

Andy Clarke
Molly E. Holzschlag

Transcending CSS



1

Entdeckung

Gestalten Sie den Webdesignprozess neu.

Verbessern Sie den Design-Workflow.

Lernen Sie anregende und spannende Möglichkeiten kennen.



Einführung in Transcendent CSS

Transcendent CSS ist nicht nur ein Appell, das neueste und coolste CSS einzusetzen. Es geht vielmehr um das Bestreben, die mit CSS gelernten Lektionen zu nutzen, um zum besten Künstler und Designer zu werden. Mit Transcendent CSS legen Sie Altes ab, wenden sich Neuem zu und lassen sich inspirieren, die Workflows des Webdesigns adäquater und lebendiger zu gestalten. CSS regt Sie an, immer mehr zu lernen – sowohl über gestalterische als auch technische Herausforderungen, denen Sie in Ihrer Arbeit begegnen.

Welche Tools brauchen Sie?

Welche Tools sind für Transcendent CSS erforderlich, um mit den in diesem Buch erläuterten Richtlinien zu arbeiten? Sie brauchen nichts anderes als die Dinge, mit denen Sie wahrscheinlich sowieso schon arbeiten – also keine neuen Tools und Geräte, keine neue Software und keine neue Serverkonfiguration.

Dieses Buch ist nicht für Anfänger im Webdesign gedacht. Ich gehe davon aus, dass Sie bereits über einige praktische Erfahrung mit XHTML-Markup (eXtensible HTML) und CSS verfügen und die zentralen Konzepte der Webstandards kennen. Wenn Sie noch mit Tabellenlayouts arbeiten, erfahren Sie in diesem Buch nicht die Grundlagen von Selektoren oder allgemeinen CSS-Eigenschaften. Zu diesen Themen gibt es eine Reihe ausgezeichnete Quellen.

Doch falls Sie sich gerade erst intensiv mit CSS beschäftigen, hoffe ich, dass Sie die Konzepte und Beispiele in diesem Buch inspirieren und dass Sie mutig alles aufgreifen, um möglichst viel über CSS zu lernen. Egal, wie lange Sie vielleicht schon mit CSS arbeiten – es gibt immer Neues zu entdecken und zu lernen.

Dieses Buch setzt auch voraus, dass Sie offen für Neues sind. Obwohl nicht alles hier Besprochene für Sie oder die Arbeit in Ihrer Agentur oder für Ihre Kunden hundertprozentig relevant sein wird, möchte ich Sie dazu ermuntern, neue Ideen aufzugreifen. Passen Sie

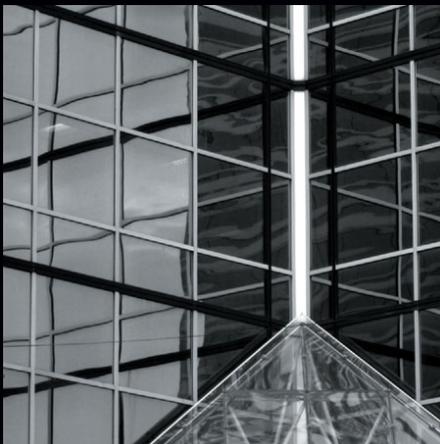
diese Ideen an Ihre Bedürfnisse an und finden Sie neue Verwendungen, von denen ich noch nicht mal geträumt habe. Vor allem aber wünsche ich mir für Sie, dass Sie sich wirklich für zukunftssträchtige Methoden interessieren und spannende und neuartige Arbeiten fürs Web schaffen wollen.

Warum Sie Transcendent CSS brauchen

Ich bin Designer und entwerfe für mein Leben gerne. Ich wünsche mir, dass ich eines Tages mal Sachen mit Kultstatus entwerfen kann, z.B. klassische Autos oder gar etwas wie den Apple iPod. Dinge, die von allen geliebt werden und mir bergeweise Geld einbringen ... damit ich mir so viele Roller und 1960er Minis kaufen kann, wie in meine Garage passen. Tja, leider habe ich gar keine Garage und obendrein macht es mir viel zu viel Spaß, Designer zu sein. Halten Sie mich für verrückt, aber ich liebe es einfach, fürs Web zu designen.

Das Web hat mir nicht immer Freude bereitet. In meiner Karriere hätte ich nach tagelangem Frust schon oft den Computer liebend gerne aus dem Fenster geworfen. Nachdem ich mich ewig mit irgendwelchen Problemen herumgeschlagen hatte, erschien mir die Vorstellung erstrebenswert, in einem Gewächshaus Gemüse zu züchten und dabei nur ein altes Radio als Gesellschaft zu haben. Dann machte ich mich doch wieder an die Arbeit und hatte nach nicht allzu langer Zeit das Problem gelöst. Meine Leidenschaft fürs Web war wieder da. Eigentlich schon komisch – ich hätte nie gedacht, dass Webdesign eine solche Herausforderung ist. Dabei ist es ja nicht wie das Ersteigen des Mount Everest – man entscheidet sich nicht gerade deswegen dafür, „weil das Web einfach da ist“.

Doch viele Phasen des Webdesignprozesses sind für visuell denkende Designer wie mich eine echte Herausforderung! Wir arbeiten täglich mit Tools wie Adobe Photoshop oder Adobe Fire-



works und schieben für unsere Designs Pixel über den Monitor. Die eher technischen Aspekte der Dinge, durch die Websites heutzutage funktionieren, vor allem das Schreiben von aussagekräftigem Markup und CSS, sind ungewohnt oder eben einfach wenig intuitiv.

CSS ist nicht designerfreundlich

Ein Grund dafür ist, dass CSS als Technologie zwar zur Lösung von Designproblemen gedacht war, aber selbst nicht sonderlich designerfreundlich ist – vor allem deswegen, weil es von Technikern statt von Designern für Designer geschaffen wurde.

