragenden HTML5-Unterstützung) installiert wäre. Dies ist jedenfalls die beste Möglichkeit zur Nutzung von HTML5 im IE, bis sich Microsoft dazu durchringt, seinen Browser mit nennenswerter HTML5-Unterstützung auszurüsten.

Weitere Hinweise und umfangreichere Beispiele für die Installation von Google Chrome Frame finden Sie unter `http://www.chromium.org/developers/how-tos/chrome-frame-getting-started`.