

Michael Jendryschik

Einführung in XHTML, CSS und Webdesign

Standardkonforme, moderne und
barrierefreie Websites erstellen



2., aktualisierte und erweiterte Auflage



3 Hilfsmittel

3.1 Editoren

XHTML- und CSS-Dokumente sind keine ausführbaren oder Binärdateien, sondern einfache Textdateien, das heißt, sie bestehen aus für den Menschen lesbarem Text und lassen sich mit jedem einfachen textorientierten Editor (auch: ASCII-Editor) öffnen und editieren. Jedoch greifen viele Webentwickler auf umfangreiche und zum Teil verhältnismäßig teure WYSIWYG-Programmpakete namhafter Hersteller zurück, die die Erstellung und Bearbeitung von Dokumenten unabhängig von deren Quelltext erlauben. Dieser Abschnitt stellt Vor- und Nachteile beider Editorarten gegenüber.

3.1.1 WYSIWYG-Programme

• • • DEFINITION

WYSIWYG ist das Akronym für das Prinzip What You See Is What You Get (Was du siehst, ist das, was du erhältst). Bei echtem WYSIWYG wird ein Dokument während der Bearbeitung am Bildschirm exakt so angezeigt, wie es bei der Ausgabe über ein anderes Gerät aussieht, beispielsweise einem Drucker. Übertragen auf die Webentwicklung ist ein WYSIWYG-Programm ein Werkzeug, dessen Darstellung während der Bearbeitung der XHTML-Datei deren späterer Anzeige im Browser exakt entspricht.



Microsofts Word gehört zu den bekanntesten WYSIWYG-Programmen. Der Benutzer sieht anfangs eine leere weiße Seite vor sich und beginnt, sie zu editieren. Er kann Texte formatieren, Bilder einfügen und Tabellen anlegen. Die gewünschten Inhalte lassen sich über Menübefehle und Dialoge einfügen. Die Ergebnisse seiner Aktionen kann er direkt auf dem Bildschirm überprüfen.

Nach demselben Prinzip soll ein WYSIWYG-Programmpaket zur Erstellung von Webseiten funktionieren. Das Programm setzt hierbei die vom Benutzer ausgeführten Aktionen im Hintergrund in Quelltext um. Wenn er es nicht möchte, braucht der Benutzer den Raum der grafischen Darstellung also gar nicht zu verlassen. Viele WYSIWYG-Editoren bieten auch die Möglichkeit, zwischen Layout- und Quelltextansicht umzuschalten, sodass Sie die Entwicklung der Seite auch direkt im Quelltext verfolgen und dort Korrekturen und Änderungen vornehmen können.

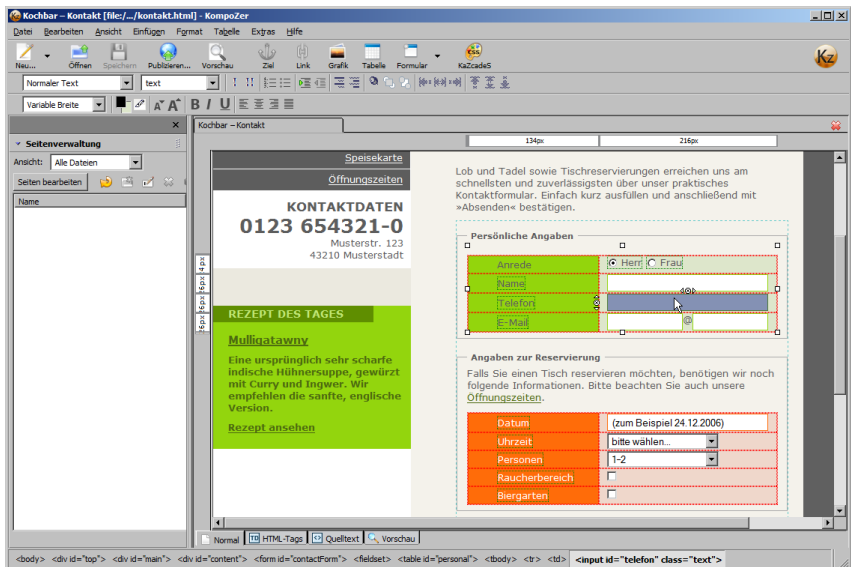


Abbildung 3.1: Der WYSIWYG-Editor KompoZer

Vor- und Nachteile

Die Vorteile eines WYSIWYG-Programms liegen in der Natur der Sache:

- Vorteile
- ▶ Sie sind sehr schnell in der Lage, eine Seite fertigzustellen oder ein bestehendes Grundgerüst mit Inhalten zu füllen.
 - ▶ Das Transkribieren von der abstrakten Quelltextebene hin zu der Seitendarstellung im Browser fällt weg.

- ▶ WYSIWYG-Programme verlangen keine oder kaum XHTML-Kenntnisse zur Fertigstellung einer Seite und verfügen zusätzlich oft über Layoutvorlagen, die Ihnen die Konzeption eines Screendesigns weitgehend abnehmen.
- ▶ Viele WYSIWYG-Programme verfügen über ein ausgereiftes Projektmanagement, das Ihnen die konzeptionelle Arbeit zum Teil abnimmt.

WYSIWYG-Programmpakete scheinen ein guter Einstieg in die Webentwicklung zu sein, da Sie mit minimalen XHTML- und CSS-Kenntnissen rasche Ergebnisse erhalten und schnell ein Gefühl dafür entwickeln, welche Möglichkeiten klassisches Webauthoring bietet und welche nicht. Leider führt diese Arbeitserleichterung insgesamt zu einer Simplifizierung des Erstellens von Webseiten. Dadurch dass das Programm zum einen die Gedankenarbeit sowohl beim Projektmanagement als auch beim Erstellen des Seitenkonzepts und zum anderen auch die Erstellung des Quelltexts abnimmt, bekommt der Anwender schnell das Gefühl, die Erstellung von Webseiten sei eine sehr einfache Sache. Er hat damit nur dann recht, wenn er sich mit den Ergebnissen des WYSIWYG-Programms zufriedengibt. Dies sollten Sie jedoch nicht, denn leider haben derartige Programme auch sehr viele Nachteile:

- ▶ WYSIWYG-Programme erzeugen in einigen Fällen unerklärlicherweise viel überflüssigen Code, was längere Ladezeiten im Browser und eine große Unübersichtlichkeit des Quelltexts bedeutet. Nachteile
- ▶ Oftmals entspricht der erzeugte Quelltext nicht den W3C-Empfehlungen; Sie erzeugen folglich ungültige Dokumente. Dies kann dazu führen, dass kein Browser die Webseite tatsächlich so darstellt, wie es die Vorschau Ihres WYSIWYG-Programmpakets, auf dem Sie sie entworfen haben, zeigt. Dies bedeutet unter Umständen kleinen Abweichungen, aber auch das völlige Fehlen wichtiger Seiteninhalte (siehe auch Kapitel 3.2 »Validatoren«).
- ▶ Ein nicht unwesentlicher Nachteil ist die häufig mangelhafte Trennung von Markup und Layout und das inkonsequente Einsetzen von CSS, was Sie als Webautor per Hand wesentlich effizienter gestalten können.

WYSIWYG-Programmpakete werden zu Unrecht so genannt. Das WYSIWYG-Prinzip stimmt immer nur für den Autor einer Webseite selbst und für einen gewissen Nutzerkreis, der über die gleichen Voraussetzungen und die gleiche Umgebung wie der Autor verfügt. Im World Wide Web müssen Sie davon ausgehen, dass die Besucher die erstellten Seiten mit anderen Browsern auf anderen Betriebssystemen über andere Auflösungen betrachten. Dadurch können die Seiten bei den Besuchern anders aussehen als beim Autor. WYSIWYG-Programme sind somit vielmehr Das Web kennt kein WYSIWYG

- ▶ WYSIWYG- (*What You See Is What You Might Get*, Was du siehst, ist das, was du vielleicht erhältst),
- ▶ WYSIMOLWYG- (*What You See Is More Or Less What You Get*, Was du siehst, ist mehr oder weniger das, was du erhältst),
- ▶ WYTYSIWYTYG- (*What You Think You See Is What You Think You Get*, Was du zu sehen glaubst, ist das, was du zu erhalten glaubst) oder
- ▶ WYSIWYM-Programme (*What You See Is What You Mean*, Was du siehst, ist, was du meinst) – vielleicht die treffendste Bezeichnung.

Aus diesem Grund verwendet kaum ein Hersteller eines Programms, das man in die Reihe der WYSIWYG-Programme einordnet, diesen Begriff selbst; da scheinen sich eher Bezeichnungen durchzusetzen wie »Werkzeug für Web-Authoring« (*Macromedia Dreamweaver*) oder »Tool für die Anwendungsentwicklung und die Webseitenerstellung« (*Microsoft FrontPage*).

3.1.2 Texteditoren

• • • DEFINITION

Ein textorientierter Editor ist ein Programm zum Bearbeiten von Texten, beispielsweise von XHTML-Dokumenten auf Quelltextebene. Der Editor lädt das zu bearbeitende Textdokument und zeigt seinen Inhalt auf dem Bildschirm an. Durch diverse Aktionen können die Daten dann bearbeitet werden, beispielsweise durch Einfügen, Löschen oder Kopieren.

Vor- und Nachteile

Die Nachteile von Texteditoren sind – gerade für Anfänger – offensichtlich:

- Nachteile
- ▶ Sie benötigen XHTML- und CSS-Kenntnisse, um eine Seite zu erstellen.
 - ▶ Das Tippen des Quelltexts dauert länger als das Zusammenklicken eines Dokuments in einem WYSIWYG-Programmpaket, selbst wenn Sie ausgiebig mit Includes (Mehrfachverwendung von in der Regel zentral abgelegtem Quelltext) oder Kopieren und Einfügen arbeiten.
 - ▶ Texteditoren verfügen in den meisten Fällen über kein umfangreiches Projekt- und Dokumentenmanagement.
 - ▶ Sie sind auf sich allein gestellt, wenn es um die Erarbeitung einer Informationsarchitektur geht.
 - ▶ Das Umsetzen eines Entwurfskonzepts in einen gut strukturierten XHTML-Quelltext erfordert große Abstraktionsfähigkeiten.

Die meisten Vorteile von Texteditoren offenbaren sich erst nach kurzer Eingewöhnungszeit:

- ▶ Sie haben direkte Kontrolle über die Entstehung des Seitenquelltexts. Vorteile
- ▶ Sie können die Auswirkungen jeder noch so geringen Änderung am Quelltext sofort im Browser überprüfen.
- ▶ Sie beschränken sich beim Schreiben Ihrer Seite auf das Wesentliche und schreiben dadurch keinen überflüssigen Code.
- ▶ Sie ordnen Ihren Quelltext für Sie verständlich und logisch.
- ▶ Sie haben enorme Lernfortschritte mit jeder Seite, die Sie schreiben. Sie gewinnen immer mehr Verständnis für die Zusammenhänge und sehen die Textauszeichnung im Quelltext getrennt von der Darstellung im Browser. Dadurch entwickeln Sie ein besseres Verständnis für stilistisch gutes Webauthoring.
- ▶ Ein Wechsel des Editors ist leichter als bei WYSIWYG-Programmen, da der eigene Quellcode mit einem anderen Editor nach dem gleichen Prinzip weiterverarbeitet werden kann.
- ▶ Es liegt in Ihrer Hand, nicht in der Hand eines Programms, ob Sie beim Erstellen der Seite Fehler machen oder nicht.

Erfahrene Webautoren bevorzugen in der Regel reine Texteditoren. Wie bereits erwähnt, reichen einfachste Texteditoren wie Notepad unter Windows vollkommen aus. Björn Höhrmann schrieb vor einigen Jahren überspitzt, aber nicht ganz zu Unrecht:

»Notepad ist kein HTML-Editor, Notepad ist das Schweizer Armeemesser mit integriertem Whirlpool für jeden, der nur ein bisschen auf sich hält und effektiv arbeiten will.«

Dennoch sollten Sie bei der Wahl Ihres Editors auf einige nützliche Funktionen achten, die Ihnen die Arbeit an Ihrer Website ein wenig erleichtern:

- ▶ **Syntaxhervorhebung** (*syntax highlighting*) von mindestens XHTML und CSS. Dadurch werden einzelne syntaktische Bestandteile eines Dokuments und eigentliche Inhalte unterschiedlich eingefärbt, wodurch die Orientierung im Quelltext erleichtert wird und Fehler schneller sichtbar werden. Syntaxhervorhebung für weitere relevante Sprachen wie JavaScript, PHP, Perl und andere ist wünschenswert und in den meisten Editoren auch vorhanden oder integrierbar. Kriterien für gute Texteditoren

```

index.html - Editor
Datei Bearbeiten Format Ansicht ?
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="de" lang="de">
  <head>
    <meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
    <title>Kochbar</title>
    <meta name="author-personal"
content="Michael Jendryschik" />
    <meta name="author-mail"
content="mailto:michael@jendryschik.de" />
    <meta name="description" content="Vorstellung der Kochbar
in Musterstadt mit Informationen zur Küche, dem
Biergarten sowie dem Rezept des Tages." />
    <link rel="stylesheet" href="default.css"
type="text/css" media="screen,projection" />
  </head>
  <body>
    <div id="top">
      <div id="head">
        <!-- globale Navigation -->
        <ul>
          <li><strong title="Hier befinden Sie sich
gerade.">Startseite</strong></li>
          <li><a href="kontakt.html">kontakt</a></li>
          <li class="skiplink"><a href="#content">
Direkt zum Inhalt</a></li>
        </ul>

```

Abbildung 3.2: Minimalausstattung: Notepad

- ▶ **Dateiübergreifendes Suchen und Ersetzen.** Müssen Änderungen in einem Dokument oder in mehreren vorgenommen werden, führt eine einfache Suchen-und-Ersetzen-Funktion zu schnellen Ergebnissen. Die Verwendung von **regulären Ausdrücken** (*regular expressions*) wird der Profi ebenfalls zu schätzen wissen.
- ▶ **UTF-8-Fähigkeit** und ganz allgemein die Unterstützung und Konvertierung von möglichst vielen Kodierungen und Zeichensätzen, wie den Familien ISO-8859-n, Windows 125 oder MacRoman, um alle nötigen Zeichen eingeben, anzeigen und abspeichern zu können (siehe Kapitel 2.10 »Zeichenkodierung«).
- ▶ Zusätzlich eingebaute oder nachrüstbare **Werkzeuge** wie Validatoren, Rechtschreibprüfung, Quelltextvervollständigung, Makrofähigkeit, Verwendung von Textbausteinen, automatische Einrückung von Quelltextzeilen und Includes können die Arbeit ebenso vereinfachen wie benutzerdefinierte Werkzeuge leisten mit häufig verwendeten Kommandos oder eigene Tastenkombinationen.