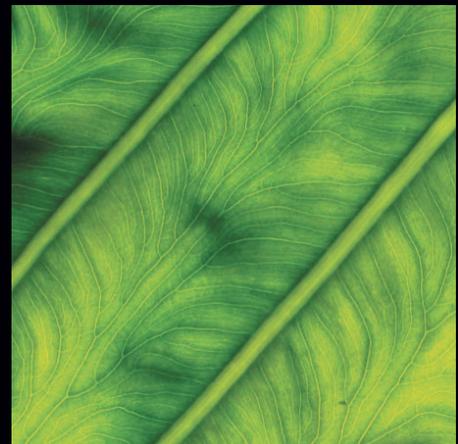
Obwohl die Grundprinzipien ganz einfach sind, beispielsweise

```
p { color : #000; }  
body { background-color : #fff; }
```

sind für manche Designer Begriffe wie *Kaskade* und *Vererbung* schwierig zu verstehen. Wenn es dann noch um *Positionierung*, *zusammenfallende Ränder* oder das *Boxmodell* geht, werden Sie verstehen, warum Designer relativ lange gebraucht haben, um CSS zu akzeptieren.

Nach seiner Einführung war CSS die meiste Zeit die Domäne der Techniker. Mit ihren trainierten grauen Zellen konnten sie solche Konzepte wie *Spezifität* oder die unzähligen, weitgehend unverständlichen CSS-Hacks besser begreifen, ohne die ein einigermaßen browserübergreifend einheitliches Design nicht zu bewerkstelligen war.

Diese Schwierigkeiten trugen wenig dazu bei, die Wissenskluft zu überbrücken, die schon immer zwischen visuellen Designern und technischen Entwicklern, die ihre Designs über Code implementieren, existierte. Designer waren von CSS oft einfach nur noch frustriert.



HTML-Editoren lassen Designer im Stich

Zwar konnten visuelle HTML-Editoren wie Adobe Dreamweaver im Laufe der Jahre ihr generiertes Markup und CSS deutlich verbessern, doch eines der wichtigsten Probleme für Webdesigner konnte noch nicht adäquat gelöst werden: Wie steht das gewünschte Design in Beziehung zu dem aussagekräftigen Markup, das dieses Design ermöglicht?

Heute ist es ein wesentlicher Bestandteil der Arbeit eines Webdesigners, die Grundlagen von aussagekräftigem Markup und CSS zu begreifen. Visuelle HTML-Editoren wie Dreamweaver müssen Designern dabei helfen, „hinter“ ihre Designs zu schauen und ihren Code „visualisieren“ zu können.

Die mangelnde Darstellung der Browser hemmt den Fortschritt

Und dann werden wir natürlich von den Browsern herausgefordert. Seit CSS zum ersten Mal im Microsoft Internet Explorer 3 (der erste etablierte Browser mit CSS-Unterstützung) implementiert wurde, ließ einen die Arbeit mit CSS oft verzweifeln. Browser-Bugs, Darstellungsfehler oder die schiere Stupidität mancher Browser kosteten uns alle häufig den letzten Nerv.

Die Situation verbesserte sich langsam – Netscape 4.x war besser als der IE 3 und der IE 5 für Mac zeigte zum ersten Mal, dass CSS „funktionieren“ konnte. Der IE 6 bot beim Erscheinen die beste CSS-Unterstützung aller Browser – trotz seiner wohlbekannten Sammlung von Bugs mit den ausgefallensten Namen.

Ich stelle mir meinen 14-jährigen Sohn vor, der gerade anfängt, sich mit Webdesign zu beschäftigen. Wie sehr wird er später darüber lachen, wenn er vom „Double-Margin Float“ oder dem „Peekaboo-Bug“ hört! Vielleicht fragt er sich auch, ob „3px Jog“ etwas war, was die Designer als Training vor dem Frühstück machten.

Unterschätze nie die Macht des Einzelnen

Und doch fand sich eine Lösung, vor allem durch die Entschlossenheit von Entwicklern wie Tantek Çelik, Todd Fahrner und Eric Meyer, deren Arbeit den Weg für solche Designer wie Douglas Bowman und Dave Shea freimachte. Sie konnten zeigen, dass die Verwendung von CSS nicht nur erstrebenswert ist, sondern auch praktische Realität sein kann. In all den Jahren, seit die erste CSS-Spezifikation erschienen war, setzten sich solche Leute weltweit engagiert mit fast allen Problemen auseinander, auf die Designer in der Arbeit mit CSS trafen, und sie fanden viele Workarounds.

Es sollte auch nicht vergessen werden, dass die Mitglieder des an der Basis entstandenen Web Standards Project – dazu gehören Rachel Andrew, Molly E. Holzschlag und Dori Smith – in Zusammenarbeit mit Browserherstellern und Softwareentwicklern und durch unermüdliche Arbeit in der Fortbildung wichtige Beiträge geleistet haben, um das Bewusstsein für Standards zu steigern.

Ohne diese Leute, die unabhängig in kleinen Gruppen arbeiteten, wäre CSS nie das leistungsfähige Webdesign-Tool geworden, das es heute ist.

Im Vergleich zu den Pioniertagen von CSS haben wir es heute relativ einfach. Viele neue Webdesigner werden sich nie mit Tabellenlayouts beschäftigen oder frustriert sein, wenn sie CSS-Layouts in heutzutage als alttümlich betrachteten Browsern wie Netscape 4 zum Laufen bringen müssen.

Einige Sites über Accessibility sind ausgesprochen hässlich, doch das Problem liegt bei den Designern dieser Sites und nicht bei der Accessibility, die keine visuellen Punktabzüge erfordert. Das Gleiche gilt für Webstandards, auch wenn das Look & Feel der W3C-Website wenig dazu geeignet scheint, Designer zum Erlernen von XML und CSS2 zu motivieren.

