
dpi
Bettina K. Lechner
Bernhard Stockmann
CSS PUR!
Ultimative Weblösungen mit Stil
ADDISON-WESLEY
CD

Einleitung

Vorwort

"... auch wenn das Look&Feel der W3C-Website wenig dazu geeignet scheint, Designer und Designerinnen zum Erlernen von XML und CSS zu motivieren."

Jeffrey Zeldman, 2006

Das Web als grenzenloser Raum lädt immer mehr Menschen ein, sich nicht nur mit der Rolle als LeserIn zu begnügen, also nur Information abzurufen, sondern öfter und vermehrt direkt einzugreifen, um Gedanken, Ideen, Bilder, Videos etc. zu veröffentlichen, sich zu vernetzen und an Communities teilzuhaben. Will man als aktiver Webnutzer nicht von Halbfertiglösungen bzw. fertigen Templates – also grafischen Vorlagen – abhängig sein, kommt man um Webdesign und damit (X)HTML und CSS nicht herum.

Die Webprogrammierung in (X)HTML in Verbindung mit CSS bietet so viel Freiraum wie noch nie zuvor in der Webgestaltung. Die Befreiung von den Fesseln des Tabellenlayoutens hat großartige Designs ermöglicht, die nicht nur gut aussehen, sondern den Webstandards entsprechen und damit einen Beitrag zur besseren Accessibility (Barrierefreiheit) und Usability (Zugänglichkeit) leisten. Und genau darin sehen wir den Sinn und Auftrag dieses Buchs: die Verbreitung der gängigsten und wichtigsten Webstandards – verständlich aufbereitet und leicht nachzuvollziehen. Mit diesem Wissen werden Sie imstande sein, eigene kreative Ideen umzusetzen und auch weiterzugeben.

An wen richtet sich dieses Buch?

Das Buch ist für Sie genau richtig, wenn Sie ...

- das Programmieren von Websites mit CSS erlernen möchten und HTML beherrschen.
- Ihre Websites bisher mit Tabellen oder Frames strukturiert haben und nun Ihre Kenntnisse auf den neuesten Stand bringen wollen.
- Templates (Website-Vorlagen, z.B. für CMS wie Joomla!, Typo3, Websitebaker etc.) verändern oder erstellen möchten.
- mit CSS bereits Ihre ersten Schritte gewagt haben und an Ihre Grenzen gestoßen sind.
- Ihr CSS-Know-how vertiefen und erweitern wollen.
- CSS unterrichten und Anregungen und Beispiele für den Unterricht benötigen.

Inhaltlicher Aufbau

Das vorliegende Buch lässt sich in vier große Bereiche einteilen:

1 Im ersten Kapitel **Grundlegendes** geht es um die Erläuterungen der Grundprinzipien von CSS. Sie erfahren, welche Vorteile es bringt, sich mit CSS auseinanderzusetzen, und warum Accessibility & Usability nicht nur behinderten Menschen nützen. Wir blättern in den wichtigsten Referenzen – also Websites, wo Sie nachschlagen können. Wir sehen uns Prüfwerkzeuge an, mit deren Hilfe Sie Ihre Werke validieren lassen, und wir statten Ihren Computer mit den wichtigsten und gängigsten Programmen und Erweiterungen für die Webprogrammierung aus.

2 Im zweiten Teil steigen wir in die **Basics** von CSS ein. Sie erlernen Schritt für Schritt, wie Sie eine HTML-Datei korrekt deklarieren und welche Möglichkeiten es gibt, CSS in HTML einzubinden. Wir gehen auf die einzelnen Selektoren ein, besprechen den Seitenaufbau und Sie lernen das Box-Modell, den Unterschied zwischen Block- und Inline-Elementen und erfahren, was Sie bei der Darstellung der Websites in den verschiedenen Browsern berücksichtigen und wissen müssen. Als Vorbereitung für den Praxisteil 3 normalisieren wir alle Stile, so dass wir von gleichen Abständen ausgehen.

3 Der große dritte Abschnitt erklärt anhand von **drei Praxisbeispielen (Prototypen)** alle wichtigen CSS-Elemente. Wir erstellen eine einfache Visitenkarten-Webseite, eine umfangreichere Business-Webseite und sogar ein mächtiges Webportal. Diese drei Prototypen enthalten alle CSS-Elemente, die zur Umsetzung nötig sind, werden detailliert erklärt und sind für Sie in genauen Schritt-für-Schritt-Anleitungen nachvollziehbar. Anhand dieser drei Beispielwebseiten führen wir Sie durch einen vollständigen Programmierworkflow und jedes Kapitel ist mit Tipps & Tricks gespickt, die die Autoren aus ihrer alltäglichen Programmierpraxis verraten. Auch auf die Vor- und Nachteile von YAML gehen wir ein.

4 Im letzten Teil 4 befassen wir uns mit **Ausblicken** und Visionen auf dem weiten CSS-Feld. In einer **Linkliste** fassen wir alle in dem Buch vorhandenen Links zusammen. Und natürlich finden Sie hier den Index zur Schlagwortsuche.

Dieses Buch bringt Ihnen die revolutionären Ideen von CSS näher – denn es geht um weit mehr als nur das simple Ersetzen von Tabellen durch DIVs.

Die Autoren

"Anyone who slaps a 'this page is best viewed with Browser X' label on a Web page appears to be yearning for the bad old days, before the Web, when you had very little chance of reading a document written on another computer, another word processor, or another network."