An dieser Stelle soll nicht verschwiegen werden, dass die Welt der Editoren nicht nur schwarz-weiß ist. Die meisten WYSIWYM-Programme ermöglichen beide Arbeitsformen, und es gibt Entwickler, die eine zweigleisige Arbeitsweise als die effektivste empfinden. Die WYSIWYM-Oberfläche dient dabei nur dem bequemen Editieren und Auszeichnen mit Standardelementen, während der Quelltext jederzeit parallel sichtbar ist und die Gestaltung vollständig

über CSS vorgenommen wird – häufig assistiert durch das jeweilige Programm. Abbildung 3.3 zeigt ein Beispiel für diese Arbeitsweise.

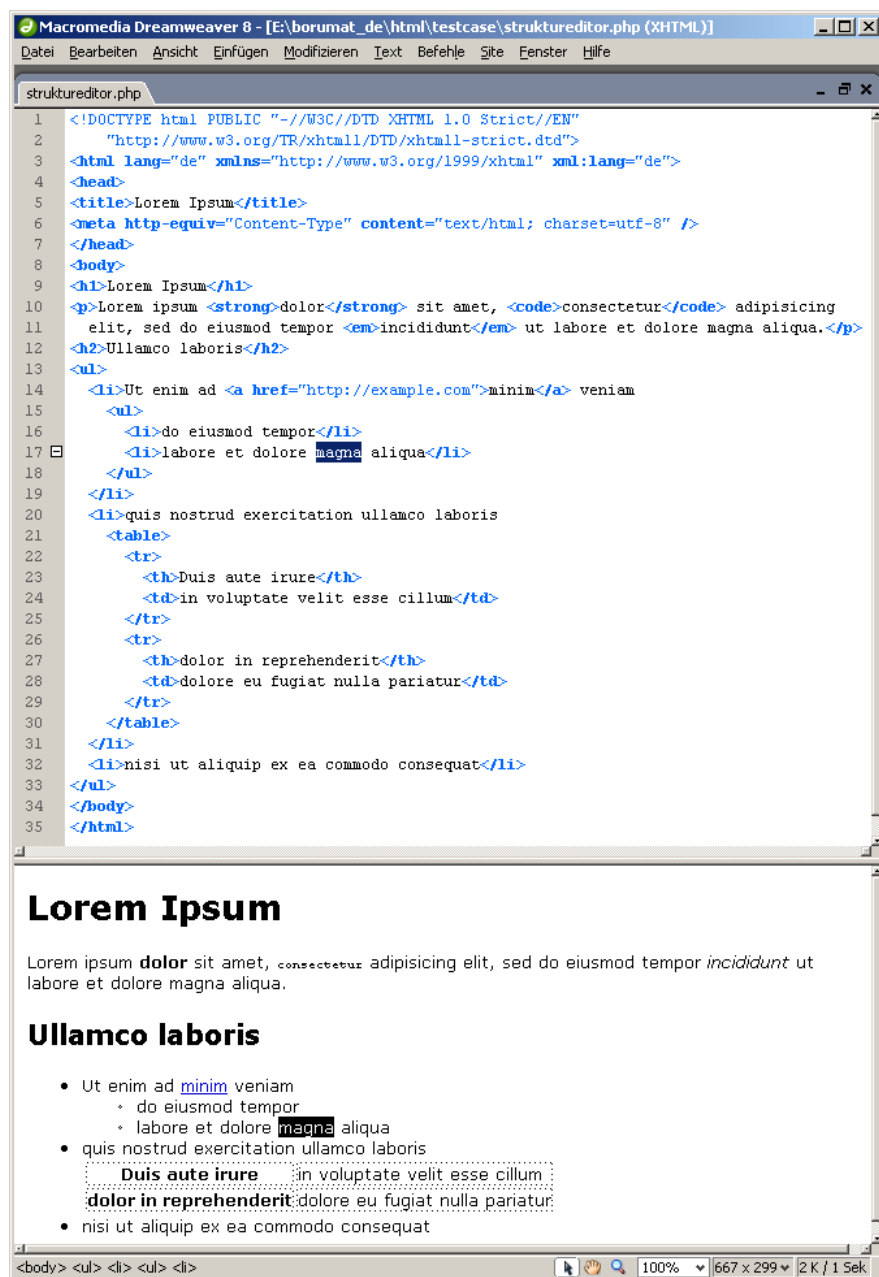


Abbildung 3.3: Kombinierte Quelltext- und WYSIWYG-Ansicht in Macromedia Dreamweaver 8 (Screenshot von Andreas Borutta)

3.1.3 Fazit

Je nach verwendetem Betriebssystem, Geschmack und der erforderlichen Einarbeitungszeit mag die Wahl auf den einen oder anderen Editor fallen. Manche bevorzugen universelle und extrem mächtige Editoren wie *vim* oder *UltraEdit*, deren Bedienung allerdings längere Zeit erlernt werden will, die dann aber keine Wünsche offen lassen und zum Teil auf zahlreichen Betriebssystemen verfügbar sind. Andere bevorzugen einfach zu bedienende Editoren wie *Notepad++* oder *BBEdit*. Auf Websprachen spezialisierte Editoren sind *Phase 5*, *Homesite* und *Bluefish*. Programmierer werden XHTML- und CSS-Dokumente mit dem Quelltexteditor ihrer Software-Entwicklungsumgebung bearbeiten, beispielsweise *Eclipse*. Tabelle 3.1 und Tabelle 3.2 listen einige empfehlenswerte Editoren auf. Ausführliche Auflistungen finden Sie darüber hinaus bei Dr. Web: <http://www.drweb.de/editoren/wysiwyg-editoren-markt.shtml> listet nützliche WYSIWYG-Programme auf, <http://www.drweb.de/editoren/html-editoren-markt.shtml> eine Reihe von Texteditoren.

Editor	OS	Kostenlos?	URL
Arachnophilia	Win	ja	http://www.arachnoid.com/arachnophilia/
BBEdit	Mac	nein	http://www.barebones.com/products/bbedit/
BlueFish	Unix	ja	http://bluefish.openoffice.nl/
Eclipse	Win/Mac/ Unix	ja	http://www.eclipse.org/
HomeSite	Win	nein	http://www.adobe.com/products/homesite/
Notepad++	Win	ja	http://notepad-plus.sourceforge.net/de/site.htm
Notepad2	Win	ja	http://www.flos-freeware.ch/notepad2.html
PageSpinner	Mac	nein	http://www.optima-system.com/pagespinner/
Phase 5	Win	eingeschränkt	http://www.phase5.info/
UltraEdit	Win	nein	http://www.ultraedit.com/
vim	Win/Mac/ Unix	ja	http://www.vim.org/

Tabelle 3.1: Empfehlenswerte Texteditoren

Editor	OS	Kosten- los?	URL
Dreamweaver	Win/Mac	nein	http://www.adobe.com/products/dreamweaver/
Namo WebEditor	Win	nein	http://www.namo.com/products/webeditor.php
NetObjects Fusion	Win	nein	http://www.netobjects.de/
KompoZer	Win/Mac/ Unix/OS2	ja	http://www.kompozer.net/

Tabelle 3.2: Empfehlenswerte WYSIWYM-Editoren

Es gibt Hunderte guter Editoren für die unterschiedlichsten Ansprüche. Schauen Sie sich in Ruhe einige an und probieren Sie sie längere Zeit aus. Ich selbst arbeite seit mehreren Jahren mit dem Texteditor *UltraEdit*. Auf diesen oder einen anderen Texteditor sollten Sie vor allem dann zurückgreifen, wenn Sie diese Einführung durcharbeiten.

3.2 Validatoren

! ! ! ACHTUNG

Um diesen Abschnitt umfassend verstehen zu können, benötigen Sie Kenntnisse über die Syntax und das Vokabular von XHTML und sollten wissen, wie HTML und XHTML aus SGML und XML entstanden sind. Zum Teil greife ich Inhalten weiterer Teile dieser Einführung vor, allerdings denke ich, dass ein Abschnitt über Validatoren in diesem Kapitel am besten aufgehoben ist – schließlich gehören Validatoren zu den wichtigsten Hilfsmitteln eines Webautors überhaupt. Anfänger sollten diesen Abschnitt zunächst überspringen und ihn sich für einen späteren Zeitpunkt vormerken.

3.2.1 Natürliche und formale Sprachen

Damit sich zwei Menschen miteinander verständigen können, benötigen sie eine gemeinsame (natürliche) Sprache. Zum einen müssen sich beide ein gemeinsames Vokabular teilen, also über eine ausreichend große Menge an Wörtern verfügen und sich über deren Bedeutung und Funktion einig sein. Zum anderen müssen sie zumindest in groben Zügen dieselben Regeln dazu befolgen, wie aus einzelnen Wörtern Bedeutungsgruppen und daraus wiederum Sätze geformt werden. Im Idealfall lassen sich dann Sätze formulieren, deren Bedeutung eindeutig und beiden Beteiligten klar ist.

So einfach ist dies jedoch nicht. Jeder weiß aus der alltäglichen Kommunikation mit Menschen in seiner Umgebung, dass sich die Bedeutung von Ausdrücken oft nur aus dem Zusammenhang ergibt. Wesentliche Bestandteile einer

Mehrdeutigkeiten natürlicher Sprachen

natürlichen Sprache sind nicht nur Vokabular und Grammatik, sondern auch Melodie und Rhythmus, Mimik und Gestik, Ironie und Sarkasmus, Emotion und Beziehungsdefinitionen. Hinzu kommt, dass viele Äußerungen in natürlicher Sprache nicht nur in ihrer Bedeutung (Semantik), sondern auch in ihrer grammatikalischen Struktur (Syntax) mehrdeutig sind. Der Ausdruck »die Auswahl des Mannes mit dem Zeigestock« veranschaulicht dieses Problem. Der Mann könnte ausgewählt worden sein oder selbst eine Auswahl getroffen haben; es kann sowohl »die Auswahl mit dem Zeigestock« gemeint sein als auch »der Mann mit dem Zeigestock«.

Natürliche Sprachen sind nicht vollständig durch ein festgelegtes Regelwerk, sondern durch ihren Gebrauch definiert und unterliegen ständigem Wandel. Es kann daher nicht immer zweifelsfrei bestimmt werden, ob eine Äußerung oder ein Satz einer bestimmten natürlichen Sprache zuzuordnen und in diesem Zusammenhang »richtig« oder »falsch« ist. Daher sind natürliche Sprachen für die Kommunikation zwischen Menschen und Computern oder Computerprogrammen untereinander nicht geeignet. Anders als wir Menschen kann ein Computer Mehrdeutigkeiten nicht auflösen und ist auch nicht in der Lage, zwischen den Zeilen zu lesen und die Bedeutung eines Ausdrucks aus dem Kontext heraus zu interpretieren. Besser eignen sich formale Sprachen, deren Semantik und syntaktische Regeln anders als bei natürlichen Sprachen eindeutig sind.

