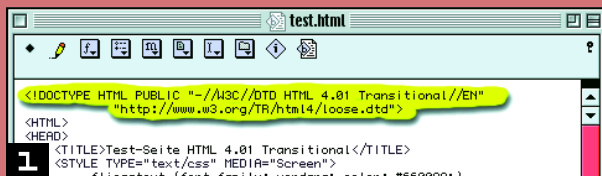


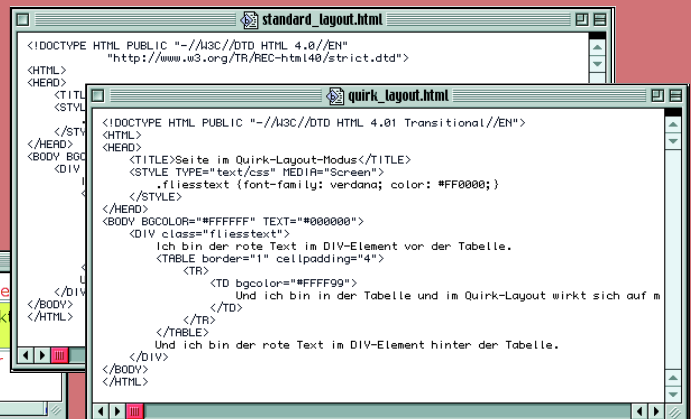
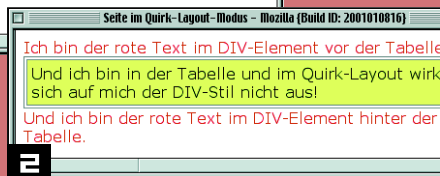
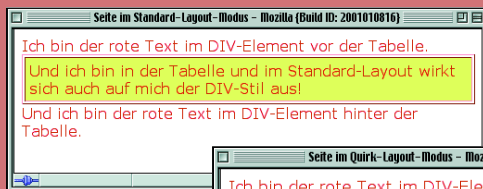
CROSS-BROWSER-PUBLISHING

Mit Netscape 6 und der Macintosh-Version von Internet Explorer 5 kamen im letzten Jahr zwei neue Browser auf den Markt, die die Webstandards des W3C erstmals genauer einhalten. Was die neue Browser-Generation der Webdesign-Praxis bringt, beleuchtet dieser SBO-Workshop.



1

Das Element „Doctype“ im HTML-Code gibt Aufschluss über die Version der verwendeten HTML-Spezifikation.



Schon die Urväter der aktuellen Browser waren nicht in der Lage, HTML-Seiten browserübergreifend identisch anzuzeigen. In der Vergangenheit hinkte die nach eigenen Aussagen so fortschrittliche Softwareindustrie mit ihren Browsern den Standardisierungsprozessen gewaltig hinterher. Erst allmählich erkennen die Browserhersteller die Notwendigkeit und die Vorteile der Einhaltung von Standards. Zur Erinnerung: Webstandards wie CSS1 verabschiedete das W3C bereits im Dezember 1996, HTML 4.0 im Dezember 1997 und CSS2 im Mai 1998. Der erste „große“ Browser, der allgemein mit dem Etikett „standardkonform“ ausgezeichnet wurde, stammt aus dem vergangenen Jahr: Microsoft veröffentlichte im März 2000 die Mac-OS-Version von Internet Explorer 5, die laut Hersteller zu zirka 85 Prozent aktuelle Webstandards unterstützt.

Dabei ist das Wort „Standard“ aus der Sicht vieler Webdesigner nicht nur positiv behaftet: Es bedeutet für die meisten nicht allein „einheitliche Darstellung in allen Browsern“, sondern steht auch als Synonym für „kleinster gemeinsamer Nenner“, „Abschied von der schönen Subnavigation nehmen“ oder auch „Kreativitätskastration“. Ganz anders sehen das die Verfechter der Standards. Sie beschreiben in ihren Reden die „Welt der Standards“ als eine „blühende Landschaft“, die viele Vorteile sowohl für den Anwender als auch für Webdesigner verspricht:

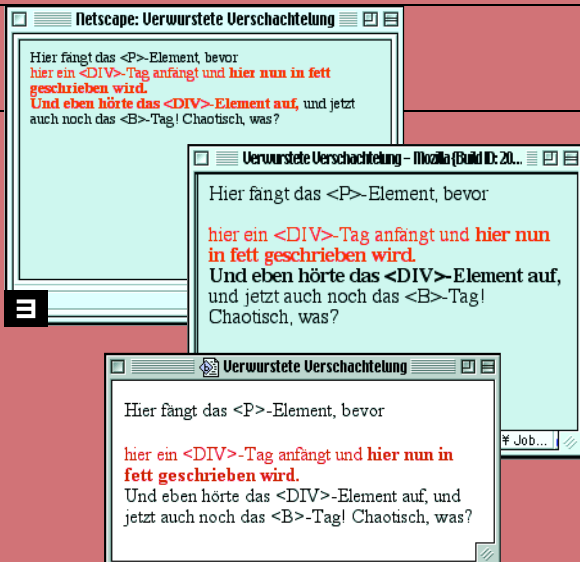
- ➔ Hat man ein Dokument erst einmal standardkonform programmiert, so lässt es sich auf allen Endgeräten darstellen, egal ob man einen PC mit 21-Zoll-Monitor oder einen hoch technisierten Internetkühlschrank nutzt. Ohne Standards wäre das Anlegen von zehn Versionen pro Seite ein kaum noch zu bewältigender Albtraum. Standards ersparen stattdessen Budget verschlingende