Tim Berners-Lee, Erfinder des World Wide Web und Direktor des W3C

Browsersupport

Die Zeile „optimiert für Browser XY" auf einer Website ist symptomatisch für den Grad der Verzweiflung eines Designers: Die Anpassung eines Webdesigns an die zahlreichen unterschiedlichen Browser auf den verschiedenen Plattformen ist eine unserer großen Herausforderungen und wie oft und gerne hätte ich eine Rechnung für die vielen Entwicklungsstunden an so manchen Browser-Hersteller geschickt. Selbstverständlich ist es unmöglich, einen bestimmten Browser zu ignorieren – als WebdesignerIn ist es Ihre Aufgabe, allen BesucherInnen zumindest einen Zugang zu den Inhalten zu bieten.

Im Jahre 2006 veröffentlichte Nate Koechley einen Artikel, in dem er den „stufenweisen Browsersupport" (graded browser support) propagierte. Er dehnte darin den Begriff Support weiter aus und definierte drei Grade von Browser-Unterstützung:

- **A-Grad**
  Browser, die in dieser Kategorie gelistet sind, müssen voll unterstützt werden, sie sind identifizierbar, leistungsfähig, modern und verbreitet. Zirka 96% aller Yahoo!-UserInnen verwenden einen dieser Kategorie zugeordneten Browser. Fehlern wird daher hohe Aufmerksamkeit geschenkt.
- **C-Grad**
  Basisunterstützung. Browser dieser Kategorie sind zwar identifiziert, aber unzulänglich, antiquiert oder selten. Verbreitung: ca. 3%. Es wird auf eine fehlerfreie Darstellung von wesentlichen Inhalten und Funktionen geachtet, jedoch ohne Ausgestaltung. Der Browser erhält nur reines, semantisches HTML.
- **X-Grad**
  Unbekannte Browser-Typen, leistungsfähig, modern und selten. Fehler sind weitgehend unbekannt, daher kann nicht auf sie eingegangen werden. In diese Kategorie fallen meist ganz neue Browser, die im Gegensatz zum C-Grad volles Design zur Verfügung gestellt bekommen.

Diese Einteilung stellt eine Möglichkeit dar, um eine Entscheidung für die Wertigkeit der Browser-Unterstützung zu treffen. Denn bei der Fülle an Varianten und Versionen von Betriebssystemen, inkl./exkl. Service-Updates, und Browser-Versionen ist es zeitlich fast undenkbar, wirklich alle durchzutesten.

Einteilung der Browser nach Yahoo!, Stand Juli 2009

	Windows 2000	Windows XP	Windows Vista	Apple OS X 10.4	Apple OS X 10.5
Firefox 3.X		A-Grad	A-Grad		A-Grad
Firefox 2.X		A-Grad			A-Grad
IE 8.0		A-Grad	A-Grad		
IE 7.0		A-Grad	A-Grad		
IE 6.0	A-Grad	A-Grad			
Opera 9.6		A-Grad			A-Grad
Safari 3.1				A-Grad	A-Grad

Quelle: http://developer.yahoo.com/yui/articles/gbs/

Eine aktuelle prozentuelle Verteilung der weltweit verwendeten Browser bzw. Betriebssysteme finden Sie unter *http://marketshare.hitslink.com*.

Verwendete Browser und Plattformen im vorliegenden Buch

In diesem Buch wurden die Beispiele und Prototypen auf den folgenden Plattformen und Browsern getestet:

	Windows XP	Windows Vista	Windows 7 (RC)	Apple OS X 10.4	Apple OS X 10.5
Firefox 3.X	X	X	X	X	X
Firefox 2.X	X				X
IE 8.0			X		
IE 7.0	X	X			
IE 6.0	X				
Opera 9.5	X				X
Safari 3.1	X			X	X
Chrome		X			

Wie Sie eine optimale Testumgebung einrichten lesen Sie auf den Seiten 38-40.

	Fedora	Ubuntu
Firefox 3.X	X	X
Firefox 2.X		
IE 8.0		
IE 7.0		
IE 6.0		
Opera 9.5		X
Konqueror		X
Chrome		

Chrome wird zur Zeit von Ubuntu-Entwicklern für Linux portiert. Fortschritte sehen Sie unter *http://launchpad.net* (Suche nach Chromium. Chromium ist das Open Source Projekt hinter Google Chrome). Sie können auch bei dem Projekt mitarbeiten!

Das CSS-Konzept

Was ist CSS?

CSS ist die Abkürzung von Cascading Style Sheets und lässt sich mit „stufenförmige Stilvorlagen“ übersetzen. Die Stufen sagen dem Browser, in welcher Reihenfolge die Regeln abgearbeitet werden sollen. Dazu später mehr. Vereinfacht gesagt geht es bei CSS um die Formatierung von (X)HTML-Elementen. Analog zu einem Textverarbeitungsprogramm können wir im weitesten Sinne von Formatvorlagen sprechen – auch wenn wir mit CSS noch weit mehr abdecken als nur Überschriften-Stilvorlagen.

Foto: www.flickr.com/stephanemartin

Eine Kaskade ist ein treppenförmiger Wasserfall. CSS-Regeln ergießen sich wie ein Wasserfall – also über mehrere Stufen – über das XHTML-Dokument.

Mit CSS formatieren Sie zum Beispiel:

- Texte: Schriftgröße, -art, -farbe, Stil, Laufweite von Zeichen und Wörtern, Hyperlinkfarben, Unterstreichungen von Hyperlinks
- Absatzbreite, Zeilenhöhe,
- Positionierung von Elementen und deren Abstände
- Rahmenbreite, -farbe, -stil
- Hintergrundfarben, Hintergrundgrafiken
- Tabellen, wie z.B. Zellenrahmen, Zellenabstand
- Formulare
- uvm.

Sie werden im Laufe dieses Buchs erkennen, wie viel Spaß es macht, mit CSS zu arbeiten, wie viel einfacher das Designen damit ist, und Sie werden sehr bald in der Lage sein, eigene kreative Ideen umzusetzen.