Syntax und Grammatik

HTML als SGML-Dokumenttyp und XHTML als XML-Dokumenttyp sind formale Sprachen. Die Grammatik wird durch die syntaktischen Regeln gebildet, die für SGML beziehungsweise XML definiert sind, sowie durch die Elementstrukturen, die die jeweilige **Dokumenttyp-Definition** (DTD) oder das Schema vorgibt – also welche Elemente auf welche Weise ineinander verschachtelt werden dürfen, welche Attribute sie haben dürfen oder müssen und welche Werte diese annehmen können. Das folgende Beispiel ist aus zwei Gründen falsch: Es verstößt gegen die Syntax von XHTML, weil ein Element `p` keine anderen Blockelemente enthalten darf, und hält sich nicht an das vorgegebene Vokabular, da es einen Elementtyp `et` verwendet, der nicht in XHTML enthalten ist:

```
<p>Das folgende Beispiel ist aus zwei Gründen falsch: Es
<ul>
  <li>verstößt gegen die Syntax von XHTML, weil ein
    Element <et>p</et> keine anderen Blockelemente
    enthalten darf, und</li>
  <li>hält sich nicht an das vorgegebene Vokabular, da es
    einen Elementtyp <et>et</et> verwendet, der nicht in
    XHTML enthalten ist.</li>
</ul>
</p>
```

Ein Computer, genauer gesagt ein Validator, kann HTML- und XHTML-Dokumente mithilfe eines Parsers überprüfen und Fehler so eindeutig erkennen.

3.2.2 Syntaktische Fehler und proprietäre Erweiterungen

Viele sogenannte WYSIWYG-Editoren produzieren noch heute teilweise unsaubereres oder gar fehlerhaftes (X)HTML. Aber auch Webautoren, die ihre Dokumente per Hand schreiben, sind vor Fehlern nicht gefeit; wie schnell vergisst man, ein schließendes Tag zu setzen, oder fügt einen URL ein und übersieht ein nicht maskiertes Ampersand (&). Derartige Fehler fallen bei mit dem MIME-Typ `text/html` ausgelieferten Dokumenten nicht auf, zumindest nicht auf den ersten Blick; es sei denn, man bemerkt fehlende oder falsch angezeigte Inhalte in der Darstellung. Dennoch bringen sie Probleme mit sich.

Sind Dokumente fehlerhaft geschrieben, muss der Browser die falschen Stellen interpretieren und sich für eine mögliche Darstellung entscheiden. Das bläht Webseiten auf und macht ihre Verarbeitung langsam. Darüber hinaus ist Fehlerbehandlung nicht genormt. Wer syntaktisch falsches (X)HTML schreibt, kann sich nicht darauf verlassen, dass aktuelle und zukünftige Browser die Dokumente richtig verarbeiten können und morgen noch genau so darstellen wie heute. Werden XHTML-Dokumente mit dem empfohlenen MIME-Typ `application/xhtml+xml` oder einem anderen XML-MIME-Typ ausgeliefert, zumindest an Browser wie Mozilla oder Firefox, die damit etwas anfangen können, wird die Verarbeitung des Dokuments sofort abgebrochen und eine Fehlermeldung angezeigt (siehe Abbildung 3.4) – die Informationen, die dargestellt werden sollten, sind somit nicht mehr zugänglich. Zudem erhalten potenzielle Besucher oder Kunden natürlich auch einen denkbar schlechten Eindruck von der fachlichen Qualifikation des jeweiligen Webautors.

Fehler-
behandlung

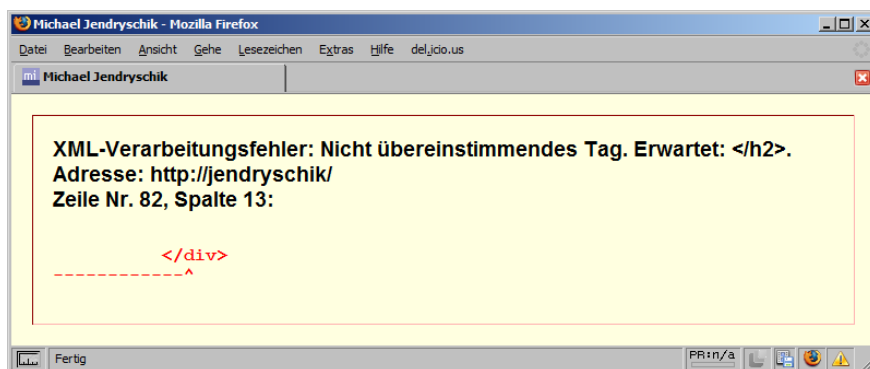


Abbildung 3.4: Firefox zeigt Verstöße gegen die Wohlgeformtheit eines XHTML-Dokuments gnadenlos an

Bei Verstößen gegen das Vokabular der verwendeten Sprachversion handelt es sich oft um Elemente oder Attribute, die nur in der Transitional-DTD enthalten

Erweiterungen

sind, aber in einem strikten Dokument verwendet werden, oder um proprietäre Erweiterungen, die in keiner (X)HTML-DTD enthalten, also nicht genormt sind. Letzteres sind in den meisten Fällen Erweiterungen einzelner Browserentwickler, die oft auch nur in Browsern dieser Hersteller interpretiert werden. Vom Gebrauch solcher Erweiterungen ist abzuraten, da es sehr wahrscheinlich ist, dass sie in späteren Generationen des Browsers, in anderen Browsern oder auf anderen Plattformen nicht dargestellt werden oder zumindest anders als gewünscht, was viel schlimmer ist. Darüber hinaus können Werkzeuge wie Tidy (siehe Kapitel 3.4.1) Dokumente, die unbekannte Erweiterungen enthalten, nicht bearbeiten, da sie nicht wissen, wie sie mit diesen Erweiterungen umzugehen haben. Beispielsweise bleibt bei unbekanntenen Elementen innerhalb von XHTML-Dokumenten die Frage offen, ob das **End-Tag** erforderlich (wie bei `a`), optional (`p`) oder verboten (`img`) ist.

3.2.3 Gültigkeit und Wohlgeformtheit

• • • DEFINITION

Ein Dokument ist **gültig** (*valide*), wenn es erfolgreich gegen eine zugrunde liegende Grammatik (DTD, Schema etc.) geprüft werden kann. Es dürfen also nur die Elemente, Attribute und Attributwerte auf die Art und Weise verwendet (und verschachtelt) werden, wie es für die gewählte Sprachversion definiert ist. Nur ein vollständig fehlerbereinigtes und korrekt geschriebenes Dokument verdient das Attribut **gültig**.

• • • DEFINITION

In XML wird das Konzept der **Wohlgeformtheit** eingeführt. Demnach ist ein XML-Dokument genau dann wohlgeformt, wenn es nach den in Kapitel 2.1 der XML 1.0-Empfehlung definierten Regeln strukturiert ist. Kurz zusammengefasst besagen diese Regeln:

- ▶ Es existiert genau ein Wurzelement. In XHTML-Dokumenten ist es das Element `html`.
- ▶ Alle Elemente werden durch ein Start- und ein End-Tag begrenzt oder bestehen aus einem Leeres-Element-Tag. Anders als bei HTML dürfen Webautoren bei XHTML-Dokumenten beispielsweise das schließende `</p>` nicht mehr weglassen.
- ▶ Elemente sind korrekt ineinander verschachtelt. `barmag gültiges HTML sein; XHTML ist dies nicht. In XHTML müssen Sie bar oder bar schreiben: Die Tags dürfen sich nicht überkreuzen. Welche Variante Sie wählen, ist übrigens abhängig davon, ob Sie Links betonen oder Betonungen verlinken möchten. Sind Sie empfänglich für solche feinen Unterschiede?`
- ▶ Attributwerte müssen immer in Anführungszeichen gesetzt werden.
- ▶ Es gibt keine Attributminimierung. `<input type="checkbox" checked />` ist somit nicht wohlgeformt, `<input type="checkbox" checked="checked" />` hingegen schon.

Die XML-Spezifikation definiert den Begriff »gültig« nur für XML-Dokumente, und genau genommen ist ein XML-Dokument nur dann ein solches, wenn es wohlgeformt ist. XHTML-Dokumente sind XML-Dokumente bestimmter Ausprägung, folglich ist Wohlgeformtheit ein notwendiges Kriterium, um XHTML-Dokumente auf Gültigkeit gegen eine bestimmte Dokumenttyp-Definition prüfen zu können. Diese Voraussetzung gibt es für HTML als SGML-Dokumenttyp nicht, denn SGML kennt das Konzept der Wohlgeformtheit nicht. Jedoch wird seit Aufkommen von X(HT)ML auch ein HTML-Dokument hin und wieder informell als »quasi-wohlgeformt« bezeichnet, wenn es den oben angegebenen, sofern in HTML möglichen, Regeln genügt.

Gültigkeit als Voraussetzung für Wohlgeformtheit

3.2.4 W3C Markup Validation Service

Um zu überprüfen, ob ein Dokument gültig ist, verwenden Webautoren einen für diesen Zweck programmierten Validator. Davon gibt es mittlerweile eine ganze Reihe auf dem Markt und frei für jeden verfügbar im Web. Der *W3C Markup Validation Service*¹ ist der bekannteste. Er beschreibt sich selbst als freien Service, über den Webautoren Dokumente wie HTML- und XHTML-Dokumente bezüglich ihrer Konformität zu den W3C-Empfehlungen und anderen Standards überprüfen können. Aber auch der Validator der *Web Design Group*² ist sehr populär. Nebenbei bemerkt, bietet dieser über die Funktion *Validate entire site* die Möglichkeit, bis zu 100 Dokumente einer Website in einem Schritt zu überprüfen.

Typische Fehlermeldungen und ihre Bedeutung

Es gibt nicht viel, was Sie bei der Erstellung von (X)HTML-Dokumenten falsch machen können. Natürlich ist die Arbeit eines Webautors fehleranfällig, aber es handelt sich fast immer um dieselben Arten von Fehlern. Deswegen werden Sie, wenn Sie eine Weile mit dem W3C-Validator gearbeitet haben, das Gefühl haben, stets die gleichen Fehlermeldungen und Warnungen zu lesen. Zu den am häufigsten vorkommenden Fehlern gehören die folgenden:

► *No DOCTYPE Declaration Found!*

Das Dokument enthält keine **Dokumenttyp-Deklaration** (siehe Kapitel 4.2.2), oder sie ist nicht korrekt.

¹ <http://validator.w3.org/>

² <http://www.htmlhelp.com/tools/validator/>

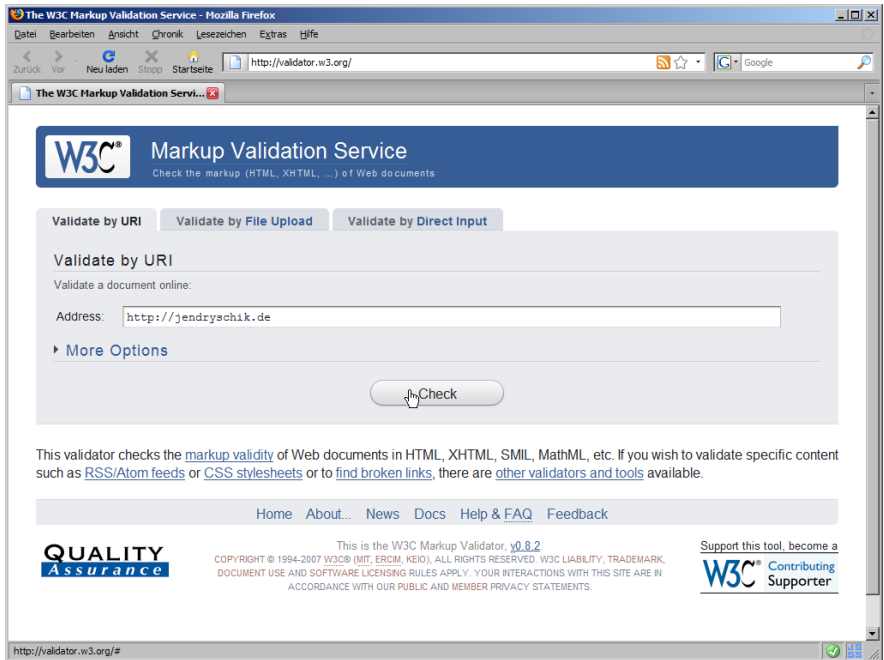


Abbildung 3.5: Der W3C Markup Validation Service

► *No Character Encoding Found!*

Das Dokument wird ohne Informationen zur Kodierung ausgeliefert. Überprüfen Sie Ihre Webservereinstellungen. Falls Sie lokal testen oder ein Dokument hochgeladen haben, müssen Sie die Kodierungsinformation per meta-Element angeben (siehe Kapitel 4.5).

► *Document type does not allow element "p" here; assuming missing "li" start-tag.*

Ein Element `p` steht im Quelltext an einer Stelle, an die es nicht gehört. Der Validator weist Sie darauf hin, dass offenbar ein Element `` fehlt.

► *End tag for element "p" which is not open.*

Im Quelltext steht ein `</p>`, für das es kein öffnendes `<p>` gibt. Vielleicht ist der Grund dafür in einer fehlerhaften Elementverschachtelung zu suchen (also beispielsweise `<p>Es ist ein Fehler aufgetreten!</p>`).

► *End tag for "div" omitted, but OMITTAG NO was specified.*

Im Quelltext steht ein `<div>`, für das es kein schließendes `</div>` gibt. Auch hier könnten Sie Elemente falsch verschachtelt haben, meistens jedoch haben Sie das schließende Tag schlichtweg vergessen.

- ▶ *Element "u" undefined.*

Sie verwenden ein Element (hier `u`), das in der verwendeten (X)HTML-Version nicht enthalten und somit nicht erlaubt ist.

