



Mikroformate

Autor:	Stefan Münz <stefan.muenz@googlemail.com>
Stand:	16.03.2007
Dokumentversion:	1
Beschreibung:	Mikroformate sind auf Konvention beruhende, HTML-basierte Beschreibungen bestimmter Daten, zum Beispiel von Kontaktdaten. Mikroformate sind vor allem für Webanwendungen wie etwa Suchmaschinen interessant, die aus den Mikroformaten datenfeldorientierte Informationen ziehen können. Das vorliegende Dokument erklärt die grundsätzliche Funktionsweise von Mikroformaten und stellt die wichtigsten Formate an Hand von Beispielen vor.
Download-Adresse:	http://webkompetenz.googlegroups.com/web/Mikroformate.pdf

Hinweise

Dieses Dokument fällt unter die Creative-Commons-Lizenz in folgender Variante:

Namensnennung-NichtKommerziell-KeineBearbeitung 2.0 Deutschland
<http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Details und Ergänzendes:

Das Ausdrucken, Kopieren und Weitergeben dieses Dokuments ist erlaubt.

Das Veröffentlichen im Web, in Intranets, in Mailinglisten oder ähnlichen Medien ist erlaubt. Ebenfalls erlaubt ist das Veröffentlichen auf Datenträgern wie CD-ROMs oder DVDs. Wenn diese Datenträger kommerziell orientiert sind, ist die Veröffentlichung nur dann erlaubt, wenn dieses Dokument nicht Hauptinhalt des Datenträgers ist.

Das Dokument darf als Schulungsunterlage verwendet werden.

Das wörtliche Übernehmen von einzelnen Passagen dieses Dokuments in eigene Texte ist in Form eines Zitats erlaubt. Das Zitat muss die Quelle des Dokuments eindeutig benennen. Das Dokument wird durch die oben angegebene Download-Adresse adressiert. Zur Kennzeichnung der Passage innerhalb des Dokuments können wie üblich Seitenzahlen oder Kapitel-/Abschnittsnummern verwendet werden.

Im Dokument enthaltene Code-Beispiele, die in separaten Kästen notiert sind, können ohne Quellenangabe kopiert und verwendet werden. Nur wenn Code-Beispiele selbst wieder zu Beispielen einer anderen Dokumentation, eines Buches oder dergleichen werden sollen, ist eine Quellenangabe erforderlich.

Der Autor übernimmt keine Haftung für die Richtigkeit der Inhalte dieses Dokuments. Das Verwenden von Wissen oder Beispielen aus diesem Dokument erfolgt in jedem Fall auf eigene Gefahr.



Inhalt

1 Allgemeine Fragen und Antworten.....	3
1.1 Was sind Mikroformate?.....	3
1.2 Wo kann ich Mikroformate in Aktion sehen?.....	4
1.3 Welche Mikroformate gibt es?.....	5
1.4 Was sind Design-Patterns bei Mikroformaten?.....	7
1.5 Wie werden Mikroformate standardisiert?.....	8
2 Beispiele für Mikroformate.....	9
2.1 Mikroformat für Kontaktdaten (hCard).....	9
2.2 Mikroformat für Kalendereinträge (hCalendar).....	12
2.3 Mikroformat für Reviews (hReview).....	16
2.4 Mikroformate für Hyperlinks.....	19
3 Weiterführendes zu Mikroformaten.....	24
3.1 Microformats.org.....	24
3.2 Deutschsprachige Quellen.....	24
3.3 Internationale Quellen.....	25



1 Allgemeine Fragen und Antworten

Wenn Sie sich mit Mikroformaten beschäftigen wollen, müssen Sie zunächst einmal über solide HTML-4-Kenntnisse verfügen, denn Mikroformate sind nichts anderes als spezielle, aber standardkonforme Auszeichnungen in HTML.

1.1 Was sind Mikroformate?

HTML ist bekanntlich die Basis-Auszeichnungssprache für Webseiten. Wer schon mit HTML gearbeitet hat, weiß, dass man damit Texte strukturiert auszeichnen kann. HTML kennt verschiedene Überschriftenebenen, Textabsätze, Aufzählungslisten, nummerierte Listen, Glossarlisten, Tabellen. Weiterhin diverse Inline-Auszeichnungen, etwa für hervorgehoben Text, Zitate oder Abkürzungen. Außerdem Auszeichnungen für Verweise, für Grafikreferenzen, Multimediareferenzen, Formulare, ihre Dialogelemente und verweisensensitive Grafiken. Und last but not least eine Reihe von Auszeichnungen für Metadaten, logische Beziehungen zu anderen Inhalten, sowie zum Einbinden von Scripts und Style-sheets.