Welche CSS-Version verwenden wir?

Aktuell verwenden wir die CSS-Version 2. Die Version 2.1 ist ein Release Candidate, also noch nicht offiziell freigegeben, und die Version 3 ist bereits in Vorbereitung zur Veröffentlichung. Einige spannende Ein- und Ausblicke darauf haben wir Ihnen im Anhang zusammengestellt (siehe Seite 366).

Wozu CSS?

Zunächst einmal: Erst mit dem Einsatz von CSS arbeiten Sie korrekt! Nicht mit (X)HTML werden Websites gestaltet, sondern mit CSS.

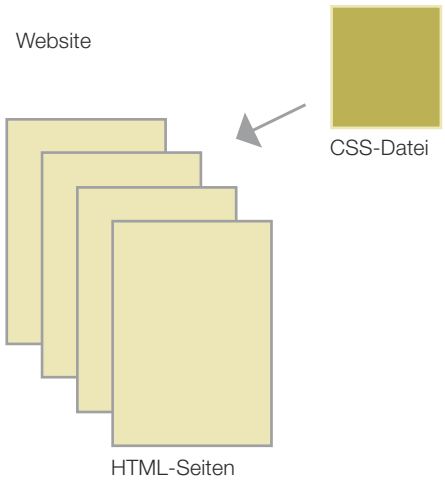
Mit (X)HTML stößt man als DesignerIn bald an die Grenzen des Machbaren – einfach deshalb, weil die Hypertext Markup Language ursprünglich auch nur dazu gedacht war, Seiten zu strukturieren und nicht zu layouten. Niemand hat mit der enormen Geschwindigkeit gerechnet, mit der sich das Web entwickelte. Webdesigner suchten verzweifelt nach Möglich-

keiten, die Seiten attraktiver zu machen, und sehr bald kam die Idee auf, Websites mit Tabellen und leer.gifs zu gestalten. Doch das war seitens des World Wide Web Consortium (W3C, *www.w3c.org*) nie so geplant und es gab und gibt schon seit ewigen Zeiten zahlreiche Gegner dieser Technik. Bereits 1997 (!) forderte David Siegel, der Erfinder des leer.gifs: „Say goodbye to the single-pixel GIF!“ und geiselte sich selbst für seine Erfindung: „Das Web ist ruiniert und ich habe es ruiniert.“ Doch zum Glück ist das Web ja nun bald gerettet, auch wenn es lange gedauert hat, bis ein Umdenken erfolgte.

Durch den Einsatz von CSS sind Sie in der Lage, die HTML-Seite sozusagen von außen zu layouten. Das HTML-Rohgerüst wird durch CSS in Form gebracht. Liegen die CSS-Informationen in einer eigenen, externen Datei, steuert diese zentral das Design. Im Quellcode der HTML-Datei sieht man neben ein paar Tags dann nur noch Content, also relevante Inhalte – optimal für Screenreader (Blindenprogramme) und Suchmaschinen. Selbstverständlich können die CSS-Informationen auch in mehreren Dateien liegen, um so z.B. für mehr Übersicht zu sorgen oder einzelne Browser gezielt anzusprechen.

2009 feiert das W3C das 13 (!)-jährige Bestehen von CSS. Faszinierend, wie lange es gedauert hat, bis sich CSS allgemein durchsetzte! Wann haben Sie denn zum ersten Mal von CSS gehört oder es angewendet?

Welche Vorteile bringt die Layoutierung mit CSS?



Wenn Sie mit CSS arbeiten, sorgen Sie für eine saubere **Trennung von Inhalt und Design** und gehen damit nicht nur konform mit den Empfehlungen des W3C, sondern erzielen auch zahlreiche Vorteile:

- Übersichtlicher und schlanker Quellcode, schnellere Ladezeit
- Flexible und ökonomische Designs
- Zentrale Steuerung des Designs
- Bessere Indexierung durch Suchmaschinen
- Accessibility – optimaler Zugang, z.B. für Blindenprogramme
- Usability – höhere Benutzerfreundlichkeit und damit zufriedenere UserInnen

Gerade wenn Sie als LehrerIn, TrainerIn oder WebdesignerIn tätig sind – also ein/e MultiplikatorIn sind –, haben Sie eine große Verantwortung, Webstandards einzuhalten. Der Appell von Jens Meiert, Spezialist für benutzerfreundliches und technisch hochwertiges Webdesign, und South African Technology Events macht es deutlich:

- Konzentrieren Sie sich aufs Lernen.
- Konzentrieren Sie sich auf Barrierefreiheit.
- Konzentrieren Sie sich auf Performance.
- Konzentrieren Sie sich auf Semantik.
- Konzentrieren Sie sich auf Validierung.
- Konzentrieren Sie sich auf Wartbarkeit.
- Konzentrieren Sie sich auf Zusammenarbeit.
- Konzentrieren Sie sich auf Dokumentation.
- Konzentrieren Sie sich auf Qualität.
- Konzentrieren Sie sich aufs Unterrichten.

Sagen Sie's weiter. Auch per stiller Post. *http://coderesponsibly.org/de/*

Webstandards sind unter *http://www.webstandards.org* zusammengefasst. Mit deutscher Übersetzung.

HTML, XHTML, XML – was ist was?

HTML (Hypertext Markup Language) ist eine Auszeichnungssprache (Markup Language). Sie dient zum Strukturieren von Texten und ist in der Lage, per Verweis Bilder, Multimedia-Dateien etc. einzubinden (Referenz). So weit so gut, sagen Sie, so oder so ähnlich hätten Sie das auch beschrieben. Das klassische HTML – jetzt sind wir bei Version 4.0 – basiert auf SGML (Standard Generalized Markup Language). SGML ist die Basis aller Auszeichnungssprachen.