- ▶ *There is no attribute "name".*

Sie verwenden ein Attribut (hier `name`), das in der verwendeten (X)HTML-Version an dieser Stelle nicht erlaubt ist.

- ▶ *Required attribute "alt" not specified.*

Sie haben ein erforderliches Attribut nicht angegeben (hier das `alt`-Attribut des `img`-Elements).

- ▶ *Character "<" is the first character of a delimiter but occurred as data.*

Eine spitze Klammer steht an einer Stelle, an der sie nicht stehen darf. Möglicherweise steht sie irgendwo einsam und allein (beispielsweise `4 < 7`), dann können Sie sie ziemlich einfach finden. Häufiger jedoch ist die Meldung eher irreführend. Der Validator zeigt denselben Fehler an, wenn Sie das abschließende Anführungszeichen eines Attributs vergessen (zum Beispiel `<p class="error">Es ist ein Fehler aufgetreten.</p>`).

- ▶ *Unclosed start-tag requires SHORTTAG YES.*

Sie haben ein **Start-Tag** nicht ordnungsgemäß geschlossen (beispielsweise `<div` anstatt `<div>`).

- ▶ *Character "&" is the first character of a delimiter but occurred as data.*

Das Dokument enthält ein allein stehendes Ampersand (&). Weshalb das falsch ist und wie Sie das Zeichen korrekt notieren, lesen Sie in Abschnitt 4.1.4.

- ▶ *Cannot generate system identifier for general entity "ap".*

Sie haben eine benannte Zeichenreferenz (hier `≈`) notiert, die es nicht gibt.

- ▶ *XML Parsing Error: EntityRef: expecting ';'.*

Sie haben eine Zeichenreferenz ohne abschließendes Semikolon notiert (beispielsweise `&160` anstatt `&160;`).

* * * TIPP

Versuchen Sie niemals, die komplette Fehlerliste auf einmal abzuarbeiten, es sei denn, sie besteht nur aus einer Handvoll Meldungen, die Sie unmittelbar zuordnen können. Der W3C-Validator nimmt es sehr genau und gibt für die gleiche Fehlerursache häufig unterschiedliche Meldungen aus. Zudem können Fehler einander bedingen. Beginnen Sie folglich stets am Anfang der Liste, beheben Sie den ersten oder die ersten beiden Fehler und überprüfen Sie Ihr Dokument anschließend erneut. Gehen Sie so Schritt für Schritt vor, bis Ihr Dokument fehlerfrei ist.

Einschränkungen

Ein Validator ist ein gutes Werkzeug, das Webautoren effizient dabei unterstützt, fehlerfreie Dokumente zu erzeugen. Er ist aber lediglich ein Werkzeug. Es ist nicht gesagt, dass Dokumente, die ein Validator als gültig klassifiziert, auch wirklich fehlerfrei im Sinne der Spezifikation sind. Eine DTD kann nicht alle formalen Einschränkungen beschreiben, denen ein Dokumenttyp unterliegt. So lassen sich keine Datentypen wie Integer oder Float definieren; Daten sind praktisch nur Zeichenketten. Das Element

```
<a href="ich bin kein uri">Anker</a>
```

beispielsweise ist unsinnig im Sinne der Empfehlung, die als Wert des href-Attributs einen URI fordert, aber es ist gültig im Sinne der DTD, die für das href-Attribut den Inhaltstyp CDATA definiert, also lediglich eine Folge von Zeichen. Darüber hinaus soll nicht unerwähnt bleiben, dass es einige merkwürdig anmutende SGML-Konstrukte gibt, die ein SGML-Validator als gültig erkennt, die Darstellung von Dokumenten jedoch gehörig durcheinanderbringt. Nicht nur

```
<p><em>Foo</em></p>
```

ist gültiges HTML. Webautoren dürften auch

```
<p><em/Foo/</p>
```

schreiben (Stichwort: Shorttags), es gibt jedoch kaum ein Benutzerprogramm, das dieses Element dann noch anzeigen kann.

3.2.5 Schema-Validator

XML Schema Webautoren, die XHTML schreiben, können die meisten Validatoren derzeit nur mit Einschränkungen verwenden. Die Gründe liegen zum einen in den angesprochenen Beschränkungen von DTDs, zum anderen darin, dass es signifikante Unterschiede zwischen SGML und XML gibt, die bisher nicht weitreichend genug beachtet werden. Beim W3C-Validator wird es noch eine Weile dauern, bis er über eine vollständige XML-Unterstützung verfügt. Es gibt

allerdings bereits Abhilfe. Die W3C-Notiz *XHTML 1.0 in XML Schema*³ vom September 2002 definiert die XHTML-Varianten Strict, Transitional und Frameset als XML Schema, einem XML-Dokumenttyp, mit dem sich die Struktur und die Daten eines XML-Dokuments wesentlich genauer beschreiben und prüfen lassen als mithilfe von DTDs. Es können unter anderem Datentypen sowie genaue Wertebereichsbeschränkungen angegeben und die formale Grammatik von XHTML dadurch deutlich präziser beschrieben werden. Christoph Schneegans stellt einen Schema-Validator⁴ bereit, über den XHTML-Dokumente gegen das entsprechende Schema validiert werden können. Dabei werden Fehler aufgedeckt, die herkömmliche Validatoren nicht finden. Der W3C-Validator beanstandet den fehlerhaften Wert des `width`-Attributs nicht:

```
<p></p>
```

Der Schema-Validator hingegen erkennt den Fehler und wirft eine Fehlermeldung: *The 'width' attribute is invalid. The value 'bar' is invalid according to its datatype 'http://www.w3.org/1999/xhtml:Length'. The Pattern constraint failed.*

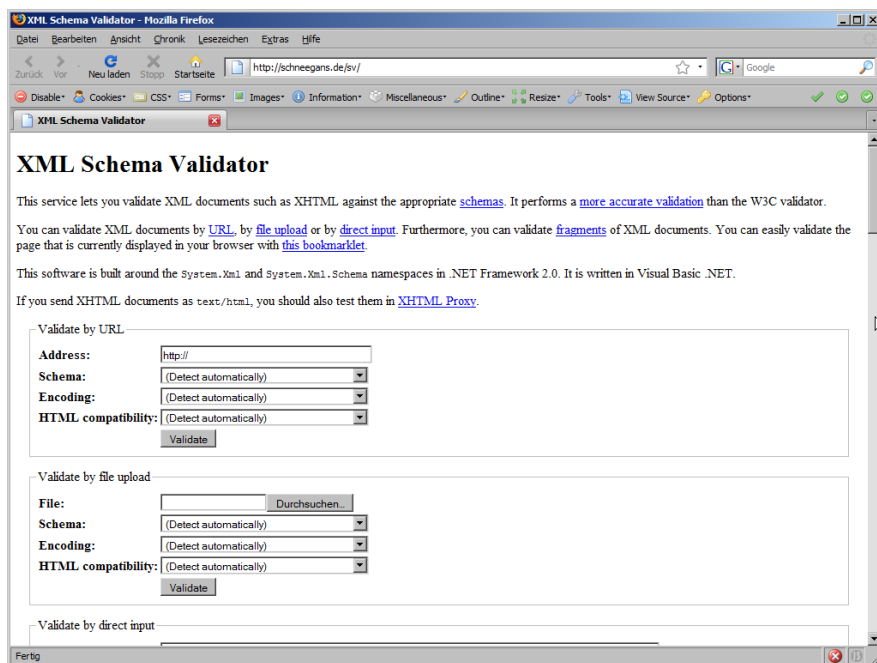


Abbildung 3.6: Der Schema-Validator von Christoph Schneegans

³ <http://www.w3.org/TR/xhtml1-schema>

⁴ <http://www.schneegans.de/sv/>

3.2.6 Grenzen

Ein weiterer wichtiger Aspekt sollte nicht unerwähnt bleiben: Validatoren überprüfen, ob Dokumente syntaktisch korrekt sind und sich an das vorgegebene Vokabular halten; sie können aber nicht beurteilen, ob Webautoren alle Informationen sinnvoll strukturiert haben oder ob Dokumente barrierefrei und zugänglich sind. Sie könnten auf nahezu alle strukturellen Elemente zugunsten einer reinen `div`- und `span`-Suppe verzichtet oder Dutzende von Layouttabellen ineinander verschachtelt haben – das Dokument wäre gültig. Folgender Schachtelsatz⁵ ist in jeder Hinsicht formal korrekt, dabei aber vollkommen unverständlich.

»Denken Sie, wie schön der Krieger, der die Botschaft, die den Sieg, den die Athener bei Marathon, obwohl sie in der Minderheit waren, nach Athen, das in großer Sorge, ob es die Perser nicht zerstören würden, schwebte, erfochten hatten, verkündete, brachte, starb!«

Niemand würde solche Sätze formulieren. Aber es gibt genügend Webautoren, die vergleichbares Markup schreiben. Prägen Sie sich ein: Gültigkeit ist nur ein Aspekt guten Webdesigns – aber längst nicht der wichtigste.⁶

3.2.7 CSS-Validator

Natürlich gibt es auch CSS-Validatoren, also Werkzeuge, die es ermöglichen, Ihre CSS-Stylesheets sowie in XHTML-Dokumente eingebettetes CSS in Bezug auf Standardkonformität zu überprüfen. Der bekannteste und beste CSS-Validator ist der CSS-Validierungsservice des W3C⁷. Wie der *W3C Markup Validation Service* bietet auch er mehrere Möglichkeiten der Validierung.

Bevor Sie Ihr CSS überprüfen, sollten Sie zunächst sicherstellen, dass die Dokumente, auf die Sie das CSS anwenden, korrekt sind. Fehler im XHTML können zur Folge haben, dass unerwünschte und unerwartete Effekte auftreten, obwohl das CSS fehlerfrei ist.

Auch für die CSS-Validierung gilt: Beheben Sie einen Fehler nach dem anderen und führen Sie mehrere Validierungsabläufe durch, bis Ihre Dokumente und Stylesheets fehlerfrei sind.

Weitere Tipps zur Benutzung des W3C-Online-Validators sowie Erläuterungen zu den häufigsten Fehlermeldungen finden Sie auf der Website von Klaus Langenberg unter der Adresse <http://www.thestyleworks.de/basics/validation.shtml>.

⁵ entnommen aus [Schneider 2001]

⁶ siehe auch [Korpela 2008]

⁷ <http://jigsaw.w3.org/css-validator/>

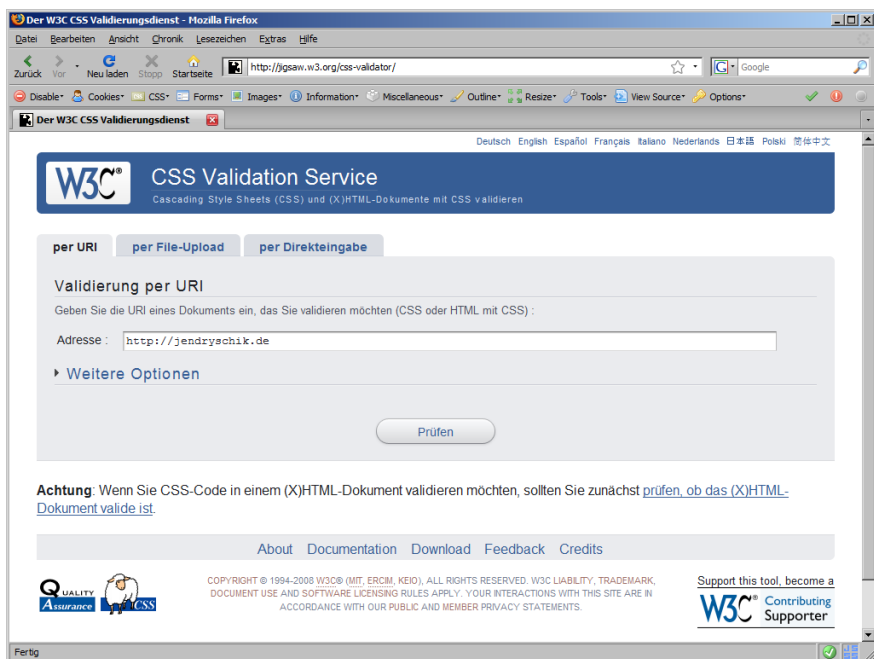


Abbildung 3.7: Der CSS-Validierungsservice des W3C

3.3 Bildbearbeitung

Im Web haben sich vier Dateiformate für Grafiken durchgesetzt: die Formate GIF, JPEG, PNG und – mit Einschränkungen – auch SVG. Die ersten drei Formate beschreiben Rastergrafiken, das vierte ein Vektorformat.

3.3.1 Rastergrafiken

DEFINITION

Rastergrafiken (auch: Pixelgrafiken, Bitmaps) sind eine Methode zur Beschreibung zweidimensionaler Bilder und bestehen aus einer matrixförmigen Anordnung von Pixeln, denen jeweils eine Farbe zugeordnet ist. Die Hauptmerkmale einer Rastergrafik sind die Breite und die Höhe in Pixeln sowie die Farbtiefe. Rastergrafiken eignen sich zur Repräsentation komplexer Bilder, beispielsweise Fotos.