Dennoch klagen viele, die mit HTML arbeiten, immer wieder darüber, dass es für viele Fälle keine ordentlichen Auszeichnungen gibt. Eine News etwa, die aus Schlagzeile, Anlesertext, Langtext und Autorenangabe besteht, lässt sich in HTML nur durch Elemente abbilden, die in keiner Weise darauf hindeuten, dass es sich um eine News handelt. Wer mit HTML ein Meeting-Protokoll auszeichnen will, hat keine adäquaten Elemente, um anwesende Teilnehmer, Tagesordnungspunkte, vorgebrachte Argumente, Ergebnisse und Beschlüsse auszuzeichnen. Und für Kontaktdaten gibt es zwar ein Element namens `<address>...</address>`, doch keine Substruktur für Daten wie Anschrift, Telefon, E-Mail usw.

Genau an dieser Stelle setzen Mikroformate an. Wer jetzt allerdings irgendetwas Besonderes und Geheimnisvolles erwartet, wird enttäuscht sein. Denn die meisten Webentwickler basteln sich fortwährend ihre eigenen Mikroformate. Wenn Sie beispielsweise ein modernes Webseitenlayout mit sauber getrenntem HTML und CSS entwerfen und die HTML-Struktur notieren, dann werden Sie vermutlich HTML-Konstrukte wie `<div id="header">...</div>`, ``, `<h3 class="newstitle">...</h3>` verwenden. Sie verwenden also allgemeine Elemente wie `div` und/oder Universalattribute wie `id=` oder `class=`. Die Werte, die Sie diesen Attributen zuweisen, sind Namen, die Sie für sinnvoll (bedeutungsvoll) halten.

An dieser Stelle liegt der Knackpunkt. Die von Ihnen vergebenen Namen sind für Sie und auch für andere Menschen sinnvoll weil verständlich. Doch der verarbeitenden Software ist es egal, ob Sie `<h3 class="newstitle">...</h3>` verwenden oder `<h3 class="bla-bla">...</h3>`. Der HTML-Parser eines Browsers ist zufrieden, weil er ein syntaktisch korrekt notiertes `h3`-Element mit einem erlaubten Attribut `class=` erkennt. Der Wert des `class`-Attributs mag im Hinblick auf CSS interessant sein. Doch den HTML-Parser interessiert das nicht weiter. Er weiß also nichts davon, dass Sie mit der Überschrift den Titel



einer News auszeichnen. Eine Software kann ohne weitreichende künstliche Intelligenz einfach nicht wissen, was Sie mit selbstvergebenen Namen „meinen“.

Was aber, wenn man sich weltweit einigen würde, News-Schlagzeilen in HTML immer und überall mit `<h3 class="newstitle">...</h3>` auszuzeichnen? Es würde ein **Standard** entstehen. Und genau diese Art von Standards bezeichnet man als Mikroformate.

1.2 Wo kann ich Mikroformate in Aktion sehen?

Suchmaschinen senden bekanntlich „Wanderprogramme“ aus, Robots, Spider oder Crawler genannt. Diese Programme lesen den Quelltext der Webseiten und folgen den darin enthaltenen Links. Auch der übrige HTML-Code wird dabei analysiert. So werden beispielsweise Überschriften erkannt, und ihr Inhalt erhält bei der Suche beispielsweise mehr Relevanz als normaler Text. Was aber, wenn Suchmaschinen auch Mikroformate erkennen?

Eine Suchmaschine dieser Art ist *Technorati Kitchen*:

<http://kitchen.technorati.com/search/>

Geben Sie zum Testen im Suchfeld der Seite beispielsweise den Namen einer großen Stadt wie Berlin, Hamburg oder München ein. Sie erhalten dann alle Kontakte, Kalenderereignisse und Reviews (z.B. von Restaurants), die Technorati Kitchen auf Grund von Mikroformaten in der betreffenden Stadt finden konnte. Wenn Sie Suchtreffer anklicken, landen Sie auf Seiten, in denen die Mikroformate angewendet werden. Es handelt sich um ganz normale Webseiten, genauer um Blogs. Letzteres liegt daran, dass Technorati eine Suchmaschine speziell für Blogs ist. Es gehört jedoch nicht viel Phantasie dazu sich auszumalen, wie alle Suchmaschinen schnell zu weltweiten Adressbüchern oder Veranstaltungskalendern werden könnten, wenn die Autoren von Webseiten solche Daten konsequent mit den entsprechenden Mikroformaten auszeichnen.

Neben Suchmaschinen können aber auch Web-Browser Nutzen aus Mikroformaten ziehen. Wenn Sie den Firefox-Browser einsetzen, können Sie mögliche Anwendungsfälle bereits über eine Erweiterung ausprobieren.