—JEFFREY ZELDMAN
Webdesign mit Webstandards, 2006

Erweiterte kreative Möglichkeiten

Da Browser bei der Unterstützung solcher Standards wie CSS nun einen gewissen Reifegrad erreicht haben, bieten sie uns eine stabilere Basis, auf der wir unsere Designs entwickeln können. Also wird es Zeit, in die Gänge zu kommen und loszulegen!

Die Vorgaben von Tabellenlayouts beschränken unsere Designs auf ein starres Raster und verstärken die Konventionen des typischen zwei- oder dreispaltigen Layouts, die immer noch auf zahllosen Websites zu finden sind. CSS bietet neue kreative Möglichkeiten durch Floats und verschiedene Formen der Positionierung. Sie können die Reihenfolge von überlappenden Elementen (Layering) durch z-Index festlegen und beliebige Elemente über das CSS-Boxmodell stylen. Diese kreativen Möglichkeiten gab es bei Tabellenlayouts nicht (**Abbildung 1.1**).

Designs – nicht nur für den Bildschirm

Chris Lilley, später Vorsitzender der CSS-Arbeitsgruppe des W3C (World Wide Web Consortium), stellte seine Prognose, was für den Rest dieses Jahrzehnts im Web im Wesentlichen geschehen würde, zwei Jahre vor Freigabe der ersten CSS-Spezifikation und ganze sechs Monate, bevor Netscape seinen ersten Browser veröffentlichte:

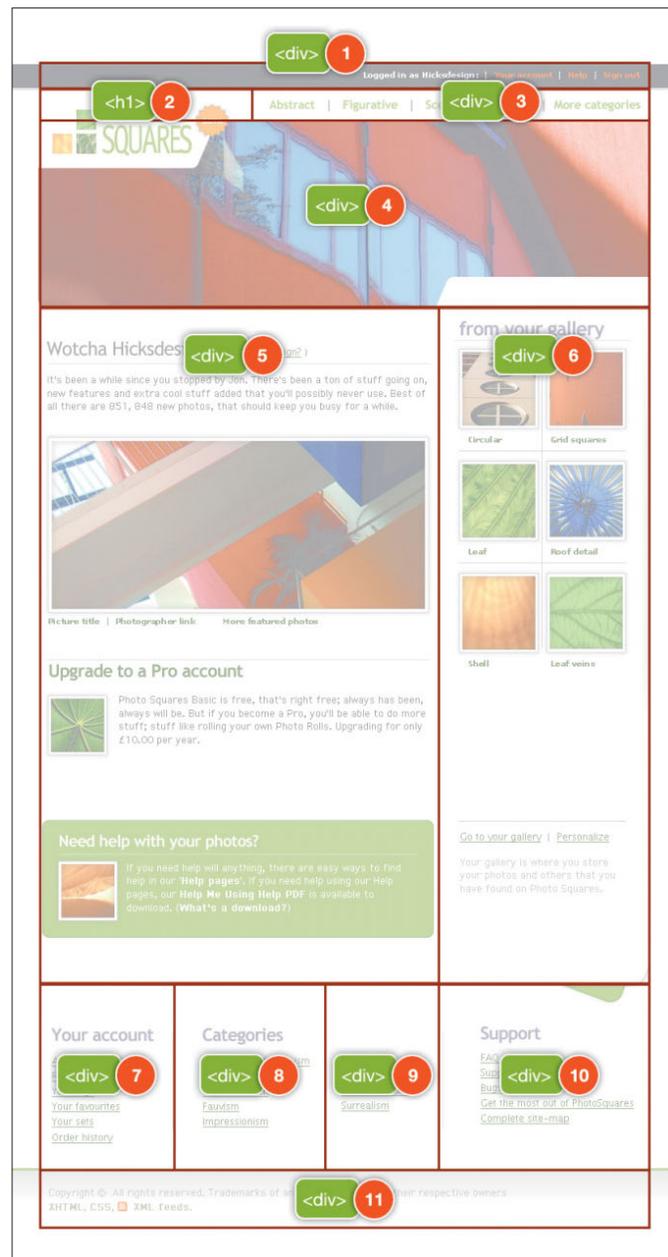
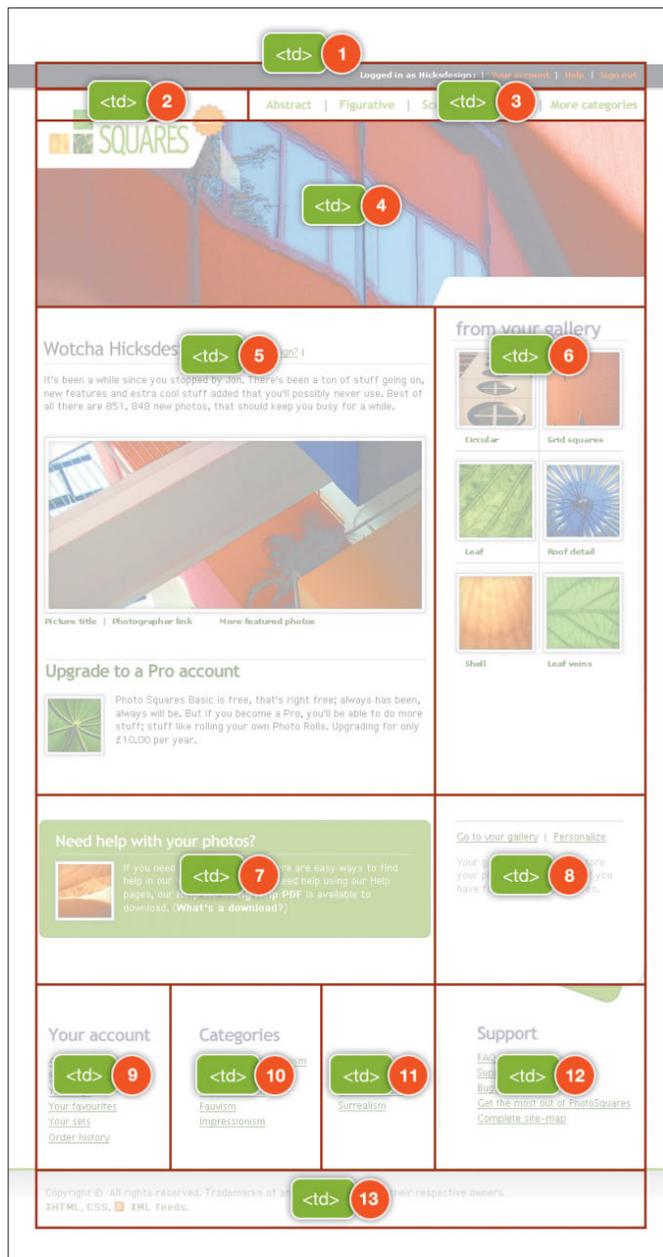
Wenn man HTML keine Stylesheets oder ähnliche Informationen hinzufügt, führt das zwangsläufig zu Dokumenten, die nur bei gewissen Browsern, bei einer bestimmten Fenstergröße, mit Standardschrift usw. gut aussehen.