GIF

• • • DEFINITION

GIF (Graphics Interchange Format) ist ein digitales Bildformat mit guter, verlustfreier Komprimierung für Bilder mit geringer Farbtiefe (bis zu 256 Farben). Das Format wurde 1987 vom US-Onlinedienst CompuServe eingeführt und ist heute vor allem wegen seiner effizienten Kompression LZW (Lempel-Ziv-Welch-Algorithmus) populär.

GIF ermöglicht das Speichern mehrerer Bilder in einer einzigen Datei, was vor allem für einfache Animationen verwendet wird. Darüber hinaus ist es möglich, eine Farbe der GIF-Grafik als transparent zu definieren, das heißt, alle Elemente, die unter der Grafik liegen, scheinen an den Stellen durch, an denen die Grafik transparent ist.

Anwendung GIF ist aufgrund seiner geringen Farbtiefe nicht für hochauflösende Grafiken und Fotos geeignet, dagegen hervorragend für weniger aufwendige Grafiken wie Hintergrundverläufe, Icons, Schaltflächen, Symbole und Zeichen.

JPEG

• • • DEFINITION

JPEG ist ein 1992 durch die Joint Photographic Experts Group entwickeltes und nach ihr benanntes standardisiertes Verfahren zur sowohl verlustbehafteten wie verlustfreien Kompression von digitalen Grafiken.

Anwendung Das Format unterstützt Grafiken mit hohen Farbtiefen (24 Bit) und eignet sich damit optimal zum Speichern von Fotos. Nicht geeignet ist JPEG zur Speicherung von Grafiken, in denen es viele benachbarte Bildpunkte mit exakt identischen Farbwerten, wenigen Farben und harten Kanten gibt (zum Beispiel Strichzeichnungen), sowie von reinen Schwarz-Weiß-Bildern ohne Graustufen.

Komprimierung JPEG bietet ein standardisiertes, verlustbehaftetes Kompressionsverfahren, den *Baseline Codec*, der im Wesentlichen wie folgt arbeitet: Es wird nach redundanten Informationen gesucht, wie zum Beispiel Ähnlichkeiten oder sich wiederholenden Mustern, die dann herausgerechnet werden, möglichst ohne die Bildqualität zu verringern. Dadurch lässt sich die Dateigröße um rund 90 Prozent reduzieren, ohne dass das ungeschulte Auge eine Verringerung der Qualität bemerken würde. Eine Kompression auf 60 Prozent ist ein guter Richtwert.

PNG

DEFINITION

PNG (Portable Network Graphics) ist ein universelles Format, das die Vorteile von GIF und JPEG vereint und über noch weitere interessante Eigenschaften verfügt. Im Jahr 1995 wurde mit der Entwicklung begonnen. Im März 1997 wurde PNG als RFC 2083 [RFC-2083 1997] veröffentlicht. Die aktuelle W3C-Empfehlung PNG 1.1 stammt vom Januar 2003 [REC-PNG 2003].

PNG kann wie GIF Pixel aus einer Farbpalette mit bis zu 256 Einträgen verarbeiten. Darüber hinaus ist die Speicherung von Graustufenbildern mit 1, 2, 4, 8 oder 16 Bit und Farbbildern mit 8 oder 16 Bit pro Kanal (also 24 beziehungsweise 48 Bit pro Pixel) möglich. PNG ermöglicht das Abspeichern zusätzlicher Informationen in der Grafikdatei, zum Beispiel Autoren- und Urheberhinweise, und unterstützt Farbkorrekturmechanismen, die sicherstellen, dass ein Bild auf verschiedenen Systemen annähernd gleich aussieht. Während es bei GIF nur möglich ist, einer einzigen Farbe vollständige Transparenz zuzuweisen, können PNG-Grafiken Transparenzinformationen für jedes einzelne Pixel oder für jede Farbe der Farbpalette enthalten. Dabei sind 8 oder 16 Bit, also 256 oder 65.536 Transparenzabstufungen möglich.

Eigenschaften

Veraltete Browser, darunter der Internet Explorer bis Version 6, haben Probleme mit der Darstellung von Transparenzen bei PNG-Grafiken. Aus diesen Gründen konnte sich das Format lange nicht so durchsetzen, wie es zu wünschen gewesen wäre. Mittlerweile kann man von einer guten PNG-Unterstützung in aktuellen Browsern sprechen, und für den IE 6 gibt es mehrere **Workarounds**. Zu den besten gehört PNG Behavior⁸.

Browserunterstützung

Software

Auch Webautoren benötigen Programme zum Erstellen und Bearbeiten digitaler Fotos und Grafiken, schließlich kommt kaum eine professionelle Website ohne Grafiken aus – angefangen beim Logo über Farbverläufe, Schatteneffekte und abgerundete Ecken bis hin zu Symbolen, Illustrationen und Fotografien. Auch wenn es nur darum geht, fertige Designvorlagen zu »schneiden«, das heißt einzelne grafische Elemente freizustellen, passend zu skalieren und im geeigneten Format abzuspeichern, benötigt man ein Bildbearbeitungsprogramm, mit dem man diese Arbeiten verrichten kann.

Das wichtigste Programm im Bereich der professionellen Bildbearbeitung ist *Adobe Photoshop* des US-amerikanischen Softwareunternehmens *Adobe Systems*.

Adobe
Photoshop

⁸ <http://webfx.eae.net/dhtml/pngbehavior/pngbehavior.html>

Aufgrund seines hohen Preises (rund 1.000 Euro) ist es für Privatanwender jedoch nahezu unerschwinglich. Für knapp ein Zehntel des Preises gibt es mit *Adobe Photoshop Elements* mittlerweile eine abgespeckte, deutlich günstigere Version, die für die Belange von Heimanwendern optimiert wurde. Gegenüber Photoshop fehlen zwar einige Funktionen, aber Anwender, die wie Webautoren ohnehin nur einen kleinen Teil der Funktionen benötigen, haben dadurch keinen Nachteil. Für Photoshop steht eine Vielzahl hilfreicher Einsteiger-Tutorials im Web zur Verfügung, unter anderem bei [Webmasterpro.de](http://www.webmasterpro.de)⁹ oder [PSD-Tutorials.de](http://www.psd-tutorials.de)¹⁰.

GIMP Eine kostenlose Alternative ist *GIMP (GNU Image Manipulation Program)*, ein Bildbearbeitungsprogramm, das als freie Software unter der GNU General Public Licence veröffentlicht wird und als ein Vorzeigeprogramm der Open-Source-Bewegung angesehen wird. Der Leistungsumfang ist mit dem Adobe Photoshops vergleichbar. Bekannt wurde das Programm hauptsächlich dadurch, dass Larry Ewing den Pinguin Tux, das bekannte Linux-Maskottchen, mit GIMP entwickelt hat. Anfänglich arbeiteten nur zwei Programmierer, die kalifornischen Informatikstudenten Peter Mattis und Spencer Kimball, an dem Projekt, mittlerweile geht die Anzahl der Entwickler und Tester in die Tausende.

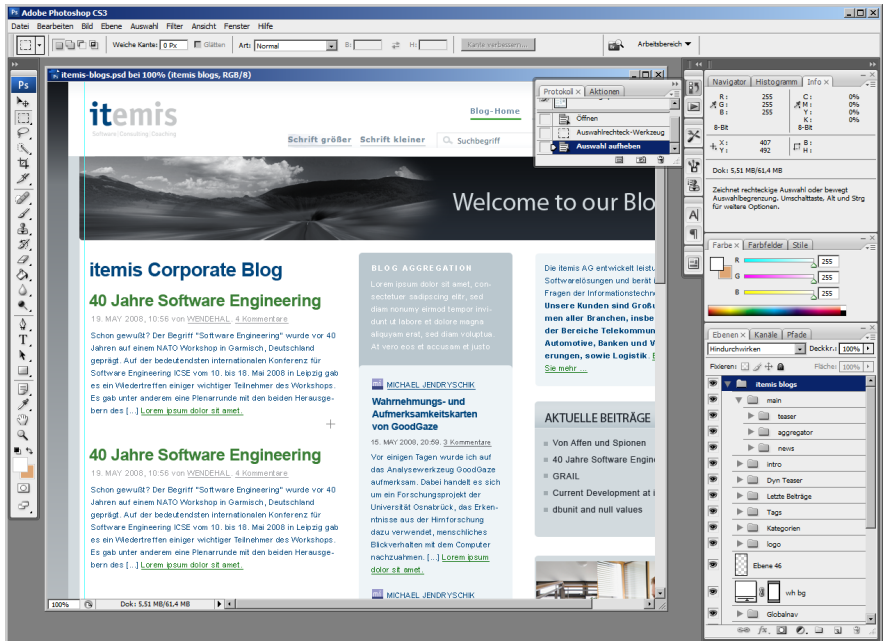


Abbildung 3.8: Adobe Photoshop CS3

9 <http://www.webmasterpro.de/design/photoshop.html>
 10 <http://www.psd-tutorials.de>

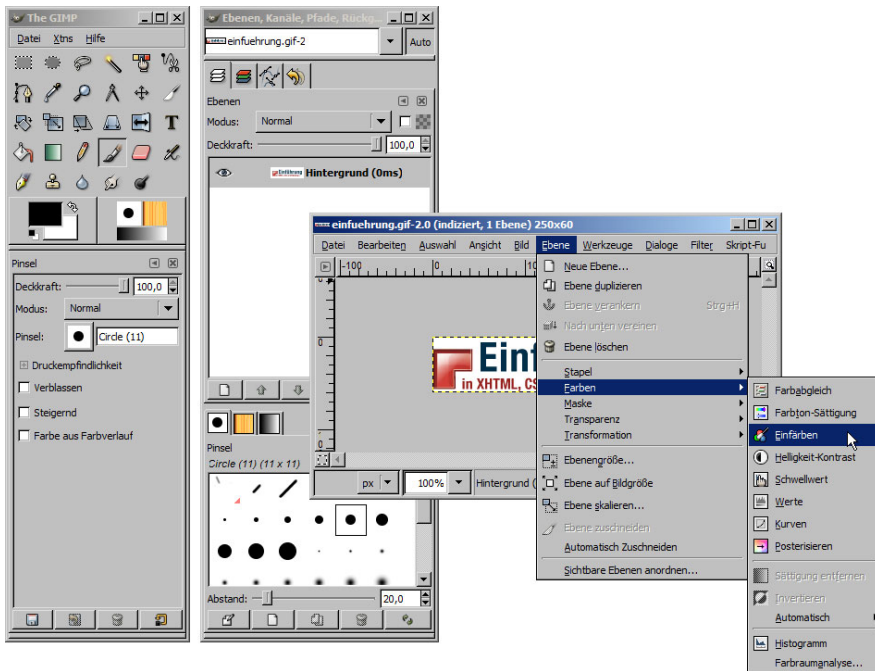


Abbildung 3.9: Das freie Bildbearbeitungsprogramm GIMP

Die aktuelle Version können Sie von der GIMP-Homepage unter der Adresse <http://www.gimp.org> herunterladen. Das deutschsprachige Benutzerhandbuch leistet Unterstützung bei der Einarbeitung. Sie finden es unter der Adresse <http://docs.gimp.org/de/>.

3.3.2 Vektorgrafiken

DEFINITION

Eine **Vektorgrafik** ist ein zwei- oder dreidimensionales Computerbild, das aus sogenannten grafischen Primitiven wie Linien, Kreisen und Polygonen geometrisch modelliert ist. Um beispielsweise einen Kreis zu zeichnen, braucht es lediglich Informationen über die Koordinaten des Kreismittelpunkts sowie die Länge des Durchmessers. Diese Informationen bleiben im Bild erhalten und können gegebenenfalls durch weitere ergänzt werden, beispielsweise Informationen über Strich- und Füllfarben sowie Muster.

Vektorgrafiken können im Gegensatz zu Rastergrafiken ohne Qualitätsverlust **Anwendung** stufenlos skaliert und verzerrt werden. Allerdings sind sie ungeeignet für die Darstellung komplexer Grafiken wie Fotos, da diese sich kaum mathematisch modellieren lassen.

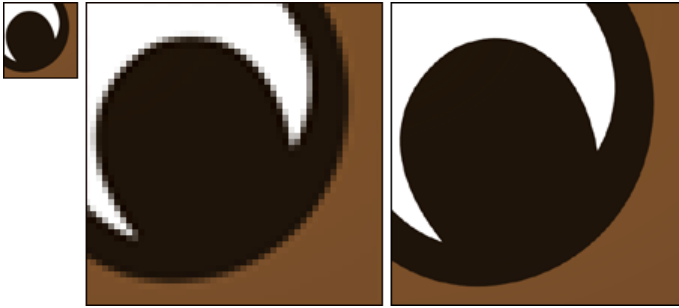


Abbildung 3.10: Rastergrafiken werden bei Vergrößerung pixelig, Vektorgrafiken hingegen nicht

Im Web haben sich zwei vektorgrafische Formate durchgesetzt: Flash, ein Grafik- und Animationsformat der amerikanischen Firma Macromedia, das hier nicht weiter behandelt wird, und SVG.

SVG

• • • DEFINITION

SVG wurde im September 2001 als W3C-Empfehlung Scalable Vector Graphics (SVG) 1.0 Specification veröffentlicht. Aktuell ist Version 1.1 vom September 2003 [REC-SVG 2003]. SVG-Dateien sind XML-Dokumente und können daher – wie XHTML-Dokumente – mit einem gewöhnlichen Texteditor erstellt und bearbeitet werden.