XML (Extensible Markup Language) hat sich von SGML emanzipiert. Auf Basis von XML werden eigene Sprachen definiert und mit XSL (Extensible Stylesheet Language) formatiert. Aber Sie können XML getrost vergessen, wenn Sie herkömmliche Websites bauen.

XHTML ist eine Weiterentwicklung von HTML und basiert auf XML. Bei XHTML sind wir zurzeit bei Version 1.1.

Bitte schlagen Sie mit all den verwirrenden Spezifikationen nicht gleich das Buch zu! Wichtig für die praktische Anwendung ist, dass Sie sich für den Aufbau des HMTL-Gerüsts die neuen Spezifikationen von XHTML ansehen. Es sind nur ein paar und sie sind schnell gemerkt, hier die wichtigsten im Überblick:

- Sie dürfen alle gewohnten HTML-Tags verwenden, außer jenen, die aus Transitional und Frameset kommen: Das sind z.B. `<font>`, `<frameset>` oder zahlreiche Attribute wie `<type>`, `<align>` oder `<bgcolor>`, `<vspace>` etc. Diese dürfen Sie unter XHTML nicht verwenden, da die Eigenschaften mittels CSS umgesetzt werden können. Eine Liste aller unerwünschten (deprecated = missbilligten) Elemente und Attribute finden Sie auf *http://de.selfhtml.org/html/referenz/varianten.htm*.
- Inline-Elemente dürfen nicht ohne Block-Elemente notiert werden, Also darf z.B. <img /> nicht alleine stehen, sondern muss z.B. innerhalb von platziert werden.
- Bei HTML war es egal, ob Sie Tags GROSS oder klein geschrieben haben, bei XHTML müssen Sie alles in Kleinbuchstaben schreiben.

  ```
  <H1>Herzlich Willkommen</H1>
  ```

  muss also nun so geschrieben werden:

  ```
  <h1>Herzlich Willkommen</h1>
  ```

- Leere Elemente wie `<img />`, `<br />`, `<input />`, `<hr />` etc. benötigen unter XHTML einen abschließenden Schrägstrich mit einem Leerzeichen davor, also so:

```
<img src="bild.jpg" />
<br />
```

- Unter HTML benötigen manche Tags wie `<body>`, `<option>`, `<td>` nicht zwingend ein abschließendes End-Tag `</body>`. Das ist unter XHTML nicht erlaubt. Hier müssen alle Tags geschlossen werden:

```
<table>
 <tr>
  <td>Stadt, Land</td>
 </tr>
</table>
```

- In XHTML müssen Sie Anführungszeichen für den Wert eines Attributs verwenden, dies ist unter HTML optional:

```
<a href="http://www.website.com">
```


XHTML verlangt einen eigene Dokumenttyp-Deklaration – mehr dazu siehe Seite 44.

Eine vollständige Auflistung für die Unterschiede HTML – XHTML finden Sie hier: *http://de.selfhtml.org/html/xhtml/unterschiede.htm*.

Semantisches XHTML & Mikroformate

Mit CSS gestalten Sie Websites, indem Sie die XHTML-Struktur ansprechen. Voraussetzung für gelungenes CSS ist somit der sinnvolle Aufbau der Webpage mit semantisch passendem (X)HTML. Ihre (X)HTML-Seite ist die Basis, das Fundament für das Layouten mit CSS.

Haben Sie bisher Websites mit Tabellen aufgebaut, dürfen und müssen Sie alles vergessen, was Sie bisher gelernt haben, und sich auf eine völlig neue Strukturierung einlassen. Es genügt nicht, einfach statt <table>-Tags `<div>`-Tags einzusetzen. Relevant ist die jeweilige **Bedeutung** (= Semantik) des Inhalts, der auf der Webseite präsentiert wird, unabhängig davon, wie er später im Layout dargestellt wird.

Verwenden Sie daher passende HTML-Notationen zu den jeweiligen Inhalten. Haben Sie beispielsweise bis jetzt eine Überschrift so wie im folgenden angegebenen Beispiel notiert ...

```
<p><font size="5">Fotografie</font></p>
```

... dann ist dies aus zwei Gründen nun nicht mehr zulässig: Erstens dürfen Sie bei der Dokumenttyp-Deklaration Strict (siehe Seite 44) kein `<font>` mehr verwenden und zweitens stimmt die Angabe semantisch nicht.

Verwenden Sie daher stets korrekterweise das passende Element für eine Überschrift, z.B. `<h1>`:

```
<h1>Fotografie></h1>
```

Bekannterweise sieht `<h1>` im Rohzustand beispielsweise so aus:

Fotografie

Mit CSS gestaltet, kann die Überschrift auch so aussehen:

Fotografie

Ohne eine einzige Änderung am HTML-Element oder Inhalt gestalten Sie mittels CSS aus der Überschrift zum Beispiel auch das:

» fotografie

Der HTML-Code **(1)** bleibt schlank und aussagekräftig, die Formatierung kommt komplett aus dem CSS **(2)**. Das Zitate-Zeichen „»“ stellen IE 6 und IE 7 übrigens leider nicht dar (siehe auch Tipp am Seitenrand). Dies ist somit bereits ein Beispiel dafür, welche Probematiken auf den CSS-Designer zukommen, doch damit wollen wir uns im Moment noch nicht beschäftigen. Wenn Sie die Inhalte korrekt bezeichnen, stellen Sie sicher, dass der Content

```
<style type="text/css">
<!--
    h1 {
        color:#ce7428;
        font-style:italic;
        font-size:45pt;
        text-transform: lowercase;
        letter-spacing: -0.08em;
        background-color:#bcb04e;
        padding: 20px 0 0 15px;
        font-family:arial,verdana;
        border-top:solid 8px #c6c6c6;
        border-bottom: dotted 4px #c6c6c6;
        width:300px;
}

h1:before {
content:"\00BB"" ";
color:#fff;

}
-->
</style>

</head>
    <body>
        <h1>Fotografie</h1>
    </body>

</html>
```

2

1

Semantik (gr. „bezeichnen“), auch Bedeutungslehre, nennt man die Theorie oder Wissenschaft von der Bedeutung der (sprachlichen) Zeichen. Aus: *de.wikipedia.org*

Beispiele für den Aufbau semantisch sinnvoller Websites finden Sie auf Seite 48 (Struktureller Aufbau Prototyp 1) bzw. ab Seite 128, wo wir mit der Umsetzung der Prototypen starten.