Bugfixes und aufwendige Workarounds. Die dafür in etwa notwendigen 25 Prozent der Entwicklungskosten könnten Webentwickler sinnvoller in die Gestaltung oder auch den weiteren Ausbau der Site investieren.

- ➔ Auch die Anwender kommen in den Genuss von mehr „Accessibility“: Standards wie Alternativtexte zu Grafiken erleichtern sehbehinderten Usern den Zugang zu einer Website und machen Grafiken auditiv erfassbar. Dabei sammelt spezielle Software die Alternativtexte und liest diese dem Sehbehinderten vor. Ein Minimum an „Accessibility“ für die Anwender ist heute teilweise sogar schon gesetzlich festgeschrieben ❶.

- ➔ Standards verleihen auch Programmen „Accessibility“ und erleichtern den externen Zugriff auf den Quellcode und damit das dynamische Generieren von HTML-Seiten aus Datenbanken. Auch das Indizieren der Seiten, etwa durch Suchmaschinen und die Verwaltung von Dokumenten, gestaltet sich dadurch einfacher.

Allerdings sind nicht für jede Website alle Standards gleich wichtig. Der Auftritt einer Tageszeitung legt zum Beispiel mehr Wert auf Zugänglichkeit für unterschiedliche Browser und Geräte, um News für eine breite Zielgruppe bereitstellen zu können. Um Aktualität zu garantieren, muss sich der Content möglichst unkompliziert dynamisch aus einer Datenbank speisen lassen. Eine Multimedia-Agentur verfolgt dagegen meist andere Ziele – sie will potenziellen Kunden zeigen, wo im Medium Internet „der Hammer hängt“, und damit die technischen Möglichkeiten der aktuellen Browser voll ausschöpfen. Dementsprechend ist hier die Bereitschaft höher, Mehraufwand für browserspezifische Lösungen in Kauf zu nehmen.



Bei unsauberer Verschachtelung der HTML-Elemente zeigen drei Browser drei unterschiedliche Ergebnisse an. Von oben nach unten: Die Mac-OS-Versionen von Netscape Navigator 4.05, Mozilla 0.7 und Internet Explorer 5.

Der einzige Unterschied zwischen den beiden HTML-Dokumenten ist der „Doctype“, der zu verschiedenen Interpretationen des Dokuments führt.

SBOKOMPAKT

- Die vom W3C empfohlenen Webstandards versprechen Vorteile für Webdesigner und -nutzer, wurden aber bisher nur teilweise von den Browserherstellern eingehalten
- Mit Netscape 6/Mozilla und Internet Explorer 5 (Mac OS) erschienen die ersten nahezu standardkonformen Browser
- Eine gute Cross-Browser-Kompatibilität erfordert häufig manuelle Nachbesserungen im Quellcode
- Effizient strukturierte Layouts und überschaubarer Quellcode erleichtern die Produktion und versprechen mehr Kompatibilität
- Frühe und häufige Tests im Browser sind essenziell wichtig und durch keine Tools zu ersetzen

1

Nationale Gesetze und Verordnungen zu Accessibility: <http://w3.org/WAI/References/Policy>

2

www.macromedia.com/support/dreamweaver/ts/documents/netscape_6.htm



Die wichtigsten Browser und einige HTML-Validatoren finden Sie auf unserer Heft-CD-ROM.

Dennoch brauchen Webdesigner nicht gleich zur „Großpackung Standards“ zu greifen und diese bis in die letzte Konsequenz einzuhalten. Vielmehr sollte man sich überlegen, welche Zielgruppe man erreichen will, und dementsprechend immer die richtigen Standards aus dem Regal nehmen. Momentan vollziehen die Webdesigner eine schwierige Lernphase, denn sie müssen sich im Übergang zu den neuen Browsern erst mit den Standards anfreunden. Außerdem nehmen es die Browserhersteller mit der Implementation der Standards lediglich insoweit genau, dass hübsche Schlagwörter für Pressemitteilungen herauspringen. Größtenteils noch völlig unbefleckt vom „S-Wort“ sind dagegen die Hersteller der populären Webeditoren, so dass in der Produktion schwierige Kompromisse gefordert sind. Die folgenden Ansätze helfen Ihnen, eine Lösung zu finden:

„Defensives“ Coden erleichtert spätere Änderungen

Zunächst sollten Sie versuchen, „defensiv“ zu coden, indem Sie sich auf einfachere, aber solide HTML-Programmierung beschränken. Die aktuellen visuellen Webeditoren wie Dreamweaver und Golive locken mit vielen Javascript- oder DHTML(Dynamic HTML)-Features, die auch Webdesigner ohne fundierte Programmiererfahrung per Menü oder Drag-and-drop anwenden können. Doch Vorsicht, hier erlebt man schnell böse Überraschungen: Mit dem Erscheinen von Netscape 6/Mozilla standen Webdesigner, die sich blind auf die Resultate von Dreamweaver oder Golive verließen, alleine da. Zum Beispiel beruhen einige Behaviors respektive Aktionen in Dreamweaver und Golive immer noch auf dem „Layer“-Element ², obwohl seit mehr als einem Jahr bekannt ist, dass Netscape 6 dieses

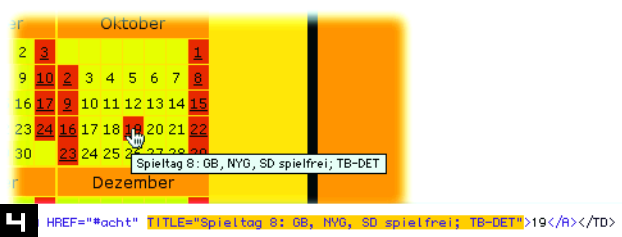
nicht mehr unterstützen wird. Die schwer wiegende Folge davon ist, dass die entsprechenden Seiten jetzt extra umgestaltet werden müssen, um in Netscape 6 fehlerfrei zu funktionieren.

Außerdem unterstützt HTML 4.0 manche Basiselemente in den gängigen HTML-Editoren nicht optimal. Diese Editoren schreiben einige in HTML 4 als „required“ markierte Attribute nicht in den Quellcode des Dokuments und tragen zum Beispiel das bereits erwähnte „Alt“-Attribut nach dem Einfügen einer Grafik auch nicht automatisch ein.

Die Konsequenz für den Webdesigner: Je „standardisierter“ das HTML-Dokument sein soll, desto weniger kann man sich auf visuelle Editoren verlassen. Oft ist es nötig, am Quellcode selbst Hand anzulegen und beispielsweise fehlende Attribute zu ergänzen oder nicht standardkonforme Stellen zu modifizieren. Dies meistern Sie allerdings nur dann, wenn Sie den Code noch „entziffern“ und vor allem nachvollziehen können. Daher sollte der Quellcode möglichst simpel und leicht verständlich bleiben. Verwenden Sie also nicht jeden Schnickschnack, den ihr HTML-Editor anbietet, denn damit fügen Sie oft schwer nachvollziehbaren Javascript-Code in das Dokument ein. Bleiben Sie auf dem Terrain, auf dem Sie sich auskennen, damit Sie im Code ihrer Seiten jederzeit auf „Handsteuerung“ wechseln können.

„KISS!“ – Keep it simple, stupid!

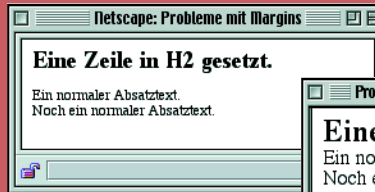
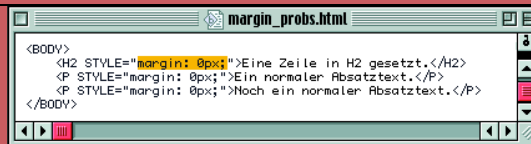
Komplexe Programmierung via Javascript und DHTML bietet zwar viele interaktive Gestaltungsmöglichkeiten, erfordert aber im Moment einen enormen Aufwand, sofern man alle gängigen Browser (Internet Explorer 4, 5 und 5.5, Netscape Navigator 4 und 6, →



4 HREF="#acht" TITLE="Spieltag 8: GB, NYG, SD spielfrei; TB-DET">19</TD>

4

Das „Title“-Attribut zeigt bei den neueren Browsern Zusatzinformationen im Kontext an.



3 www.macromedia.com/solutions/richmedia/creating/matrix.html

4 www.macromedia.com/macromedia/accessibility/tools

5 www.w3.org/People/Raggett/tidy/

→ AOL, Mozilla) unterstützen will. Oft ist in diesem Fall „Browser-switching“ die einzige Möglichkeit, um alle Probleme zu vermeiden, was bedeutet, dass man eine speziell angepasste Variante für jede Browserversion erstellen muss. Dies wiederum heißt, dass später mit jeder neu erscheinenden Browserversion aufwendige Korrekturen und Ergänzungen anfallen. Deswegen sollten Sie ihre HTML-Seiten von Anfang an optisch so effizient wie möglich gestalten.