Ressourcen Einführungen wie Ralf Pohlmanns SVG-Tutorial¹¹, Marco Rosenthals SELFSVG¹² oder die Site scale-a-vector.de von Petra Kukofka¹³ erleichtern den Einstieg. Dr. Thomas Meinikes *Learning By Coding*¹⁴ zeigt anhand vieler einfacher Beispiele, was mit SVG so alles möglich ist.

Es gibt mittlerweile spezielle Programme wie das kostenlos verfügbare Inkscape¹⁵, die die Tipparbeit zum größten Teil abnehmen und die Erstellung von SVG-Grafiken deutlich vereinfachen.

11 <http://svg.tutorial.aptico.de>

12 <http://www.selfsvg.info>

13 <http://www.scale-a-vector.de>

14 <http://svglbc.datenverdrahten.de>

15 <http://www.inkscape.org/>



Abbildung 3.11: Mit Inkscape lassen sich verhältnismäßig einfach anspruchsvolle SVG-Grafiken erstellen

Viele Browser können SVG darstellen, dazu gehören Opera ab Version 8, Safari 3, Konqueror sowie Firefox ab Version 1.5. Andere Browser benötigen ein spezielles Plug-in wie den kostenlosen *Adobe SVG Viewer*¹⁶, dessen Entwicklung jedoch Anfang 2009 eingestellt wird. Aufgrund des enormen Potenzials ist es allerdings nur eine Frage der Zeit, bis Alternativen zur Verfügung stehen werden. Der Renesis Player¹⁷ der examotion GmbH könnte eine solche sein.

Browserunter-
stützung

Dadurch dass SVG ein Klartextformat ist, können Programmierer und Grafiker SVG-Grafiken serverseitig generieren, sie mit Datenbanken kommunizieren lassen, die DOM-Struktur (siehe Kapitel 4.3) einer Grafik clientseitig per Script manipulieren, auf Benutzereingaben reagieren und vieles mehr. SVG-Grafiken können auch aus anderen XML-Formaten generiert werden.

¹⁶ <http://www.adobe.com/svg/viewer/install/main.html>

¹⁷ <http://www.examotion.com>

3.4 Nützliche Werkzeuge

3.4.1 Tidy

• • • DEFINITION

Tidy (heißt so viel wie sauber, ordentlich) ist ein kleines und wirkungsvolles Werkzeug, das hauptsächlich dazu dient, (X)HTML-Dokumente zu prüfen und auf Wunsch automatisch so aufzubereiten, dass sie der Prüfung eines Validators (siehe Kapitel 3.2) standhalten. Tidy erkennt fehlerhafte Syntax, falsche oder redundante Auszeichnungselemente und beseitigt je nach Konfiguration die Fehler automatisch. Das Werkzeug bietet erstaunlich viele Funktionen und ist gut konfigurierbar.

- Funktionen ▶ Auf Wunsch konvertiert Tidy HTML in XHTML und umgekehrt,
- ▶ ändert die Zeichenkodierung,
- ▶ wandelt Attribute zur visuellen Auszeichnung von Inhalten, beispielsweise Angaben zu Schrift oder Farbe, in CSS-Angaben um,
- ▶ entfernt unerwünschte und veraltete Elemente,
- ▶ formatiert den Quelltext neu,
- ▶ korrigiert durch sogenannte WYSIWYG-Editoren (siehe Kapitel 3.1.1) produziertes, teilweise unsauberes oder gar fehlerhaftes (X)HTML und
- ▶ überführt durch Microsoft Word erzeugte HTML-Dokumente – in der Regel haarsträubend verkompliziertes HTML, durchsetzt von für viele Browser nutzlosen proprietären Auszeichnungselementen – in schlanke, gültige Dokumente.

Unter der Adresse tidy.sourceforge.net können Sie Tidy herunterladen. Beachten Sie, dass zwar ab und zu binäre Versionen von Tidy bereitgestellt werden, allerdings nicht regelmäßig und nicht für alle Plattformen. Um immer die aktuelle Version im Einsatz zu haben, nutzen Sie daher stets die allerneueste Version aus dem CVS des Projekts. Näheres dazu finden Sie auf der betreffenden CVS-Webseite des Tidy-Projekts: http://sourceforge.net/cvs/?group_id=27659. Die Projektverantwortlichen pflegen kein Versionierungsschema. Die Version von Tidy ist immer das Datum der letzten Änderung.

- Entwicklung Ursprünglich wurde Tidy von Dave Raggett, dem (Ko-)Autor von HTML 3 bis 4.01, als »HTML Tidy« unter dem Dach des W3C entwickelt. Der inzwischen verstorbene Terry Teague prägte in der Anfangszeit das Projekt durch eigenen Code entscheidend mit und half, es auf mehrere Plattformen zu portieren. Maßgeblich treiben inzwischen Arnaud Desitter und Björn Höhrmann in

enger Abstimmung mit dem W3C das Projekt als Open Source voran. Einige Entwickler arbeiten parallel engagiert in anderen Projekten des Konsortiums mit, unter anderem am *Markup Validation Service*, wodurch Webentwickler mehr und mehr durch einheitliche Fehlererkennung und -behandlung beider Werkzeuge profitieren.

Die *TidyLib*, das zentrale Herz aller Tidy-Erscheinungsformen, ist für viele Plattformen in unterschiedlichen Ausführungen erhältlich. Die wohl meistverbreitete und ursprünglichste Erscheinungsform ist die als eigenständiges Programm, das vor allem für Windows erhältlich ist. Aber auch verschiedene Linux-Distributoren bieten Tidy an, ebenso wie es mit *Balthisar Tidy*¹⁸ eine binäre Cocoa-Version für Mac OS X gibt. In vielen WYSIWYG- und Texteditoren ist Tidy integriert oder lässt sich leicht manuell einbinden. Einigen aktuellen Linux-Distributionen liegt es als Paket bei. Für Mozilla und Firefox gibt es eine hervorragende Erweiterung namens *Html Validator*, die im nächsten Abschnitt, »Browsererweiterungen«, vorgestellt wird.

Automatisierter Einsatz

Meist nutzen Webautoren Tidy zur Bearbeitung einzelner Dokumente. Alle Dokumente einer Website auf diese Weise zu überprüfen, kann bei größeren Projekten mit Hunderten oder Tausenden von einzelnen Dokumenten umständlich sein. Tidy lässt sich jedoch automatisiert einsetzen, im einfachsten Fall über die Kommandozeile. Im Folgenden sei davon ausgegangen, dass Tidy für Windows als `C:\tidy\tidy.exe` vorliegt und im Suchpfad für ausführbare Programme eingetragen ist. Das Kommando

```
tidy -f ausgabe.txt -q foo.html
```

schreibt Fehlermeldungen und Warnungen in eine Datei und blendet die weiteren Informationszeilen zur Datei und zu Tidy aus. Bei einem fehlerfreien Dokument ist `ausgabe.txt` leer. Eine Überprüfung von Listing 3.1 führt zu einer Ausgabe, die unter anderem die Warnungen `missing <!DOCTYPE> declaration` und `missing </title> before </head>` enthält. Eine automatische Korrektur der untersuchten Datei erledigt der Parameter `-m`.

```
<html>
  <head><title>Tidy Beispiel</head>
  <body>Hallo Welt!</body>
</html>
```

Listing 3.1: Fehlerhaftes HTML-Dokument

¹⁸ <http://www.balthisar.com/products/tidy>

Sämtliche Konfigurationsoptionen sind in der Quick Reference¹⁹ aufgeführt. Sie lassen sich jedem Aufruf direkt mitgeben oder, komfortabler, in einer Konfigurationsdatei notieren. Listing 3.2 zeigt eine solche. Die Angabe

```
tidy -m -config c:\tidy\tidy.cfg foo.html
```

wandelt Listing 3.1 in das XHTML-Dokument in Listing 3.3 um.

```
doctype: strict
enclose-text: true
indent: true
indent-spaces: 2
language: de
output-xhtml: true
tab-size: 2
tidy-mark: false
write-back: true
```

Listing 3.2: Tidy-Konfigurationsdatei

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>
      Tidy Beispiel
    </title>
  </head>
  <body>
    <p>
      Hallo Welt!
    </p>
  </body>
</html>
```

Listing 3.3: Listing 1 nach Tidys Korrektur

Folgende Zeile wendet Tidy auf alle Dokumente mit der Dateinamenserweiterung `html` im aktuellen Verzeichnis und allen Unterverzeichnissen an – unter Berücksichtigung einer Konfigurationsdatei:

```
for /R %f in (*.html) do tidy.exe -m -config C:\tidy\tidy.cfg %f
```

Weitere Portierungen Es geht jedoch produktiver. Viele Webentwickler nutzen für ihre Projekte eine lokale Entwicklungsumgebung, vor allem für Testzwecke. Idealerweise spiegelt sie die Bedingungen der späteren Serverumgebung für die jeweiligen Webprojekte wider. Zwei Alternativen bieten sich für solche Umgebungen an:

¹⁹ <http://tidy.sourceforge.net/docs/quickref.html>

Tidy als PHP- und als Webservererweiterung. In beiden Fällen durchläuft die (X)HTML-Ausgabe automatisch zunächst die Tidy-Bibliothek, wird von ihr geprüft und je nach Konfiguration überarbeitet. Entwickler oder Anwender merken bei fehlerfreien Dokumenten im Idealfall nichts von alledem, höchstens eine leichte zeitliche Verzögerung. Dies stellt eine große Arbeitserleichterung dar und ermöglicht ein zügiges und kostensparendes Entwickeln von Webseiten.

Dank John Coggeshall existiert Tidy als PECL-Erweiterung²⁰ für die Versionen 4.3+, die als *shared module* eingebunden wird. Seit PHP 5.0+ ist diese Tidy-Erweiterung Bestandteil der PHP-Distribution und muss nur noch in der Konfiguration aktiviert werden (vorausgesetzt, die Tidylib- beziehungsweise libtidy-Bibliothek ist installiert). Erfahrene PHP-Programmierer können Tidy daher leicht in ihre Projekte integrieren.

Sebastian Tusk hat vor Jahren ein Modul für den Apache 2 Webserver entwickelt: *mod_tidy*²¹. Da Tidy als PHP-Erweiterung nur in einer PHP-Entwicklungsumgebung funktionieren kann, bringt der Einsatz als Apache-Modul größeren Nutzen, weil es auf diese Weise jede Art von Webdokument automatisch überprüfen kann und sein Einsatz nicht auf PHP-Dokumente beschränkt bleibt. Mittlerweile hat Sierk Bornemann die Arbeit an *mod_tidy* übernommen.

Im Java Server Faces AJAX Framework ist eine gut gepflegte Tidy-Klasse enthalten, um innerhalb von Java Server Faces die Eigenschaften von Tidy zu nutzen. Für Java existiert mit *Jtidy*²² außerdem eine direkte Tidy-Portierung, doch leider wird diese seit Längerem nicht mehr gepflegt und ist deswegen, was die Aktualität des Tidy-Codes betrifft, nicht mehr auf der Höhe der Zeit. Sie sei hiermit der Vollständigkeit halber trotzdem erwähnt.

Für die populäre Entwickler-IDE Eclipse existiert mit dem Tidy-Eclipse-Plugin²³ eine Möglichkeit, Eclipse um die Prüfmöglichkeiten von Tidy zu bereichern.

Ein Webautor, der lediglich ab und zu einzelne Webseiten erstellt, ist mit dem eigenständigen Programm oder der Version als Browsererweiterung gut bedient. Für solche Fälle lohnt sich der Aufbau einer lokalen Entwicklungsumgebung meistens nicht, oder der Autor könnte damit überfordert sein. Für Webautoren, die regelmäßig größere Webprojekte zu bearbeiten haben, sind die Eigenschaften der PHP-Erweiterung, des Apache-Moduls, der betreffenden Java-Klassen oder einfach der Firefox-Erweiterung im Alltag hilfreich.

20 <http://www.coggeshall.org/oss/tidy/index.php/4/>

21 <http://mod-tidy.sourceforge.net>

22 <http://jtidy.sourceforge.net>

23 <http://eclipsetidy.sourceforge.net>

Wenigstens eines dieser Werkzeuge zum Kontrollieren und gegebenenfalls Korrigieren der HTML-Syntax sollte in keiner professionellen Entwicklungsumgebung fehlen.

3.4.2 FTP

Ihre Webseiten erstellen Sie höchstwahrscheinlich lokal auf Ihrem Rechner. Damit ist es unwahrscheinlich, dass Sie – wie es ja im Allgemeinen gewünscht ist – für jeden Nutzer zu jeder Zeit erreichbar sind; schließlich wollen Sie Ihren Rechner gewiss nicht 24 Stunden pro Tag durchlaufen lassen und permanent online sein. In der Regel mieten Sie entsprechenden Speicherplatz bei einem Hosting-Provider an, der über eine entsprechende Technik und Anbindung verfügt. Dazu zählen in der Regel vollklimatisierte Serverräume, Brandschutz, unterbrechungsfreie Stromversorgung, Netzwerküberwachung und regelmäßige Backups. Es gibt eine Fülle von Anbietern unterschiedlicher Qualität, die sich in Preis und Leistung zum Teil deutlich unterscheiden.

Um die lokal auf Ihrer Festplatte liegenden Dateien Ihrer Website zu publizieren, das heißt der Allgemeinheit zugänglich zu machen, müssen Sie diese auf den Ihnen zugewiesenen Speicherplatz auf den Servern Ihres Hosting-Providers überspielen. Das klassische Protokoll für diesen Zweck ist FTP.

• • • DEFINITION

FTP (File Transfer Protocol) ist ein Netzwerkprotokoll zur Übertragung von Dateien vom Server zum Client (Download), vom Client zum Server (Upload) oder clientgesteuert zwischen zwei Servern.

Von Ihrem Provider erhalten Sie alle erforderlichen Daten, die Sie benötigen, um Ihre Dateien auf den Server übertragen, sie von dort auf Ihren Rechner herunterladen oder direkt auf dem Server verwalten zu können. Zusätzlich benötigen Sie einen FTP-Client, in den meisten Fällen ein Programm, das so aufgebaut ist wie ein Dateimanager, beispielsweise der Windows Explorer.

- SFTP Falls Sie Wert auf mehr Sicherheit legen, sollten Sie auf SFTP (SSH File Transfer Protocol), eine Art »sicheres FTP«, und entsprechende Clients zurückgreifen.
- FileZilla *FileZilla*²⁴ gehört zu den besten FTP-Programmen unter Windows, ist kostenlos erhältlich und bietet auch SFTP-Unterstützung. Für Macintosh-Rechner ist *Cyberduck*²⁵ empfehlenswert.

²⁴ <http://filezilla.sourceforge.net>

²⁵ <http://cyberduck.ch>

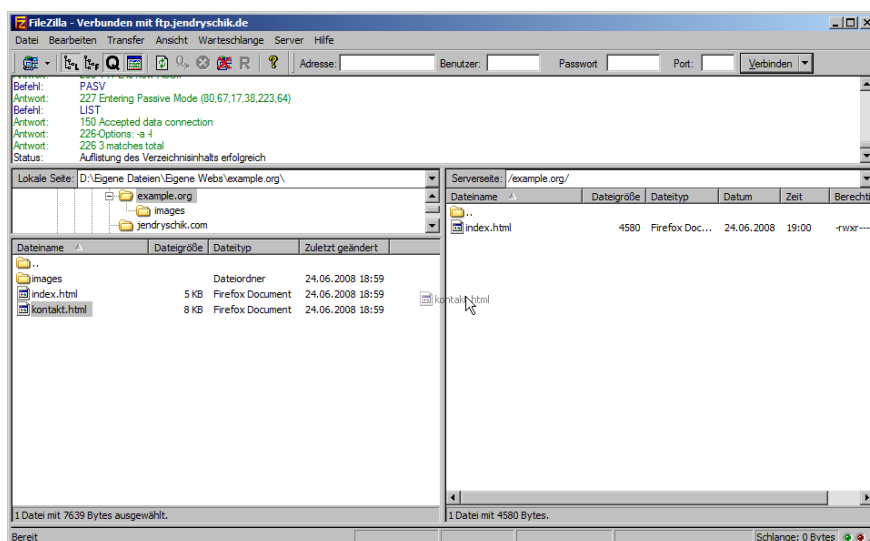


Abbildung 3.12: Der FTP-Client FileZilla

Literatur

Jürgen Schlotcke: FileZilla Dokumentation: FTP-Schnelleinstieg

Diese Anleitung zeigt Ihnen in aller Kürze die Bedienung des FileZilla-FTP-Clients.

URL: <http://www.filezilla.de/schnelleinstieg.htm>

Tim Kosse: FileZilla documentation

Vollständige englischsprachige FileZilla-Dokumentation.

URL: <http://filezilla.sourceforge.net/documentation/>

3.4.3 Xenu's Link Sleuth

Wer kennt das nicht? Man ruft eine Webressource auf – und vollkommen unerwartet findet man sich im Niemandsland des Webs wieder. Dafür kann es folgende Gründe geben:

- ▶ Der URI der Ressource wurde falsch eingetippt, entweder in der Adresszeile oder beim Anlegen des Links.
- ▶ Der Autor der Site wusste nicht, dass sich coole URIs nicht ändern,²⁶ und hat die Ressource gelöscht oder verschoben. Dadurch stimmen Lesezeichen, Links auf die Site sowie Einträge in Suchmaschinen nicht mehr.

²⁶ vgl. <http://www.w3.org/Provider/Style/URI.html>

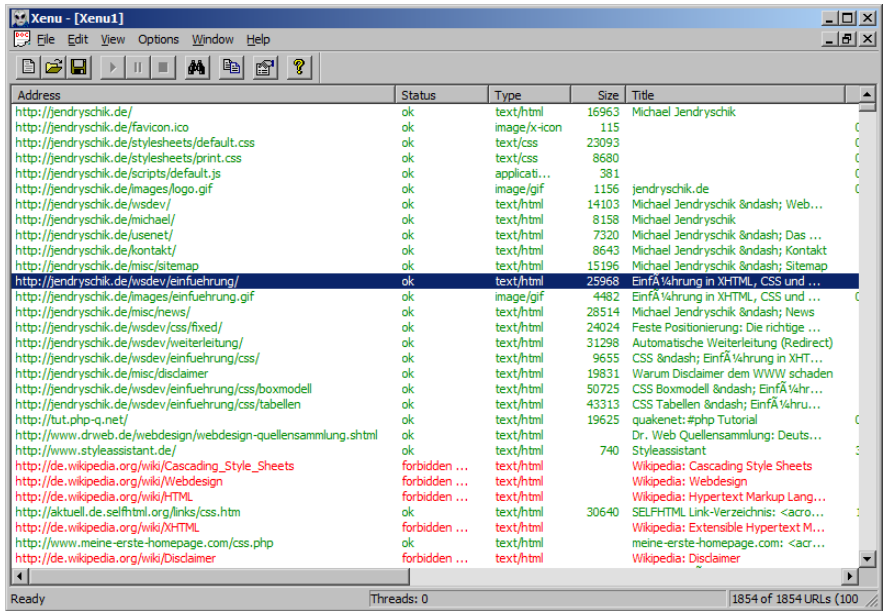


Abbildung 3.13: Xenu's Link Sleuth

Um tote Links innerhalb Ihrer Website aufzufinden, brauchen Sie nicht Dokument für Dokument einzeln zu durchsuchen und jeden Link auszuprobieren. Mit *Xenu's Link Sleuth*²⁷ von Tilman Hausherr steht ein Programm zur Verfügung, das dies automatisch und sehr komfortabel für Sie erledigt. Auf Knopfdruck können Sie eine komplette Website nach ungültigen Verweisen durchsuchen. Das Programm kann auch mit SSL verschlüsselte sowie lokale Webseiten überprüfen – gerade letzterer Umstand macht das Werkzeug so nützlich. Aus den gewonnenen Informationen wird eine Zusammenfassung in Form eines HTML-Dokuments erstellt, das im Browser angezeigt und ausgewertet werden kann.

Eine deutschsprachige Anleitung²⁸ von Christian Holm hilft Ihnen beim Einstieg.

3.4.4 Color Cop

*Color Cop*²⁹ ist ein Farbpicker, mit dem Sie mittels einer Pipette einen Farbwert eines beliebigen Punkts auf dem Desktop messen können. Der Farbwert wird als RGB- wie auch als Hexadezimalwert angegeben. Eine Bildschirmlupe hilft Ihnen dabei, den richtigen Bildschirmpunkt zu treffen.

27 <http://home.snafu.de/tilman/xenulink.html>

28 <http://www.aspheute.com/artikel/20010518.htm>

29 <http://colorcop.net>

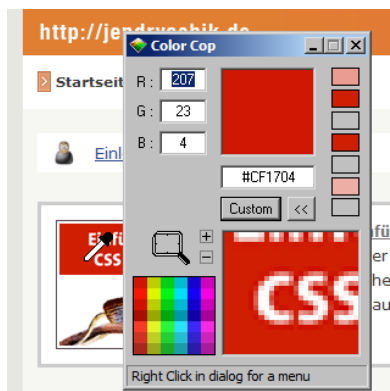


Abbildung 3.14: Color Cop

Color Cop erfüllt nur diesen einen Zweck und ist entsprechend einfach zu bedienen. Dabei gehört das Programm zu den nützlichsten Hilfsmitteln, die Sie sich als Webentwickler auf Ihrem Arbeitsrechner installieren können.

3.5 Browsererweiterungen

DEFINITION