Einige der Hürden, die uns der Internet Explorer bei der Darstellung bereitet, umgehen wir mit einem Skript, welches auf Seite 95 (Browser-Weichen & Hacks) beschrieben ist. Code, der mithilfe dieses Skripts im IE6 bzw. 7 dann also doch richtig dargestellt werden kann, versehen wir mit diesem Symbol:

IE8-Script

auch ohne CSS verständlich dargestellt wird, eine wichtige Voraussetzung für optimale Zugänglichkeit durch Screenreader und Suchmaschinen.

„Ich weiß nicht, ob es besser wird, wenn es anders wird. Aber es muss anders werden, wenn es besser werden soll."
Georg Christoph Lichtenberg

Mikroformate

Ein weiteres Beispiel für semantisch sinnvolles (X)HTML ist, wenn Sie Kontaktinformationen in ein `<address>`-Element setzen:

```
<address>
    Firma Muster <br />
    Mustergasse 12 <br />
    0101 Metropolis <br />
    office@mustergasse.at <br />
    www.mustergasse.at <br />
</address>
```

Firma Muster
Mustergasse 12
0101 Metropolis
office@mustergasse.at
www.mustergasse.at

Eine Verfeinerung der XHTML-Tags stellen **Mikroformate** dar. Diese teilen Mensch und Maschine noch konkreter mit, was die Inhalte aussagen. Es gibt Mikroformate zur Auszeichnung von Adressen, Terminen, favorisierten Websites, Bewertungen und sogar der Art der sozialen Verbindung, die Sie zu einem Menschen pflegen. So verwendet zum Beispiel XING auf den Profilseiten Mikroformate, die dadurch via Suchmaschinen leichter auffindbar werden.

Eine Liste aller Mikroformate finden Sie auf *http://microformats.org*.

Beispiel:

```
<address class="vcard">
    <span class="org">Firma Muster</span>
    <span class="n">Maria Muster</span>
    <span class="street-address">Mustergasse 12</span>
    <span class="postal-code">0101</span>
    <span class="locality">Metropolis</span>
    <span class="email">office@mustergasse.at</span>
    <span class="url">http://www.mustergasse.at</span>
</address>
```

» Das Block-Element `<address>` leitet einen Absatz über Kontaktinformationen zum/zur Seiteninhaber/-inhaberin ein.

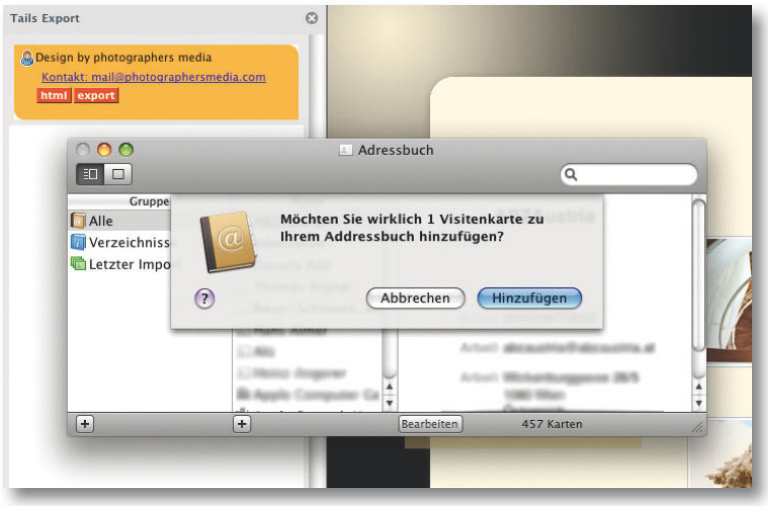
http://microformats.org/wiki/hcard

- Mit hCard (`class="vcard"`) erzeugen Sie eine digitale Visitenkarte, die sich automatisch in Adressbücher importieren lässt.
- `<span>` als Inline-Element umschließt die jeweiligen Detailinformationen des Kontakts.
- Die Klassen (z.B. `class="email"`) definieren, welcher Art der Inhalt ist: `org` für Organisation, `street-address` für den Straßennamen etc.

Durch die Anwendung von Mikroformaten werden diese Informationen website-übergreifend sowohl für Mensch als auch für Maschine leichter identifizierbar. Eine Liste der gängigen Mikroformate finden Sie auf *www.microformats.org*.

Wo Sie dieses Symbol sehen, sind Mikroformate nicht weit ...

Für Alltags-WebsurferInnen und Web-AnwenderInnen sind die Mikroformate leider noch nicht wirklich relevant, da sie von kaum einem Browser angezeigt werden. Sichtbar machen können Sie die mit Mikroformat versehenen Elemente einer Website über die Firefox-Erweiterung (Extension) Operator oder auch Tails Export. Einmal installiert lösen die beiden Skripte auf Wunsch verschiedene Aktionen aus, sobald Mikroformate auf einer Website gefunden werden. Download von *https://addons.mozilla.org/de/firefox/addon/4106* (Operator) bzw. *https://addons.mozilla.org/de/firefox/addon/2240* (Tails Export).