Das beginnt bereits beim Screendesign: Vermeiden Sie Layouts, die in der späteren Umsetzung in HTML komplexe, verschachtelte Tabellen erfordern. Auch der Anwender wird sich über einen schnelleren Aufbau der Seiten freuen, weil eine einfach gestaltete Seite im Browser schneller gerendert und damit auch schneller angezeigt werden kann. Zudem ist es praktikabler, einen zunächst einfachen Aufbau weiter zu verfeinern oder weiter zu entwickeln, als ein komplexes Layout „abzuspecken“. Darüber hinaus sind simple Dokumente pflegeleichter. Da bei jeder Website häufige Aktualisierungen anfallen, ist dieser Vorteil von großer Bedeutung. Simple Seiten verhalten sich wesentlich robuster gegenüber den Eigenheiten der einzelnen Plattformen, aus diesem Grunde kommt es zu weniger Abweichungen in der Darstellung oder gar Fehlern und Abstürzen. Gehen Sie bei ihrem Projekt zunächst von dieser einfachen Grundlage aus. Später bleibt Ihnen immer noch die Möglichkeit, zu anspruchsvolleren Features, zum Beispiel dem neueren Standard CSS2, zu greifen.

Frühe und häufige Tests ersparen Ärger und Arbeit

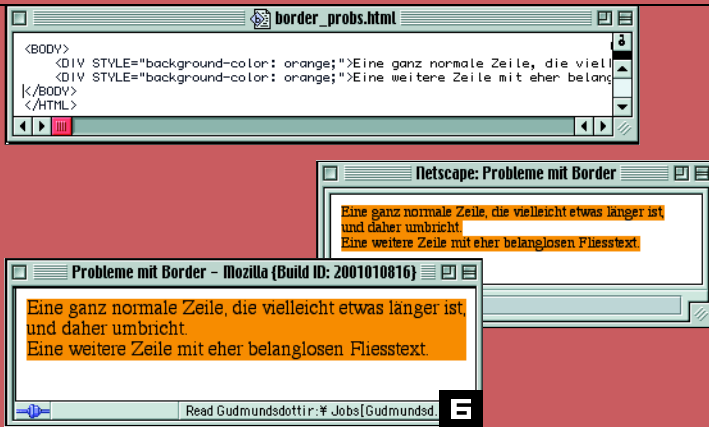
Im Webdesign gilt die Prämisse: „Test early, test often“. Das bedeutet im Klartext, dass Sie schon in einer frühen Produktionsphase beginnen sollten, ihre Seiten auf verschiedenen Browsern und Plattformen auf Fehler in Funktion und Darstellung zu untersuchen. Zwar werden Sie die Seiten kaum auf 31 verschiedenen Browser- und Betriebssystemkonfigurationen testen, wie es der Werbeanerwerblicher Doubleclick vorschlägt ³. Nichtsdestotrotz sollten Sie die Seiten gemäß ihrer vorgesehenen Zielgruppe auf den wichtigsten Browsern und Plattformen wie Windows, Mac OS oder Linux checken. Damit erkennen Sie Probleme frühzeitig und können diese mit wesentlich weniger Aufwand beheben als in der Endphase der Produktion.

Sowohl Golive als auch Dreamweaver verfügen über integrierte „Checker“ zum Überprüfen des HTML-Codes. Diese sind allerdings nur bedingt empfehlenswert, denn sie legen die Standards relativ weich aus und warnen nur unzureichend vor Fehlern wie etwa nicht korrekt geschlossenen HTML-Tags. Deshalb bieten sie keinen vollwertigen Ersatz für die Überprüfung des Codes im Browser. Für Dreamweaver gibt es die Erweiterung „Check Page for Accessibility“, die Sie gratis von Macromedias „Exchange“-Forum downloaden ⁴ und installieren können. Die Erweiterung liefert wesentlich mehr hilfreiche Hinweise als die Standardfunktion.

Noch weitaus ergiebiger als interne „Checker“ sind „Validatoren“ und „Lints“, die nicht nur die einzelnen HTML-Elemente überprüfen, sondern gleich das ganze Dokument anhand eines Regelwerks nach Fehlern durchforsten. Sie achten sozusagen nicht allein auf korrekte HTML-Rechtschreibung, sondern auch auf die HTML-Grammatik. Der bekannteste Lint ist HTML Tidy, der via W3C frei erhältlich ist. Es gibt ihn als eigenständiges Programm, als Zusatzmodul und integriert in diversen Applikationen auf verschiedenen Plattformen ⁵. Als offizieller Validator für Webseiten kann der W3C-HTML-Validator gelten, der in der Lage ist, online abgestellte Dokumente zu untersuchen.