- Chris Lilley (ehemaliger Vorsitzender der CSS-Arbeitsgruppe, damals mit dem forschen Namen Style and Formatting Properties Working Group), Mai 1994

Nach vielen Jahren harter Arbeit schafft das Web nun den Weg zu Handys, Spielekonsolen und Fernsehern; in Zukunft werden dazu alle Arten tragbarer Geräte gehören – auch solche, die heute noch gar nicht erfunden sind.

Stopp mal kurz den Download, ich fahr grad in einen Tunnel

Die Wahrheit ist, dass wir einfach nicht wissen, wo das Web als Nächstes auftauchen wird. In nicht allzu ferner Zeit wird mein Sohn über die Vorstellung lachen, dass das Internet mal an den Desktop-Computer gebunden war – so wie ich lachen muss, wenn ich an mein erstes Handy denke, dessen Batterie beinahe so groß wie ein Ziegelstein war und etwa auch so viel wog.



- 1.1** Der Wechsel zu CSS-basierten Layouts führt nicht immer zu strukturiertem oder geordnetem Code. Links: Die Darstellungsreihenfolge eines tabellenbasierten Layouts. Rechts: Die Tabellen wurden ohne inhaltliche Neuordnung durch <div>s ersetzt.

In einem sich so schnell ändernden Medium klingen Lilleys Worte jetzt vielleicht noch wahrer als damals, als er sie aufgeschrieben hat. Die heutigen Designer sollten sich zumindest darüber im Klaren sein, dass sich ihre Designs an die vielen Anforderungen dieser unterschiedlichen Umgebungen anpassen müssen. Schlanke, aussagekräftige Dokumente und CSS sind die Schlüsselemente beim erfolgreichen Übertragen eines Designs vom Computer auf andere Geräte, z.B. Drucker, Handheld-Computer, tragbare Abspielgeräte für Audio- und Videodateien oder Handys (**Abbildung 1.2**).

Accessibility ist kein Feature, sondern Design

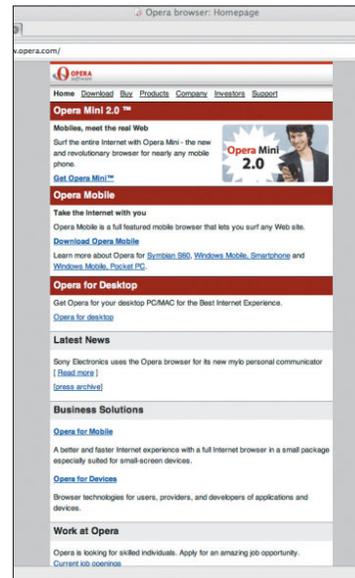
Gute Informationsarchitektur, Usability (Nutzbarkeit) und Accessibility (Barrierefreiheit) wurden zu Recht zu Aufgabengebieten für professionelle Designer und Entwickler. Vor allem ist es keine ethische Frage, sondern eine kommerzielle, die größtmögliche Accessibility zu gewährleisten. Jedoch beschränkten viele Designer, Entwickler und andere Spezialisten im Bereich der Accessibility die Diskussion fälschlicherweise einfach auf die Bedürfnisse für behinderte Menschen.

Accessibility ist eine Frage der Usability

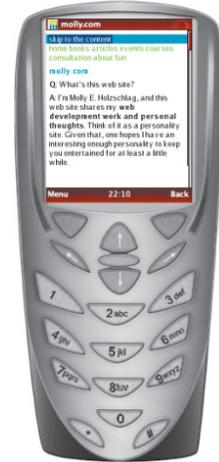
Viel von dem, was über Accessibility geschrieben wurde, zielte darauf ab, dass Websites für Blinde oder Sehbehinderte zugänglich sein müssen. Noch mehr wurde darüber geschrieben, wie Websites den Anforderungen von Accessibility-Richtlinien oder -Gesetzen genügen können. Allerdings geht das Meiste davon einfach am Wesentlichen vorbei.