Entdeckt die Firefox-Erweiterung (hier Tails Export) auf einer Website Mikroformate, leuchtet in der rechten unteren Ecke des Browsers das Mikroformate-Symbol auf. Die Informationen von `<address>` lassen sich in verschiedene Anwendungen exportieren, hier z.B. in das Mac-Adressbuch.

Grundsätzlich sollten Sie die Tags `<span>` und `<div>` sparsam einsetzen, denn der Code soll ja schlank und übersichtlich bleiben. Ergänzend muss dazu gesagt werden, dass der Einsatz von Mikroformaten abseits von Google und außerhalb von Web-2.0-Anwendungen wie z.B. Blogs, Social Networking etc. zuletzt eher wieder in den Hintergrund getreten ist und diese nur von sehr wenigen WebdesignerInnen eingesetzt und genutzt werden.

Einen interessanten Artikel zum Thema Google & Mikroformate finden Sie hier: *http://microformats.org/wiki/google-search*.

Tag-Wolke

Tag-Wolken bzw. Tag-Clouds kennt man aus dem Web. Sie stellen alphabetisch sortierte Schlüsselwörter gesammelt dar, wobei die Schriftgröße eines Worts die Häufigkeit, mit der ein Wort auf der Website vorkommt bzw. angeklickt wurde, zeigt. Bei der vorliegenden „statischen" Tag-Wolke bieten wir Ihnen einen Überblick über die aus unserer Sicht wichtigsten Begriffe des vorangegangenen Abschnitts.

<address> <h1> Accessibility Barrierefreiheit Bedeutung Block-Elemente Cascading Style Sheets Content CSS de.selfhtml.org Design Element HTML identifizierbar Indexierung Inhalt Inline-Elemente Klassenname Layout leichter Mikroformate Mensch passend Qualität Screenreader Semantisches HTML Struktur Suchmaschinen Tag Usability Validierung Vererbung Webstandards XHTML XML Zugänglichkeit

Falls Sie unterrichten, hier ein Vorschlag für eine Übung: Notieren Sie die Begriffe aus der Tag-Wolke auf Kärtchen und lassen Sie die Studenten und Studentinnen eines ziehen und den Begriff erklären. Die anderen raten. „Tag-Activity"!

Accessibility & Usability

Web für alle!

Manche Menschen, die sich im Web bewegen, haben zum Beispiel Schwierigkeiten, die Maus zu bedienen oder eine 12-Pixel-Schrift zu lesen. Daher ist es dringend nötig, bei der Programmierung auf die unterschiedlichen Bedürfnisse der BesucherInnen einer Website Rücksicht zu nehmen und damit allen Menschen Zugang zu Informationen zu ermöglichen. Unter Accessibility (engl. Zugänglichkeit) oder auch Barrierefreiheit versteht man „die Kunst, Webseiten so zu gestalten, dass jeder sie nutzen und lesen kann“ (aus: *www.barrierefreies-webdesign.de*).

Mit Opera lässt sich eine Website in unterschiedlichen Modi darstellen. So gibt es z.B. den Modus benutzerfreundlich, zugänglich. Damit sehen Sie die Website in reiner Textdarstellung.

Was kann man also als WebdesignerIn tun? Sehen wir uns doch gleich praxisnahe Lösungen für die beiden eingangs erwähnten Handycaps an: Um eine Navigation ohne Maus zu ermöglichen, definieren Sie sogenannte Accesskeys – also Zugriffstasten. Mit deren Hilfe aktivieren UserInnen den Fokus mittels Tastenkombination:

```
<li><a href="/" accesskey="A">Startseite</a></li>
<li><a href="/profil/" accesskey="B">Profil</a></li>
<li><a href="/kontakt/" accesskey="C">Kontakt</a></li>
```

Accesskeys in Opera: sehr benutzerfreundlich

Die Verwendung der **Accesskeys** (AK) ist nun abhängig vom jeweiligen Browser: So drückt man beim Internet Explorer Alt + AK + ↵, bei Mozilla Firefox unter Windows und Linux ⇧ + Alt + AK, am Mac jedoch Ctrl + AK, bei Opera zumindest einheitlich auf allen Betriebssystemen: ⇧ + Esc etc. Aufgrund der unterschiedlichen Aktivierung und auch Belegung innerhalb der Hilfsprogramme raten manche Experten und Expertinnen von Accesskeys ab. Wenn Sie Zugriffstasten anbieten, dokumentieren Sie deren Gebrauch am besten auf der Website und formatieren Sie deren erfolgreiche Aktivierung. Dafür gibt es die Pseudoklasse `:focus` (Seite 73).

Firefox 2.0 hat Probleme mit numerischen Accesskeys, verwenden Sie daher besser alphanumerische Zeichen.

Mit der Verwendung von relativen **Größenangaben für Texte** gewährleisten Sie, dass UserInnen die Schriftgröße nach ihren Bedürfnissen via Browser skalieren können. Obschon die meisten Browser auch absolute Angaben wie pt, pc etc. vergrößern bzw. verkleinern, streikt hier der Internet Explorer. Gehen Sie daher beispielsweise wie folgt vor:

```
body {font-size: 100.01%;}
h1 {font-size: 1.5em;}
p {font-size: 0.9em;}
```

(Mit 100.01% für `<body>` umgehen Sie einen Bug von Safari und Opera.) Alle nach `<body>` folgenden Kindelemente – hier zum Beispiel `<p>` – orientieren sich nun relativ zu der festgelegten Basisgröße und sind darüber hinaus über den jeweiligen Browser-Menübefehl skalierbar.