Übrigens stellt das W3C zwei weitere nützliche Werkzeuge zur Verfügung: einen CSS-Checker und einen Checker, der sämtliche Links überprüft ⁶. Alternative Checker und Validatoren sind beispielsweise in Yahoo katalogisiert ⁷, eine weitere, exzellente Website zum Überprüfen von HTML-Dokumenten ist Bobby (<http://bobby.cast.org>). Neben den normalen HTML-Standards wird die zu prüfende Seite insbesondere auf „Accessibility“ geprüft, außerdem finden sich dort ausführliche Hinweise.

Natürlich ist der verlässlichste Test immer noch derjenige, den man unter realen Bedingungen mit echten Browsern und Plattformen durchführt. Auf der Website Evolt.org finden Sie eine breite Sammlung an neuen und alten Browsern ⁸, falls Sie auch die Site auch in einem älteren Browser der Version 3 testen möchten. Man sollte jedoch bedenken, dass es weder unter Windows noch unter Mac OS möglich ist, verschiedene Internet-Explorer-Versionen gleichzeitig zu testen, da sich die neueren Explorer-Systemdateien auch auf die Anzeige des alten Explorers auswirken.



E

Um in Netscape 4.x zu einem ähnlichen Resultat wie bei anderen Browsern (kein sichtbarer Zeilenabstand) zu kommen, müsste man hier einen negativen Wert angeben: „margin: -8px;“.

E

Netscape 4.x hat Probleme mit dem CSS-Element „background-color“.

6

W3C-HTML-Validator: <http://validator.w3.org>
 W3C-CSS-Validator: <http://jigsaw.w3.org/css-validator/>
 W3C-Link-Checker: <http://validator.w3.org/checklink>

7

[http://dir.yahoo.com/Computers_and_Internet/
 Data_Formats/HTML/Validation_and_Checkers/](http://dir.yahoo.com/Computers_and_Internet/Data_Formats/HTML/Validation_and_Checkers/)

8

Browser-Friedhof auf Evolt.org:
<http://browsers.evolt.org>

Netscape 6/Mozilla unterscheidet sich beim Testen von allen anderen Browsern, indem er auch die kleinsten Abweichungen von den W3C-Standards bestraft. Das macht insbesondere den „echten“ Mozilla-Browser, der anders als Netscape 6 immer über die aktuellsten Verbesserungen verfügt, zu einer harten Teststrecke für Ihre Webseiten.

Die Tücken der neuen Browser

Mit der kürzlich erschienenen neuen Browsergeneration entstehen dem Webdesigner auch einige neue Schwierigkeiten, die es zu meistern gilt. Folgende Besonderheiten sollten Sie beachten:

HTML-Dokument: Damit der Browser in der Lage ist, ein Dokument korrekt darzustellen, hat das W3C in die HTML-Spezifikationen eine Regel aufgenommen, derzufolge jedem Dokument eine „Doctype“-Angabe voranzustellen ist. Diese gibt Aufschluss über die benutzte Dokumentensprache und -version und verweist dabei auf eine „Document Type Definition“ (DTD) **E**. Der HTML-4.01-Standard sieht drei DTDs vor:

➔ **„HTML 4.01 Strict“** umfasst den „reinen“ HTML-4.01-Standard:

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

➔ **„HTML 4.01 Transitional“** bezieht die als „veraltet“, aber noch erlaubt geltenden Elemente mit ein:

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

➔ **„HTML 4.01 Frameset“** ist das DTD für HTML 4.01 Transitional in Dokumenten, die Frames benutzen:

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

Netscape 6/Mozilla und auch Explorer 5 unter Mac OS kennen zwei Darstellungsmodi: Das „Standardlayout“ und das „Quirk-Layout“. Ersteres ist eine standardkonforme Anzeige des HTML-Dokuments, während im zweiten Fall die von den alten Browsern ge-

wohnten „Macken“ nachgeahmt werden. Die Ansicht im Browser verdeutlicht einen Unterschied **E**: Der an das „Div“-Tag angehängte Stil „fließtext“ färbt den Text rot ein. Lediglich im „Standardlayout“ wird die Einfärbung auch an den Text innerhalb der im „Div“-Tag eingebetteten Tabelle weitervererbt („Inheritance“) und damit korrekt angezeigt. Wenn Sie für Ihre HTML-Dokumente Transitional-Doctypes verwenden, bietet dies den Vorteil, dass Sie einerseits veraltete Tags benutzen können und andererseits vom Standardlayout-Modus profitieren.