Obwohl auf die Bedürfnisse behinderter Menschen geachtet werden muss, wurde das viel umfassendere Thema – dass Accessibility eine Frage der Usability ist – viel seltener angesprochen. Als professionelle Designer sollten wir unsere Inhalte insgesamt zugänglich gestalten. Im Rahmen unserer Möglichkeiten sollten wir den Ansprüchen von möglichst vielen Menschen gerecht werden – egal welche Fähigkeiten oder Einschränkungen diese Menschen haben oder mit welcher Geräteart sie ins Internet gehen.

Für den Geschäftsreisenden, der sich mit seinem tragbaren Computer ins Firmen-Intranet einloggen will, um Verkaufsberichte zu checken, ist Usability und Accessibility ebenso relevant wie für viele Kino-Websites, auf denen man zusätzliche Extras fürs Handy herunterladen kann, die aber selbst nicht mit dem Handy zugänglich sind. Durch gutes Design lassen sich viele Zugangsbeschränkungen für möglichst viele Menschen entfernen – darum geht es letzten Endes!



Opera Mini™ simulator



1.2

Überall wird CSS eingesetzt.

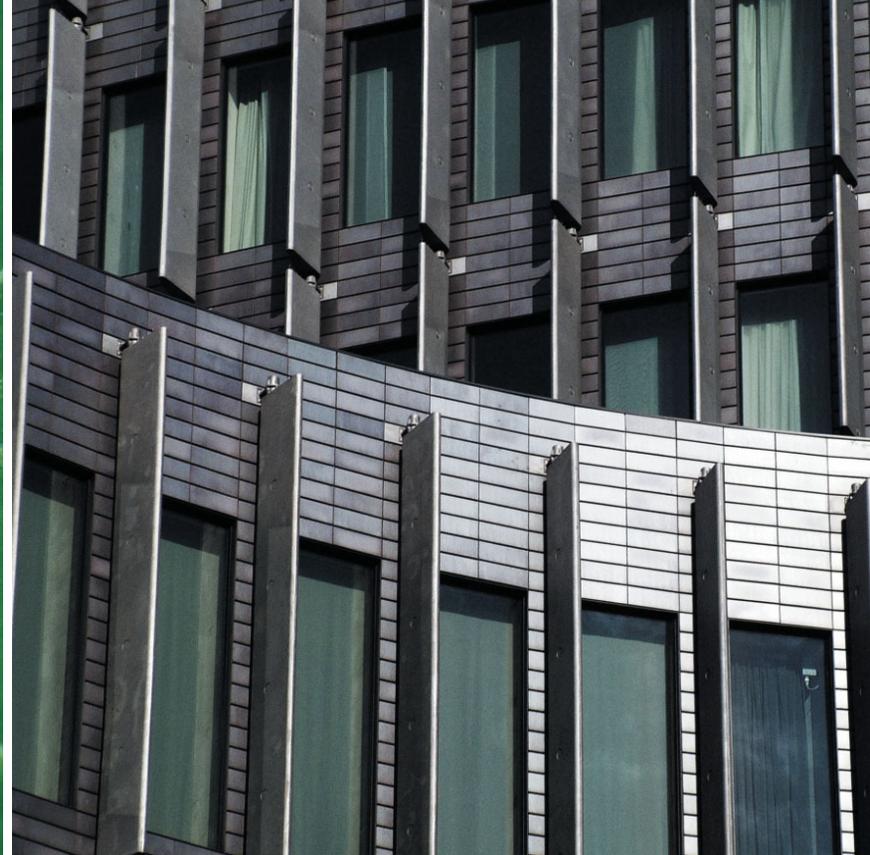
Lippenbekenntnisse reichen heutzutage nicht mehr aus

Leider betrachten viele Designer Accessibility immer noch als Einschränkung ihrer Kreativität, also eher als Richtlinien oder Gesetze, die befolgt werden müssen, anstatt als das „tägliche Brot“ eines Designers. Accessibility wurde viel zu oft als äußerer Faktor betrachtet und bei Websites erst nach der Fertigstellung getestet, um zu gewährleisten, dass sie bestimmten Standards genügen:

Wenn man Accessibility als Feature betrachtet, wird man sie wahrscheinlich weglassen. Die meisten Entwickler lassen sie sowieso weg, weil sie gar keine Ahnung von Accessibility haben oder einfach nicht wissen, dass sie überhaupt wichtig ist.

—Joe Clark (<http://joeclark.org/ice/iceweb2006-notes.html>)

Traurigerweise passiert ein Großteil der Arbeit, die für die Zugänglichkeit von Inhalten und Diensten aufgewendet wird, beim Design- und Entwicklungsprozess bestenfalls spät und schlimmstenfalls erst im Nachhinein, was Nachbesserungen erforderlich macht. Sollen Firmen sich entweder für das modernste spannende Ajax-Interface oder für Accessibility entscheiden, werden viele das dynamische Interface vorziehen und Accessibility-Tests oder -Features auf spätere Releases verschieben.



Wie Sie Ihren Kunden Transcendent CSS erklären

Viele Designer brennen eigentlich darauf, die Leistungsfähigkeit von CSS in ihrer Arbeit zu nutzen, halten sich aber zurück in der Annahme, dass ihre Kunden vielleicht erwarten, dass ihre Designs in allen Browsern gleich aussehen sollen.

Es stimmt, dass viele Kunden, Firmen und Auftraggeber vom Designer eine browserübergreifende Kompatibilität erwarten. Obwohl es zwar nicht die Aufgabe eines Designers sein sollte, einen Kunden darüber zu informieren, können ein paar einfache Metaphern dabei behilflich sein, Ihren Kunden die grundlegenden Konzepte des Transcendent CSS zu erläutern.

In vielen anderen Bereichen von Design und Technologie ist die Vorstellung lächerlich, dass es eine vom Alter oder der Tauglichkeit einer Technologie unabhängige Funktionalität oder Bedienungsweise eines Produkts geben soll. Die Konsumenten erwarten nicht nur, dass Technologien sich dauernd verbessern, sondern wollen auch das Gefühl haben, dass sie das allerneueste Produkt gekauft haben.