„em“ (sprich: „em“) als relative Einheit bezieht sich auf die Schriftgröße des jeweiligen Elements (`h1`, `h2`, `p` etc) oder des Elternelements (z.B. `body`). Einige anschauliche Bespiele dazu finden Sie auf *http://www.1ngo.de/web/em.html*.

Richtlinien & Levels

Die beiden Beispiele für Accesskeys und relative Schriftgröße veranschaulichen freilich nur einen kleinen Teil dessen, was Sie bei barrierefreier Webprogrammierung berücksichtigen sollten. Bis dato gibt es zwei Richtlinien, die die Web Accessibility Initiative (WAI), eine Initiative des W3C, unter der Bezeichnung Web Content Accessibility Guidelines 1.0 und 2.0 veröffentlicht hat.

Für die Umsetzung der Richtlinien gibt es unterschiedliche Prioritäten:

» Priorität 1: Die Richtlinie **muss** umgesetzt werden.
Beispiel: Bieten Sie Textalternativen für Inhalte, die selbst kein Text sind, also für Bilder, Videos, Buttons etc.

```
<img src="bild.png" alt="Klassenfoto" />
```

» Priorität 2: Die Richtlinie **sollte** umgesetzt werden.
Beispiel: Benennen Sie Links so, dass deren Ziel klar hervorgeht. Verlinken Sie also nicht inhaltsleere Phrasen wie „mehr dazu“, „klicken Sie hier“, sondern formulieren Sie das Ziel aus und verwenden Sie zusätzlich das `title`-Attribut:

```
Senden Sie uns Ihre Anfrage über unsere
<a href="..." title="Kontakt">Anfrage</a>
```

» Priorität 3: Diese Richtlinie **kann** umgesetzt werden.
Beispiel: Website darf nicht mehr als 3x pro Sekunden blinken.

Es würde den Rahmen dieses Buchs sprengen, hier alle Richtlinien zur barrierefreien Programmierung anzuführen; eine übersichtliche Liste der wichtigsten Richtlinien finden Sie unter *http://www.barrierefreies-web-design.de* (auf Deutsch) bzw. das englische Original mit Beispielen unter *http://www.w3.org/WAI*.

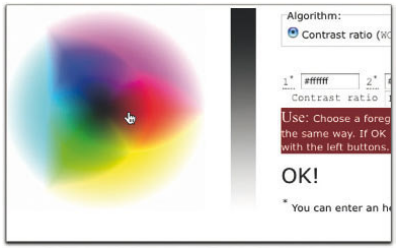
Sind alle Richtlinien der Priorität 1 korrekt umgesetzt, darf die Website das Icon für Level A tragen, bei Erfüllung von Priorität 1 + 2 das Icon Level Double-A und bei Erfüllung aller Prioritäten das Icon Level Triple-A.

Testen – testen – testen

Im Web gibt es zahlreiche – auch kostenlose – Möglichkeiten, Ihre Programmierung auf Zugänglichkeit zu testen. Beachten Sie dabei jedoch, dass Accessibility-Validatoren die Seite nur nach „maschinellen“ Kriterien überprüfen. Bleiben Sie daher auch bei „grünem Licht“ kritisch!

Der WAVE-Validator (*http://wave.webaim.org*) analysiert den Aufbau der Seite nach verschiedenen Kriterien, so zum Beispiel die Strukturierung der Seite nach der Gliederung mit Überschriften. Dabei werden Indikatoren eingebettet, die Abschnitte anzeigen. Die Nur-Text-Darstellung simuliert die entsprechende Browserdarstellung bzw. zeigt auch, wie Screenreader die Seite durchlaufen. Das W3C empfiehlt übrigens *http://www.cynthiasays.com*.

Mit dem Accessibility Color Wheel (*http://gmazzocato.altervista.org/colorwheel/wheel.php*) testen Sie Vordergrund- und Hintergrundfarben auf ihren Kontrast. Ein eingeblendetes OK zeigt einen guten Kontrast bzw. verschwindet bei zu geringem Kontrast.



Auf der Website *http://www.absv.de/sbs/sbs_intro.html* finden Sie einen Simulator, der die Wahrnehmung mit Sehbehinderungen wie Grauer Star, Grüner Star, Nachtblindheit usw. anhand von verschiedenen vorgegebenen Abbildungen darstellt. Sie können hier jedoch nicht eine bestimmte Website testen.

Wenn Sie Websites mit semantisch sinnvollem Markup aufbauen und wirklich sauberes CSS programmieren, haben Sie schon automatisch einen Teil der WAI-Richtlinien erfüllt. Mehr zur Validierung von Markup und CSS lesen Sie auf Seite 26.

Mit *http://colorfilter.wickline.org* legen Sie verschiedene Farbfilter über Ihre Website und sehen so, wie farbenfehlsichtige Menschen die Website wahrnehmen. Die verfügbaren Filter reichen von verschiedenen Rot/Grün- über Blau/Gelb-Farbschwächen bis hin zu Monochrom-Filter für Farbenblinde.

Eine Möglichkeit, sich in die Rolle eines blinden oder stark sehbehinderten Menschen hineinzuversetzen, bietet die Installation eines Screenreaders. Dieser liest u.a. Websites vor. Mehr Infos dazu finden Sie auf Seite 58.

Mit der Einhaltung der Accessibility-Richtlinien sorgen Sie nicht nur dafür, dass Menschen mit Einschränkungen Ihre Website besuchen und verstehen können, sondern teilweise gleichzeitig auch für eine gute Usability. Um was es dabei geht, sehen wir uns nun an.

Usability

Unter dem Begriff Usability versteht man die Gebrauchstauglichkeit einer Hard- oder Software.