HTML-Code: Um „Missverständnisse“ beim Rendern der Seite im Browser zu vermeiden, sollten Sie darauf achten, bei allen HTML-Elementen, die über einen „End“-Tag verfügen, diese auch im Code einzusetzen. Ebenso wichtig ist eine korrekte Verschachtelung der HTML-Elemente. Dabei dürfen sich HTML-Elemente nicht wie im folgenden Beispiel überschneiden:

```
<DIV> <B> text </DIV> </B>.
```

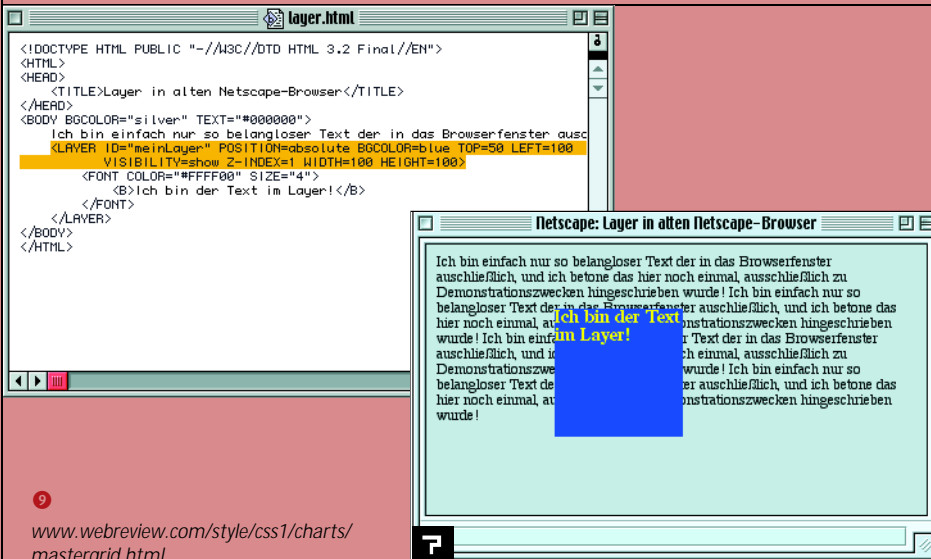
Korrekt arbeiten Sie, wenn Sie zuerst „innen“ das „“-Element mit „“ schließen und erst dann außen das „<Div>“-Element mit dem Tag „</Div>“:

```
<DIV> <B> text </B> </DIV>
```

Bei einer abweichenden Vorgehensweise würden Interpretationsschwierigkeiten in den Browsern auftreten **E**, und Sie würden in drei Browsern drei verschiedene Ergebnisse erhalten.

Interessante Optionen bieten die zahlreichen neuen Attribute in HTML 4, die über die HTML-Elemente Meta-Informationen liefern („Longdesc“, „Summary“, „Title“ etc.). Eine der praktischsten Neuerungen ist dabei das „Title“-Attribut, das man in diverse Elemente einbauen kann. Zudem zeigt es in den neueren Browsern bei Rollovern ein „Tooltip“ an, das man für detaillierte Informationen, etwa zu einem Bild oder einem Link, verwenden kann **E**.

Cascading Style Sheets (CSS): Viele Webdesigner werden sich erst durch die „hautenge“ Auslegung von Netscape 6/Mozilla bewusst, dass sie gegen die Standards des W3Cs verstoßen haben. Dabei steckt der Teufel häufig im Detail. Beispielsweise sollten ➔



Das „Layer“-Element lässt sich durch Positionierung eines „Div“-Containers mittels CSS ersetzen. Das funktioniert mit einigen Einschränkungen sogar in Netscape 4.x.

➔ Sie beachten, dass CSS-Code „case-sensitive“ ist, also ähnlich wie Javascript zwischen Groß- und Kleinschreibung unterscheidet. Erlaubt sind lediglich die Buchstaben „a“ bis „z“ in Groß- und Kleinschreibung sowie Ziffern und Bindestrich. Für den so genannten „Selector“, also den Class- oder ID-Bezeichner, ist der Unterstrich („underscore“) deshalb als Zeichen nicht erlaubt. Aus diesem Grunde sollten Sie sich bei der Bezeichnung von Stilvorlagen auf einfache Namen in Kleinschreibung (beispielsweise „fließtext“) beschränken:

```
<style type="text/css">
<!--
.fließtext { font-family: Arial, Helvetica, sans-serif; font-size: 12px;
font-weight: bold;
-->
</style>
```

Auch bei der Zuweisung des Stils sollten Sie unbedingt auf die korrekte Schreibweise im HTML-Code achten, zum Beispiel:

```
<p class="fließtext">Text </p>
```

Bei Abweichungen in der Anweisung und Zuordnung zeigt Netscape 6/Mozilla sonst den Stil schlichtweg nicht mehr an.