Sie können einem Kunden leicht erklären, dass Sie mit den neuesten Programmier-Techniken für die meisten modernen Browser arbeiten, doch dass auch Leute mit altmodischen Browsern immer noch ein gutes Ergebnis bekommen.

HIGH-DEFINITION TV

Hoch auflösendes Fernsehen (HDTV – High Definition Television) bietet den Nutzern von entsprechenden Fernsehern und Receivern eine wesentlich bessere Bildauflösung und Klangqualität als konventionelle Geräte. Hoch auflösendes Breitbandfernsehen steht in den USA schon viel länger zur Verfügung als in Deutschland. Seitdem die ersten Satellitenrundfunksender ankündigten, dass sie im Jahre 2004 in Deutschland mit hoch auflösenden Sendungen beginnen werden, haben sich Fachhandel und die TV-Gerätehersteller abgemüht, auf den „HD-ready“-Zug aufzuspringen.

Wenn Sie sich keine für HDTV ausgelegten Geräte oder keinen Anbieter für hochauflösendes Fernsehen zulegen, werden Sie nicht ausgesperrt, wenn Sie Ihre Lieblingsserie oder die Bundesliga schauen wollen. Doch weil Ihre Hardware nicht aktuell ist, bekommen Sie einfach ein etwas schlechteres, aber dennoch völlig akzeptables Bild.

DER IPOD-FAKTOR

Als Apple zum ersten Mal sein portables Musikabspielgerät iPod vorstellte, bot der Player weniger Features und war anders zu bedienen als die heutigen Modelle. In einer hart umkämpften Branche hat Apple seine Führungsposition bewahrt, indem bei fast jedem neuen Modell neue Features eingeführt wurden, dass man z.B. Albumcover oder Musikvideos abspeichern und abspielen konnte, die vom iTunes Store heruntergeladen wurden.

Ältere Versionen des iPod bieten immer noch die gleiche Funktionalität wie beim ersten Erscheinen. Kein Kunde hätte erwartet, dass ein älterer Player die gleichen Eigenschaften hat wie ein neueres Modell, und sie akzeptieren, dass sie sich ein neues Modell zulegen müssen, um die neuen Features nutzen zu können.

REGELMÄSSIGE UPGRADES FÜR IHRE SOFTWARE

Die Softwareindustrie ist bekannt dafür, dass sie fehlerhafte, instabile Softwareversionen und Betriebssysteme produziert, die sich regelmäßig aufhängen oder für Abstürze sorgen. Doch trotz dieses Mangels akzeptieren die Kunden, dass eine neuere, bessere Version gerade bevorsteht, und stellen sich auf Upgrades ein.

Die Kunden beauftragen einerseits die Designer damit, Websites zu entwickeln, sind aber andererseits auch Konsumenten, die praktisch täglich sich ändernden Technologien ausgesetzt sind. Also können sie nachvollziehen, dass bei modernen Browsern eine bessere Bedienbarkeit oder Gestaltung zu erwarten ist.

Anmerkung

Im Juni 2005 änderte ich meine Meinung, dass von der Regierung erlassene Gesetze und Regulierungen kontraproduktiv sind, um im Internet Menschen mit Behinderungen gerechter zu werden. Mein Artikel „Accessibility and a society of control“ (Accessibility und eine Gesellschaft der Kontrolle) führte zu vielen interessanten Rückmeldungen, wie man das Thema Web Accessibility weiterentwickeln kann (www.stuffandnonsense.co.uk/archives/accessibility_and_a_society_of_control.html).

Wenn Sie sich jedoch am Inhalt orientieren und dafür sauber strukturiertes und aussagekräftiges Markup nutzen, um für Struktur zu sorgen, und für die visuelle Gestaltung CSS nehmen, können Sie Accessibility gleich von Anfang an in den Designprozess integrieren. Davon profitieren alle – die Designer, die weniger mit Accessibility-Problemen kämpfen müssen, die Klienten, die Geld sparen, weil sie nicht mehr nachrüsten müssen und zufriedener Kunden haben, und vor allem die Besucher dieser Sites, weil sie nun leichter mit der Site umgehen können.

Der Weg zu Transcendent CSS

Verglichen mit den Freiheiten, die Druck- und Multimediadesigner genießen, hatten diejenigen von uns, die sich für die Arbeit mit Markup und CSS entschieden haben, stets unter Faktoren zu leiden, die unsere Designs einschränkten und oft außerhalb unseres Einflussbereichs lagen.

Seit dem Erscheinen der ersten CSS-Spezifikation wird unser kreatives Potenzial von der begrenzten Leistungsfähigkeit der Browser ausgebremst – außer wir lassen uns darauf ein, die Reinheit unserer Dokumente durch präsentationsbezogenes Markup zu opfern oder ganz in Flash zu arbeiten.

Die Browserhersteller verbesserten die Unterstützung für Webstandards weitgehend, vor allem bei CSS und dem W3C Document Object Model (DOM). Doch weil Microsoft sich dazu entschieden hatte, den IE ewig bei der Version 6 stagnieren zu lassen, sind Designer bei der Nutzung erweiterter Perspektiven von CSS eher zögerlich gewesen. Erschwerend kam noch hinzu, dass die älteren Versionen des IE immer noch weit verbreitet waren. Zum Glück werden sie immer seltener verwendet.

Als konkurrierende Browser wie Mozilla Firefox, Mozilla Camino, Apple Safari und Opera damit begannen, dem Marktanteil des IE Prozente abzugraben (**Tabelle 1.1**), erforschten vorausschauende Designer Wege, um die Nutzer dieser modernen Browser durch besonders ausgefeilte Designs zu belohnen. Diese Technik wurde unter dem Namen *Progressive Enhancement* oder *MOSe* bekannt.