Im Webbereich betrifft das vor allem die folgenden Bereiche:

- Aufbau und Darstellung des **Inhalts**: angenehme Schriftgröße und nicht zu viele unterschiedliche Schriftarten, guter Kontrast Schrift – Hintergrund, knappe, präzise Formulierungen, kurze Sätze, Auflockerung langer Inhalte mit Zwischenüberschriften, Listen und Nummerierungen.
- Übersichtliche, nicht allzu verschachtelte **Navigationsstruktur**. Ihre UserInnen sollten stets wissen, wo sie sich befinden, wie sie eine Ebene hinauf bzw. zur Startseite kommen und wo sie eventuell nötige Zusatzinformationen erhalten.
- Beachten Sie **Gestaltungsrichtlinien**. Blättern Sie renommierte Zeitschriften, Illustrierte und Zeitungen durch – betrachten Sie

Der Usability-Monitor 2008 hat diesmal die Vertragsshops von großen Telekommunikationsanbietern unter die Lupe genommen. Die nicht besonders erfreulichen Ergebnisse finden Sie auf *http://www.syzygy.de/* als PDF zum Download.

Tipp für den Unterricht: Suchen Sie Websites heraus und definieren Sie Ziele, z.B. ein Handbuch finden, eine Software herunterladen, den richtigen Ansprechpartner für ein bestimmtes Anliegen finden etc. Je zwei Studenten bzw. Studentinnen lösen eine Aufgabe, wobei eine Person klickt, laut vorsagt, was sie tut und denkt und die zweite Person dokumentiert die Wege. Abschließend erfolgt eine kleine Präsentation der Usability-Ergebnisse.

das Layout – orientieren Sie sich an grundlegenden grafischen Regeln: Gitterlayout, Ausrichtung von Elementen in Spalten, wenige verschiedene, zueinander passende Farben, dezente Hervorhebungen, Platzierung der Bilder etc. Natürlich ist das Web vordergründig nicht mit Printmedien zu vergleichen, doch es gibt allgemeingültige Designrichtlinien, die auch im Screendesign für einen gelungenen Auftritt sorgen. Surfen Sie durch das Web, bleiben Sie kritisch – auch bei Webauftritten großer Firmen.

- Setzen Sie – zumindest die wichtigsten – Zugänglichkeitsrichtlinien (vgl. Seite 20) um.
- Denken Sie bei der Konzeption stets an die Personen, die Ihre Website besuchen. Verwenden Sie deren „Sprache“, erklären Sie Fachbegriffe, suchen Sie passende Fotos zur Untermalung des Inhalts.
- Halten Sie Konventionen ein – also erfüllen Sie gewissermaßen Erwartungen, die Ihre BenutzerInnen bereits haben. So sollte z.B. die Navigation als horizontale oder vertikale Liste angeordnet werden, untergeordnete Einträge befinden sich direkt darunter und nicht „irgendwo“ auf der Seite.
- Führen Sie die BesucherInnen durch Ihre Website. Bieten Sie ausreichend interne Verlinkungen – aber nicht so viele, dass sie verwirren.

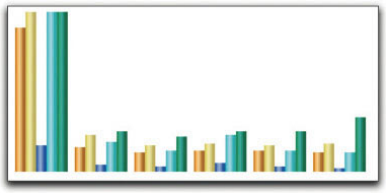
Die 58 wichtigsten Usability-Tipps hat Jens Meiert unter *http://meiert.com/de/publications/articles/20060508/#toc-browse-search* veröffentlicht.

Unter *http://www.fit-fuer-usability.de/* finden Sie ebenfalls interessante Artikel und Ratschläge zu dem Thema.

Usability-Analysen & Tests

Die Gebrauchstauglichkeit kann auf verschiedenste Art und Weise getestet werden – das reicht von einfachen Befragungen (siehe auch Tipp) über Blickbewegungsregistrierung bis hin zu Videoaufzeichnungen im Labor, abhängig davon, wie komplex die Software bzw. Webanwendung ist. Es gibt auch zahlreiche Firmen, die Usability-Analysen professionell durchführen.

Eine – wenn auch zugegeben recht oberflächliche – Usability-Analyse bietet die Zugriffsstatistik Ihrer Website. Die Zugriffsstatistiken zeigen unter anderem Einstiegs- und Ausstiegsseiten, die Verweildauer, Fehlermeldungen, Suchbegriffe, Herkunftsländer etc. Daraus können Sie schlussfolgern, welche Inhalte von den BesucherInnen angenommen werden und welche nicht und gegebenenfalls die Website daraufhin optimieren. Gute Provider bieten üblicherweise Statistiken automatisch mit an, erkundigen Sie sich danach. Alternativ binden Sie eine Zugriffsstatistik selbst ein, suchen Sie im Web nach dem Begriff „zugriffsstatistik“ – es gibt zahlreiche Informationen und Anbieter dazu.



A AA AAA Accessibility Accesskeys Barrierefreiheit besondere Bedürfnisse Beeinträchtigung Color Wheel Farben Farbenfehlsichtigkeit Farbenblindheit führen Gebrauchstauglichkeit Handycap Kontrast Konventionen Leseschwäche Listen Maus Navigation Nummerierungen Priorität Screenreader Schriftgröße Sehbehinderung Sprache testen übersichtlich Usability-Analyse Usability-Monitor Validierung Vorlesen WAVE-Validator Web Accessibility Initiative (WAI) Zugänglichkeit Zugriffsstatistik Zwischenüberschriften

Usability-Test ganz einfach: Fordern Sie Ihre Bekannten auf, Ihre Website unter Ihrem Beisein aufzurufen, sagen Sie nichts und beobachten Sie über die Schulter hinweg einfach nur, wie die Person durch die Seite navigiert. Dabei sieht man meist schon sehr gut was ankommt bzw. wo es Schwachstellen gibt (erfolglose Klickversuche).