In der Übergangsphase zwischen den 4er-Browsern und der neuen Generation erweist sich die Verwendung von CSS teilweise als besonders schwierig. Insbesondere Netscape 4.x zeichnet sich durch eine äußerst mangelhafte CSS-Implementierung aus. So werden die mit „margin“ bezeichneten Abstände bei den HTML-Elementen nicht einfach auf den angegebenen Wert gesetzt, sondern zum normalen Abstand des HTML-Elements dazuaddiert . Bei Anwendung von „background-color“ auf Text wird nur der Hintergrund der Buchstaben eingefärbt, nicht aber die gesamte umgebende Box. Darüber hinaus muss „border: none“ hinzugefügt werden . Auch zur Positionierung von Seitenelementen sind Style Sheets derzeit noch nicht empfehlenswert: Aufgrund der mangelhaften Implementation kann man noch keine komplexeren Layouts ausschließlich auf CSS basieren lassen. Verwenden Sie deswegen als Layout-Hilfsmittel weiterhin besser HTML-Tabellen anstelle von Style Sheets. Über die Unterstützung der Browser von CSS-Elementen finden Sie in Eric Meyers „CSS-Mastergrid“ detaillierte Informationen.

Auch der neue Netscape 6/Mozilla-Browser bereitet, wenn auch aus entgegengesetzten Gründen, Kopfzerbrechen: Erstmals werden in den CSS2-Spezifikationen festgelegte Regeln zum Rendering angewandt, die sich auch ohne Verwendung von CSS auf normale HTML-Befehle auswirken. Das betrifft etwa die Höhe von Tabellenelementen oder die Breite von „border“ und führt oft zu unerwarteten Resultaten.

DHTML/Javascript/DOM: Die meisten Probleme in diesem Bereich sind durch die von Netscape 6/Mozilla schon lange angekündigte Nicht-Unterstützung des „Layer“-Elements in HTML und Javascript entstanden. Der „Layer“-Tag existiert nur für Navigator 4 und stellt eine Art Container dar. Der HTML-Code im „Layer“ wird wie eine Art Folie im Browserfenster angezeigt, die Sie pixelgenau positionieren und dynamisch ein- und ausblenden können. In standardkonformen Browsern lässt sich diese Funktionalität mit CSS erreichen. Zu diesem Zweck stecken Sie den Inhalt einfach in einen „Div“-Container und setzen die entsprechenden CSS-Attribute .

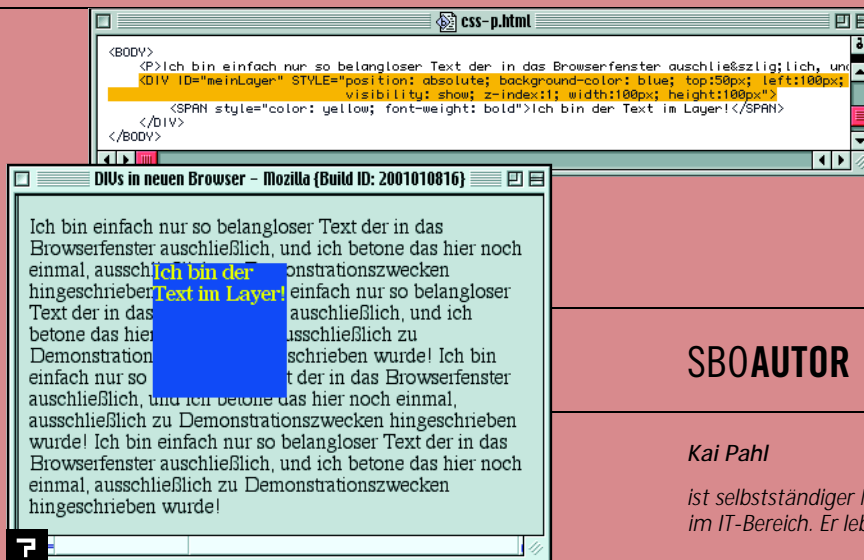
Des Weiteren sollten Sie beachten, dass der Zugriff auf einzelne DOM-Elemente, seien es HTML- oder CSS-Elemente, vereinheitlicht wurde. Den Zugriff mit „document.layers, document.tags, document.id“ etc. bewerkstelligen Sie nun mit „getElementById“ wie im folgenden Beispiel:

```
document.getElementById("meinDiv").style.color = "#ff0000";
```

Desillusioniert: Die neuen Browser sind ein Fortschritt, aber noch nicht am Ziel

Die „blühenden Landschaften“, die standardkonforme Browser versprechen, lassen auch mit der neuen Browsergeneration noch auf sich warten. Der Webdesigner wird wegen mangelhafter Implementation der Standards in Browsern und Webeditoren, aber auch wegen kaum bekannter Rendering-Vorschriften noch auf Jahre hinaus Bugfixes und Workarounds schreiben müssen und kann sich nicht darauf verlassen, dass die Standards von einem Tag auf den anderen in der Praxis angewendet werden können. Es geht Schritt für Schritt vorwärts, und je mehr Leute mitgehen, umso eher kann die Nutzer- und Webdesign-Gemeinde entsprechenden Druck auf die Softwarehersteller ausüben, um diese zur besseren Einhaltung der Standards zu bewegen.