Tabelle 1.1 Browser-Nutzung von Juli 2004–Juli 2006

	Internet Explorer 7 (beta)	Internet Explorer 6	Internet Explorer 5	Firefox	Mozilla	Netscape	Opera
2006	1,9%	57,8%	4,2%	25,0%	2,2%	0,4%	1,4%
2005	–	67,9%	5,9%	19,8%	2,6%	0,5%	1,2%
2004	–	67,2%	13,2%	–	12,6%	1,8%	2,0%

Quelle: W3 Schools (www.w3schools.com/browsers/browsers_stats.asp), August 2006

MOSe

Im Jahre 2003 stellte Dave Shea, der kanadische Designer, Autor und Schöpfer des CSS Zen Garden, einen überzeugenden Ansatz vor, um das Problem des Designs für die unterschiedlichen Fähigkeiten konkurrierender Browser anzugehen. Er nannte diesen Ansatz *MOSe* (Mozilla/Opera/Safari enhancement [Erweiterung]).

Laut Dave Shea könnten Designer durch MOSe den mangelnden Support des IE 6 für bestimmte CSS-Selektoren (z.B. Kind-, Attribut- und Adjacent Sibling-Selektoren, also Selektoren für benachbarte Elemente) ausnutzen. So sollten die Nutzer moderner anderer Browser ein erweitertes Design bekommen, das über ein für alle sichtbares Standarddesign gelegt wurde.

Theoretisch war Sheas Vorschlag nicht komplizierter als die `@import` at-Regel. Diese weit verbreitete Technik machte sich zunutze, dass Netscape 4 `@import` nicht interpretieren konnte, und bewahrte den alternden Browser vor CSS, das er entweder nicht verstand oder verpfuschte. Doch diese Technik weist einen wichtigen Unterschied auf. Als Shea seinen Artikel schrieb, war der IE 6 trotz seiner vielen Fehler ein Browser, der den Großteil der CSS-Spezifikation unterstützte. Es war der am häufigsten verwendete Browser auf dem Markt; also musste jeder Versuch, etwas an ihm vorbei zu gestalten, sehr subtil vorgenommen werden.

Workaround für Netscape 4.x

Netscape 4.x besaß eine gewisse CSS-Unterstützung. Um CSS-Stylesheets zu verdecken, die Netscape 4.x nicht interpretieren konnte, entschieden sich viele Designer dafür, durch die `@import` at-Regel alle außer den ganz grundlegenden Styles zu verstecken.

Dies ist ein grundlegendes Stylesheet, das alle Browser einschließlich Netscape 4.x sehen können:

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

Dies ist ein moderneres Stylesheet für Browser mit besserem CSS-Support:

```
@import "modern.css";
```

Alle Browser können `simple.css` laden, doch nur Browser, die die `@import` at-Regel verstehen, werden `modern.css` laden. Weil `modern.css` nach `simple.css` geladen wird, überschreiben dessen Regeln die in `simple.css`, falls dessen Regeln nicht spezifischer sind.

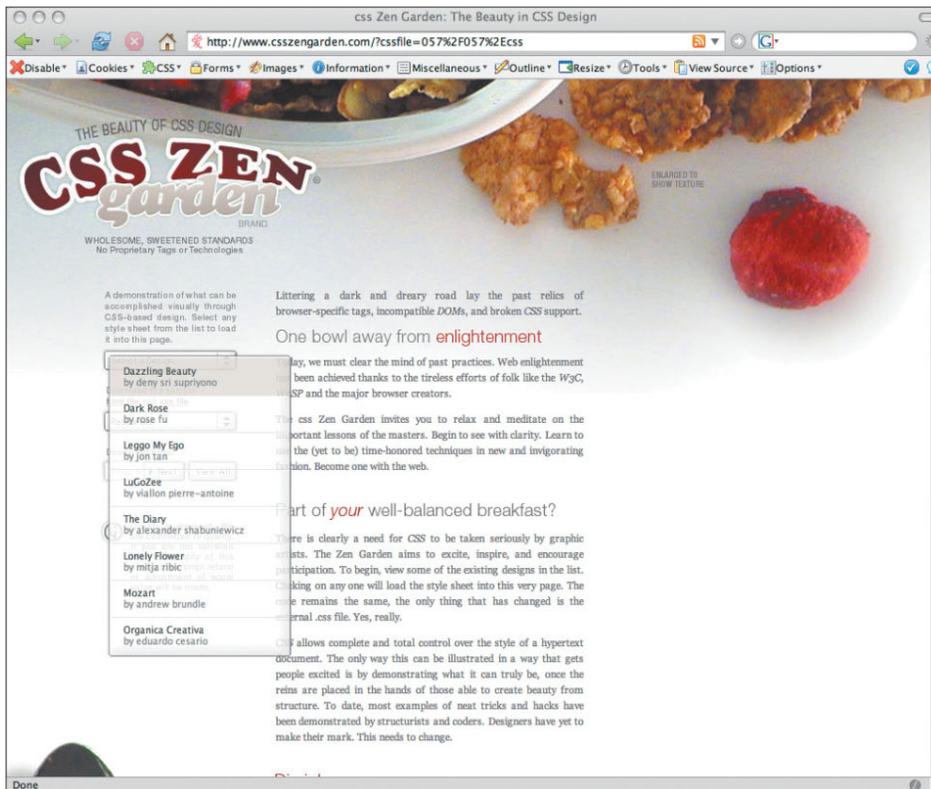
Der Schlüssel zur MOSe-Methode erinnert an die Entwicklung des Seitendesigns des NN4 (Netscape Navigator 4.x), als die Verbreitung von CSS zunahm. Nachdem man eine grundlegende, funktionierende Seite im IE erstellt hat, kann man mit diesen Selektoren zusätzliche Funktionalitäten hinzufügen.