Eine **Browsererweiterung** (Add-on, auch: Plug-in) ist ein optionales Modul, das Browserfunktionen ergänzt oder erweitert. Browserhersteller definieren Schnittstellen, mit deren Hilfe Erweiterungen Dritter eingebunden werden können. Plug-ins sind in der Regel ohne die Anwendung, für die sie geschrieben wurden, nicht funktionsfähig.

Auch Programme wie der *Flash Player* oder *Shockwave* sowie das *Reader-Plug-in* zur Anzeige von PDF-Dokumenten oder der *SVG Viewer* der Firma Adobe sind Browsererweiterungen. Im Folgenden geht es jedoch nicht um Programme, die zur Anzeige bestimmter Multimedia- oder Dokumentformate bestimmt sind, sondern um eine andere Art der Browsererweiterung: Es handelt sich hier um Erweiterungen, die den Browser um nützliche Funktionen für Webautoren ergänzen.

Erweiterungen gibt es für alle großen Browser, wirklich interessant sind jedoch nur die Erweiterungen für Browser der Mozilla-Familie, speziell für Firefox. Auf der Seite <https://addons.mozilla.org> finden Sie Hunderte von Erweiterungen aller Art. Auf Erweiterungen für andere Browser wird in diesem Abschnitt nicht eingegangen.

3.5.1 Firebug

Die Browsererweiterung Firebug³⁰ von Joe Hewitt, einem der Firefox-Entwickler, ermöglicht Webautoren, Webseiten zu »debuggen«, das heißt nach Fehlern zu suchen und diese zu korrigieren. Firebug integriert sich in Form einer Konsole in der unteren Browserhälfte – links ist der Quelltext der Seite in Form einer auf- und zuklappbaren Baumstruktur zu sehen, rechts alle (auch die vererbten) CSS-Eigenschaften. Damit können Sie anfangen, den Code Ihrer Webseite zu inspizieren. Fahren Sie mit der Maus über die Seite, und Firebug markiert den dazugehörigen Code – oder umgekehrt. Das Besondere an Firebug ist, dass Sie den Quellcode nicht nur inspizieren, sondern auch ändern können. Dazu bietet das Werkzeug die Funktion »Edit«, über die Sie markierte Teile des Quelltext korrigieren, beispielsweise CSS per Inline-Style hinzufügen können. Die Auswirkungen der Änderungen sehen Sie – und nur Sie (das ist wichtig zu wissen) – direkt live auf der Seite.

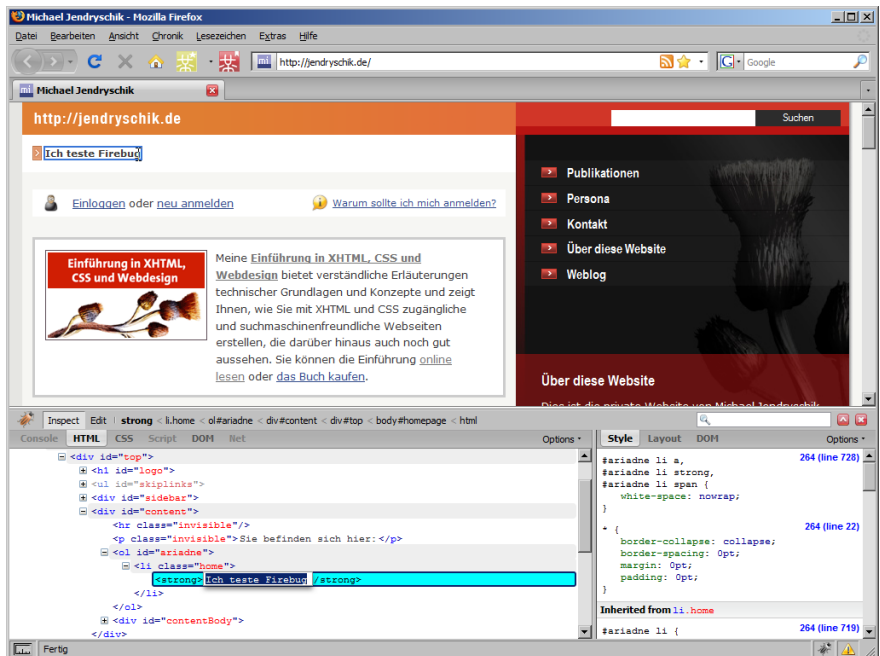


Abbildung 3.15: Mit Firebug können Webautoren live sehen, wie sich Änderungen am Quelltext auswirken

30 <https://addons.mozilla.org/de/firefox/addon/1843>

3.5.2 Web Developer Toolbar

Die *Web Developer Toolbar*³¹ von Chris Pederick gehört zu den Browsererweiterungen, auf die Sie bereits nach kurzer Zeit nicht mehr werden verzichten wollen. Für Webautoren ist ihre Installation Pflicht. Die Web Developer Toolbar basiert auf der *PNH-Developer-Toolbar*³² von Chris Casciano und verfügt über eine derartige Vielzahl von Diagnosewerkzeugen und nützlichen Hilfsmitteln, dass es eine Stunde dauern kann, jede einzelne Funktion an seiner Website auszuprobieren. Zu den interessantesten Funktionen gehören die Möglichkeiten,

- ▶ JavaScript, Java, Cookies, den Browsercache und vieles andere mit einem Mausklick aus- und wieder einzuschalten, Funktionen
- ▶ Stylesheets vollständig oder teilweise auszuschalten sowie live zu editieren (Änderungen sind direkt auf der Webseite sichtbar, auch wenn es sich nicht um eine lokale Site handelt),
- ▶ das Verhalten von Formularen zu beeinflussen,
- ▶ Informationen zu allen Bildern auf der Webseite anzuzeigen sowie die Darstellung von Bildern zu unterbinden oder auf die Anzeige ihres alternativen Attributs zu beschränken,
- ▶ das zugrunde liegende (X)HTML beziehungsweise DOM zu manipulieren und so versteckte Elemente und Kommentare anzuzeigen oder ein Dokument zu linearisieren,

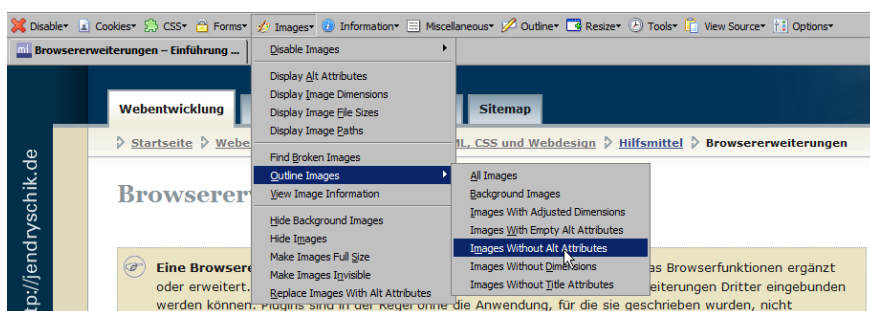


Abbildung 3.16: Die Web Developer Toolbar in Aktion

31 <http://chrispederick.com/work/webdeveloper/>

32 <http://www.placenamehere.com/pnhtoolbar/>

- ▶ Elemente zur Analyse sichtbar zu machen, etwa alle Tabellen, alle Blockelemente oder alle missbilligten Elemente, sowie
- ▶ das aktuelle Dokument – auch lokale Dokumente – direkt einer Vielzahl von Validatoren und Prüfwerkzeugen vorzulegen.

3.5.3 Html Validator (based on Tidy)

Die Erweiterung *Html Validator (based on Tidy)*³³ von Marc Gueury ergänzt die Quelltextanzeige des Firefox-Browsers um einige nützliche Funktionen. So lassen sich automatisch Fehler und Warnungen anzeigen und in einem weiteren Schritt korrigieren.

Lange Zeit als reine Implementation der TidyLib im Einsatz, beherbergt dieses Firefox-Plug-in neben der TidyLib zum Überprüfen der HTML-Syntax mittlerweile auch OpenSP, eine weitere Syntaxbibliothek aus dem Dunstkreis des W3C. OpenSP ist derselbe SGML-Parser, der im *W3C Markup Validation Service* zum Einsatz kommt. Dadurch dass dieses Firefox-Add-on nun zwei mächtige »Syntaxprüfer« an Bord hat, ist es zwar etwas größer als in der Vergangenheit, dafür aber doppelt so mächtig. Diese beiden Syntaxprüfer können, je nach Konfiguration, entweder einzeln verwendet oder sogar miteinander verkettet werden.

3.5.4 Fangs

*Fangs*³⁴ ist ein Screenreader-Emulator von Peter Krantz, also eine Erweiterung, die eine Webseite so aufbereitet, wie ein Screenreader sie vorlesen würde. Nach dessen Installation steht ein neuer Eintrag VIEW FANGS im Kontextmenü zur Verfügung, über den ein neues Fenster mit der aufbereiteten Ansicht geöffnet werden kann. Angezeigt wird unter anderem, wie viele Überschriften und Links auf einer Seite vorhanden sind und wie Links, Listen, Tabellen etc. angekündigt würden. Abbildung 3.18 zeigt die Darstellung der Google-Homepage im Fangs-Fenster.

33 <http://users.skynet.be/mgueury/mozilla/>

34 <http://www.standards-schmandards.com/index.php?show/fangs>

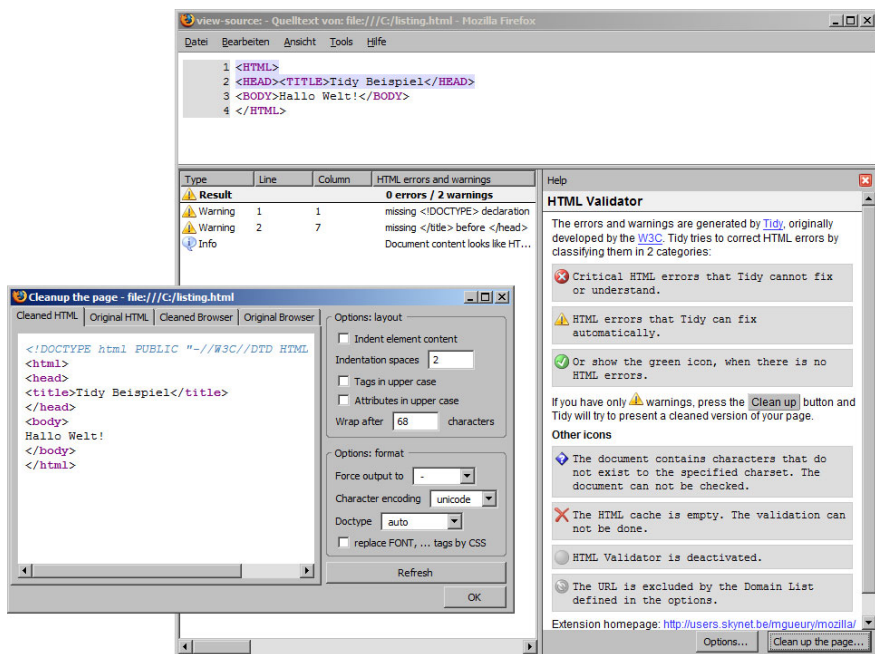


Abbildung 3.17: Der Html Validator zeigt Fehler und Warnungen an

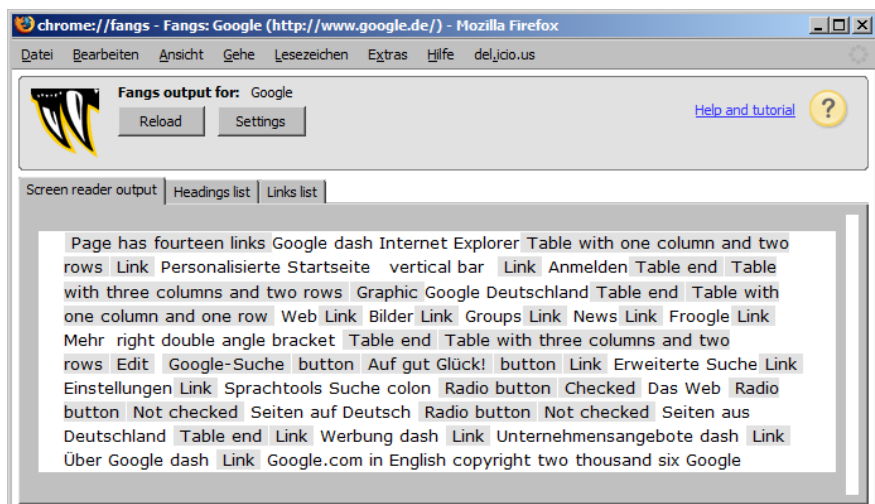


Abbildung 3.18: Der Screenreader-Emulator Fangs

3.5.5 MeasureIt

Die Erweiterung *MeasureIt*³⁵ von Kevin A. Freitas ermöglicht es, Breite und Höhe eines Bereichs einer Webseite direkt auf dem Bildschirm nachzumessen. Die maßnehmende Box lässt sich pixelweise auf dem Bildschirm verschieben.

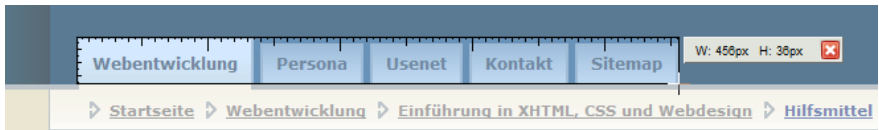


Abbildung 3.19: MeasureIt

3.5.6 LinkChecker

Wenn es darum geht, alle Links einer Website zu überprüfen, sollten Webautoren auf Werkzeuge wie *Xenu's Link Sleuth* (siehe Kapitel 3.4.3) zurückgreifen. Um sicherzustellen, dass alle Links auf einer einzigen Webseite gültig sind, reicht die Erweiterung *LinkChecker*³⁶, ebenfalls von Kevin A. Freitas, vollkommen aus. Über den Aufruf eines Eintrags im Kontextmenü werden alle Links auf der Seite durchlaufen und farbig markiert:

- ▶ Grün für gültige Links,
- ▶ Rot für ungültige,
- ▶ Gelb für Weiterleitungen oder geschützte Bereiche sowie
- ▶ Grau für Links, die nicht geprüft werden konnten und übersprungen wurden.

3.5.7 Right-Click Lynx viewer

Die Erweiterung *Right-Click Lynx viewer*³⁷, bereitgestellt durch die Yellowpipe Internet Services, fügt einen Kontextmenüeintrag hinzu, über den ein neues Fenster geöffnet werden kann, in dem die aktuelle Webseite im Lynx viewer der Yellowpipe Internet Services angezeigt wird. Webautoren können so überprüfen, wie eine Website im Textbrowser Lynx aussieht (wobei es in den meisten Fällen besser ist, Lynx zu installieren und sich direkt anzuschauen, wie die Seite darin aussieht und ob sie sich gut bedienen lässt).

35 <http://www.kevinfreitas.net/extensions/measureit/>

36 <http://www.kevinfreitas.net/extensions/linkchecker/>

37 <http://www.yellowpipe.com/yjis/tools/lynx/rightlynx/>

3.5.8 WAVE-Toolbar

Mit WAVE (Web Accessibility Evaluation Tool) stellt die WebAIM (Web Accessibility in Mind) Webentwicklern unter <http://wave.webaim.org> ein Onlinewerkzeug zur Verfügung, mit dem sie die Zugänglichkeit einer Webseite überprüfen können. Fehler oder mögliche Probleme zeigt WAVE direkt auf der zu überprüfenden Seite in Form grafischer Symbole an. Unter der Adresse <http://wave.webaim.org/toolbar> können Sie sich die WAVE-Toolbar installieren, die den Aufruf des Prüfwerkzeugs erleichtert und darüber hinaus einige weitere Funktionen bietet.

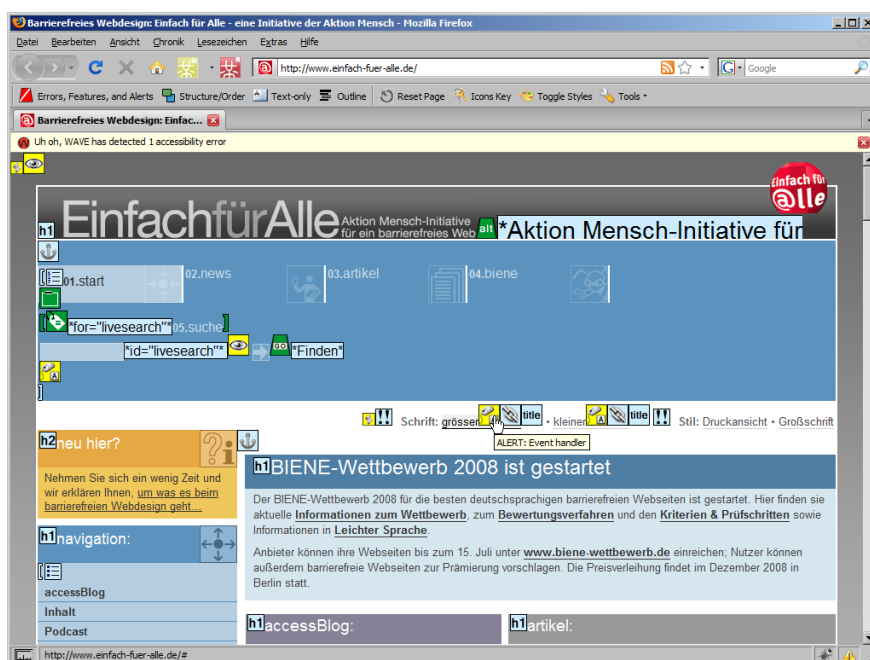


Abbildung 3.20: Die WAVE-Toolbar zeigt mögliche Accessibility-Probleme auf Webseiten an