Kai Pahl



SBOAUTOR

Kai Pahl

ist selbstständiger Multimedia-Producer und Autor im IT-Bereich. Er lebt in Hamburg.

Die neuen Browser im Überblick

10

www.webstandards.org

Netscape 6/Mozilla

Mozilla ist ein von der Open-Source-Gemeinde entwickelter Browser, den AOL zwecks Branding mit Netscape-typischen Features wie „Instant Messenger“ als Netscape 6.0 veröffentlichte. Beim „echten“ Mozilla-Browser rechnet man nicht vor Sommer dieses Jahres mit der Fertigstellung der 1.0-Version, doch erschien im Januar als neuer „Meilenstein“ bereits Mozilla 0.7.

Der Open-Source-Browser soll in der fertigen Version alle erdenklichen Webstandards unterstützen: Gegenwärtig sind HTML 4.0, CSS1, RDF (Ressource Description Framework) und DOM (Document Object Model) Level 1 vollständig, CSS2 und DOM Level 2 hingegen nur teilweise implementiert, XML unterstützt die aktuelle Version grundsätzlich vollständig. Für die Formatierung von Dokumenten bleibt momentan lediglich der Weg über XML und CSS1 (offizielle Bezeichnung: „XML+CSS“).

Internet Explorer 5/5.5 für Windows

Internet Explorer 5.5 hat Microsoft noch stärker in das hauseigene Windows-Betriebssystem zusammen mit eigenen HTML- und CSS-Erweiterungen (offizielle Bezeichnung: „CSS-Behaviors“, „HTML+TIME“) eingebunden, welche jedoch weder Cross-Browser-, noch Crossplattform-tauglich sind. Deshalb wurde an Microsofts lockerer Interpretation von Standards schon öfters Kritik geübt ¹⁰. Der Hersteller kommentiert dazu: „Standardkonformität ist einer von vielen Wegen, um das Internet zu einer reicheren Plattform für alle User zu machen.“

Zumindest haben die Entwickler die Implementierung von CSS verbessert: Bugs in der CSS1-Interpretation sind beseitigt – zum Beispiel Probleme bei der Anwendung der CSS-Elemente „border“, „padding“ und „margin“ mit einigen HTML-Elementen sowie bei der Unterstützung der Pseudo-Elemente „first-letter“ und „first-line“. Nach Aussagen von Microsoft unterstützt der Browser immer noch nicht den vollen Umfang des DOM Level 1 und CSS1. Die CSS1-Testsuite des W3C erfüllt Version 5 zu 58 Prozent und Version 5.5 zu 80 Prozent.

Im Gegensatz zu Netscape wählt Microsoft bei der Formatierung von XML-Dokumenten den Weg über XSL (Extensible Stylesheet Language). Dabei hat sich der Browserhersteller dem XSL-Standard zu einem Zeitpunkt verschrieben, als dieser noch nicht als solcher festgeschrieben war, und schob später Bugfixes nach. XML-Formatierungen auf CSS-Basis unterstützt der Browser nur unzulänglich.

Internet Explorer 5 für Mac OS

Trotz Namensgleichheit gibt es nicht nur optische Unterschiede zwischen der Mac-OS- und der Windows-Version von Internet Explorer. Der Mac-OS-Browser basiert im Gegensatz zu Windows auf einer Rendering-Engine namens „Tasman“, bei deren Entwicklung von Anfang an Standardkonformität das Ziel war. Dieses wurde auch beinahe vollständig umgesetzt: Der Browser unterstützt HTML 4 und CSS1 ganz, XML, CSS2 und DOM Level 1 jedoch nur teilweise.

Opera 5

Mit Opera 5 hat man bei Operasoftware (www.opera.com) in Norwegen Abschied vom Modell eines kostenpflichtigen Browsers genommen. Stattdessen veröffentlichte der Hersteller Opera 5 in einer „Adware“-Version mit permanent eingeblendetem Werbefbanner. Wer auf diese Einblendung lieber verzichten möchte, zahlt 39 Dollar für die werbefreie Version.

Verglichen mit den monströsen Marktführern aus den USA ist der Download des Browsers geradezu lächerlich: Wer kein Java braucht, kann sich mit dem 2 Megabyte schlanken Installer begnügen. Momentan ist Opera 5 für diverse Windows-Versionen erhältlich, Versionen für die anderen Betriebssysteme befinden sich noch in der Alpha- (Mac OS) oder Betaphase (Linux).

Opera 5 unterstützt HTML 3.2, CSS1 und XML auf Basis von CSS komplett, HTML 4.01, CSS2, XML+XSL und Javascript jedoch nur teilweise. Insbesondere DHTML-Anhänger bemängeln das Fehlen einiger DOM-Methoden wie „createElement“ oder „appendChild“.