—DAVE SHEA

www.mezzoblue.com/archives/2003/06/25/mose/

AUS DEM CSS ZEN GARDEN

Shaun Inman ging bei *This is Cereal*, seinem Design für den CSS Zen Garden, vom MOSe-Ansatz aus. Er nutzte die nicht vom Internet Explorer unterstützten CSS-Selektoren, um Surfern mit modernen Browsern eine bessere Darstellung zu ermöglichen. Shaun Inman verwandelte unsortierte Linklisten in raffinierte Drop-down-Menüs mit alphanumerischen PNG-Bildern. Obwohl er nicht als erster Designer diese Technik verwendete, war seine Arbeit sehr einflussreich. Wer mit älteren Browsern auf diese Seite kommt, sieht gestylte, allerdings einfachere unsortierte Listen und merkt nicht, dass es noch eine alternative Version gibt. Solche dezenten MOSe-Erweiterungen werden seitdem in vielen Designs für den CSS Zen Garden benutzt (**Abbildung. 1.3**).



1.3 Beispiele für MOSe im CSS Zen Garden.



„AND ALL THAT MALARKEY“

Angeregt durch Inman entwarf ich für meine eigene Website „And All That Malarkey“ zwei Designs: ein vollfarbiges, von der britischen Mod(ernisten)-Musikbewegung der 1960er inspiriertes Design und ein 20ld-Design in Schwarzweiß, das von der nüchternen Bildgestaltung des britischen Ska-Labels 2Tone inspiriert wurde.

Weil Ska vor der Mod-Musik kam, habe ich das Ska-Design nur für ältere Versionen des Internet Explorer verfügbar gemacht. Ich implementierte das Mod-Design durch CSS2.1-Selektoren, die nur von mod(ernen) Browsern verstanden wurde, und das war eine unterhaltsame Art und Weise, um die unterschiedlichen Fähigkeiten der Browser zu verdeutlichen.

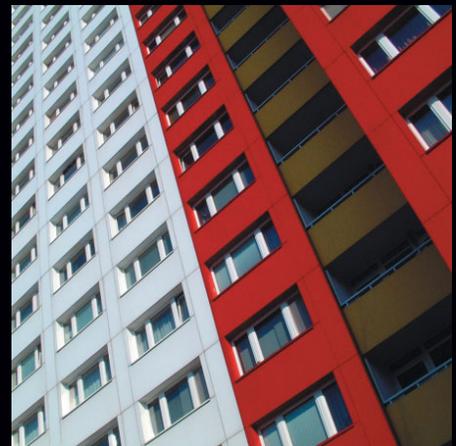
Anmerkung: Meine Entscheidung, die Besucher mit dem IE zu „bestrafen“, indem ich ihnen das komplette Mod-Design vorenthielt, traf nicht überall auf Zustimmung. Sie erfahren alles unter www.stuffandnonsense.co.uk/archives/and_all_that_design_malarkey.html.

Progressive Enhancement

Einige Monate nach Erscheinen von Sheas MOSe-Artikel wurde im Triangle TechJournal ein weiterer Beitrag veröffentlicht, der das Konzept des Progressive Enhancement (stufenweise Erweiterung der Darstellung je nach Fähigkeiten des Browsers) eingehender erklärte:

Progressive Enhancement stellt ein praktikables Vorgehen dar, bei dem die Übertragung von Informationen nach Bedarf der User aktiviert wird. Gleichzeitig wird damit Accessibility, eine zukünftige Kompatibilität und die Bestimmung der User Experience, welche auf den Fähigkeiten neuer Geräte beruht, möglich.

—Debra Chandra und Steve Champeon (http://hesketh.com/publications/progressive_enhancement_paving_way_for_future.html)



Doch bis heute ist es für Designer schwer, in kommerziellen Projekten die Methoden des Progressive Enhancement komplett zu übernehmen. Diese Schwierigkeit wurzelt nicht nur im dominierenden Marktanteil des Internet Explorer 6, sondern auch in der Überzeugung, dass es korrekt sei, einen Design-Fixpunkt festzulegen, der auf dem populärsten Browser basiert, auch wenn dieser Browser in puncto Unterstützung moderner Standards nicht so leistungsfähig und fortschrittlich ist wie seine Konkurrenten.

Für ein so junges und dynamisches Medium wie das Web erscheint es alles andere als fortschrittlich, wenn Designer nur wegen eines Browsers die Grenzen des Designs nicht ausreizen sollten – auch wenn es sich um den Marktführer handelt.

Sowohl MOSe als auch Progressive Enhancement waren Ideen, mit denen Designer ermutigt werden sollten, für Browser, die diese Selektoren und Regeln unterstützten, alle in der CSS2.1-Spezifikation zur Verfügung stehenden Tools einzusetzen. Auch heute, einige Jahre später, werden diese CSS-Regeln vielfach als *fortschrittlich* betrachtet, ungeachtet der Tatsache, dass sie nur ein paar Jahre nach der Geburt des kommerziellen Internets erfunden wurden.

MOSe und Progressive Enhancement wurden in Designer-Portfolios und Blogs eingesetzt, nur selten findet man sie in kommerziellen Projekten des Mainstream. Als Folge davon ist das progressive Webdesign größtenteils stecken geblieben und wird uninteressant werden, wenn nicht bald etwas passiert.

Meine Frage lautet: Kann man Progressive Enhancement drei Jahre nach Prägung des Begriffs immer noch *progressiv* nennen? Leider nein, also wird es Zeit, für Weiterentwicklung zu sorgen.