Laden Sie sich dazu die Tails Firefox Extension herunter:

<http://blog.codeeg.com/tails-firefox-extension-03/>

Klicken Sie den Link für die Erweiterung (xpi-Datei) an. Die Erweiterung wird installiert. Um die Erweiterung verwenden zu können, benötigen Sie außerdem sogenannte Tails Scripts. Es handelt sich um JavaScripts. Der Autor der Erweiterung bietet selbst solche Scripts an.

Webseite mit Tails Scripts:

<http://blog.codeeg.com/tails-firefox-extension-03/tails-scripts/>

Klicken Sie die Scripts an, die Sie verwenden möchten. Das jeweils angeklickte Script wird im Quelltext angezeigt. Über eine Schaltfläche oben auf der Webseite lässt sich das Script installieren.

Beenden Sie Firefox, nachdem Sie die Erweiterung und die gewünschten Tails Scripts installiert haben. Starten Sie den Browser anschließend neu. Die Erweiterung ist nun verfü-



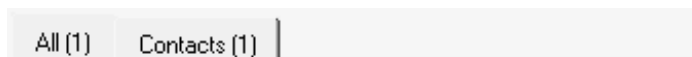
bar. Sie hat sich unauffällig in der Statuszeile von Firefox in Form eines Symbols eingemis-

 bedeutet: auf der aktuell angezeigten Webseite wurden keine Mikroformate erkannt.

 bedeutet: auf der aktuell angezeigten Webseite wurden Mikroformate erkannt.

Rufen Sie zum Testen eine Seite auf, die Mikroformate enthält, beispielsweise:
<http://www.rss-blogger.de/b2e/blogs/>

Klicken Sie das farbige Symbol an. Ein Overlay-Fenster wird angezeigt:



Objects Found on this Page:



Siegfried Hirsch

hhs social software systems

[Send Email](#) | [Map with Google](#) | [Add to Address Book](#)

Klicken Sie den Namen an. Die Kontaktdaten werden angezeigt. Hinter den angezeigten Kontextlinks wie z.B. *Send Email* oder *Map with Google* verbergen sich diejenigen Tails Scripts, die Sie heruntergeladen und installiert haben.

Um das Overlay-Fenster der Erweiterung wieder zu schließen, klicken Sie erneut das Symbol in der Statusleiste an.

Für die Version 3 des Firefox-Browsers ist die eingebaute Unterstützung aller gängigen Mikroformate und anderer Bestandteile des „semantischen Webs“ bereits fest vorgesehen. Eine direkte Unterstützung durch verbreitete Browser ist allerdings letztendlich auch erforderlich, wenn Mikroformate jemals so bedeutsam werden sollen, wie sie jetzt schon gehandelt werden.

1.3 Welche Mikroformate gibt es?

Die Frage ist deshalb nicht eindeutig zu beantworten, weil sie genaugenommen lauten muss: welche Mikroformate haben bereits den Status eines Standards? Es bleibt also an der Frage hängen, was man unter „Standard“ versteht. Keines der existierenden Mikroformate ist in einem Standard festgeschrieben, der die „industrial strength“ etwa von HTML 4.01, XHTML 1.0 oder auch HTTP 1.1 hat. Das könnte sich jedoch ändern, wenn die Akzeptanz von Mikroformaten erst einmal wächst. Mikroformate sind offene Standards, die unter einer Creative-Commons-Lizenz stehen. Die Web-Anlaufstelle für den Standardisierungsprozess ist die Microformats-Community-Site <http://microformats.org/> (weitere Details dazu in Abschnitt 1.5). Dort werden folgende Mikroformate bevorzugt aufgelistet:



- ▶ **hCard:** Mikroformat zur Auszeichnung von Kontaktdaten. Datenstrukturierung, Feldnamen und mögliche Werte orientieren sich dabei am vCard-Format, einem Standard des Internet Mail Konsortiums, der im RFC-Dokument 2426 festgeschrieben ist. Nutzen: wenn von Web-Autoren konsequent angewendet, könnten Suchmaschinen zu weltweiten Adressbüchern werden und Browser persönlich relevante Kontakte von besuchten Webseiten sammeln und für andere Anwendungen exportieren. Spezifikations-Adresse: <http://microformats.org/wiki/hcard>
- ▶ **hCalendar:** Mikroformat zur Auszeichnung von Kalenderereignissen (Termine, Veranstaltungen usw.). Datenstrukturierung, Feldnamen und mögliche Werte orientieren sich dabei am iCalendar-Format, einem Standard des Internet Mail Konsortiums, der im RFC-Dokument 2445 festgeschrieben ist. Nutzen: wenn von Web-Autoren konsequent angewendet, könnten Suchmaschinen zu weltweiten Event-Kalendern werden und Browser persönlich relevante Kalenderereignisse von besuchten Webseiten sammeln und für andere Anwendungen exportieren. Spezifikations-Adresse: <http://microformats.org/wiki/hcalendar>
- ▶ **hReview:** Mikroformat zur Auszeichnung von Bewertungen, Rezensionen oder Testberichten zu Produkten, Diensten, Veranstaltungen usw. Nutzen: wenn von Web-Autoren konsequent angewendet, könnten Suchmaschinen gezielt alle „Reviews“ etwa zu einem bestimmten Produkt, zu einem bestimmten Dienstanbieter, zu einem bestimmten Buch oder zu einer bestimmten Veranstaltung auflisten. Spezifikations-Adresse: <http://microformats.org/wiki/hreview>
- ▶ **VoteLinks:** Mikroformat zur wertenden Auszeichnung von Hyperlinks, d.h. um beispielsweise anzugeben, ob man das Verweisziel zustimmend oder ablehnend gegenübersteht. Nutzen: Suchmaschinen könnten die in Hyperlinks enthaltenen Voting-Informationen beispielsweise als Page-Ranking-Kriterium für die bewerteten Verweisziele berücksichtigen. Spezifikations-Adresse: <http://microformats.org/wiki/vote-links>
- ▶ **rel-licence:** Mikroformat zur Auszeichnung eines Hyperlinks mit der Bedeutung „dieser Link führt zu einer Seite mit Lizenzinformationen zum aktuellen Inhalt“. Nutzen: bei konsequenter Anwendung durch Web-Autoren könnten Suchmaschinen beispielsweise Statistiken darüber erstellen, wie verbreitet bestimmte Lizenzformen sind. Spezifikations-Adresse: <http://microformats.org/wiki/rel-licence>
- ▶ **rel-tag:** Mikroformat zur Auszeichnung eines Hyperlinks mit der Bedeutung „dieser Link führt zu einer Seite mit weiteren Links zum gleichen Tag (Stichwort)“. Zugleich wird damit aber auch auf HTML-Code-Ebene ein „Tag“ für die aktuelle Webseite gesetzt. Ziele sind entsprechende Adressen bei Anbietern wie *del.icio.us*, *Technorati* oder *Mr. Wong* oder auch „Labels“ etwa innerhalb von Blogs. Nutzen: Suchmaschinen, die tag-basiert arbeiten, können solche Links erkennen und berücksichtigen. Spezifikations-Adresse: <http://microformats.org/wiki/rel-tag>
- ▶ **XFN:** Mikroformat, um in einem Hyperlink zu einem persönlichen Ziel (Homepage, Blog, Mailadresse) anzugeben, in welcher Beziehung der Linksetzende zum Eigentümer des Verweisziels steht. Nutzen: wenn von Web-Autoren konsequent angewendet, könnten Suchmaschinen aus den so gewonnenen Informationen Beziehungsstrukturen zwischen Personen, Körper-



schaften usw. ermitteln und beispielsweise grafisch abbilden.

Spezifikations-Adresse: <http://www.gmpg.org/xfn/>

- ▶ **XOXO**: Mikroformat zur Auszeichnung von baumartigen Listenstrukturen aller Art. Nutzen: Web-Browser könnten die reine HTML-Auszeichnung nutzen, um explorer-artiges Aufklappen und Zuklappen von Listen zu ermöglichen. Mehr oder weniger aufwändige JavaScript-Lösungen, die heute für solche Zwecke benötigt werden, wären damit hinfällig.
Spezifikations-Adresse: <http://microformats.org/wiki/xoxo>

Neben diesen Mikroformaten gibt es diverse andere, die nicht minder nützlich sein könnten. Einige seien hier noch genannt:

- ▶ **geo**: Mikroformat zur Auszeichnung von geographischen Koordinaten (Längen- und Breitenangaben).
Spezifikations-Adresse: <http://microformats.org/wiki/geo>
- ▶ **Robots Exclusion**: Mikroformat, um Suchmaschinen-Robots mitzuteilen, dass bestimmte Teile einer Webseite nicht indiziert werden sollen. Dieses Mikroformat ist eine Verfeinerung der Datei *robots.txt* und des HTML-Meta-Tags `<meta name="robots">`.
Spezifikations-Adresse: <http://microformats.org/wiki/robots-exclusion>
- ▶ **xFolk**: Mikroformate zur Auszeichnung von „social Bookmarks“. Damit könnten Webanwendungen wie die von *del.icio.us*, *Technorati* oder *Mr. Wong* gewünschte Bookmarks direkt aus dem Quelltext von Webseiten ziehen.
Spezifikations-Adresse: <http://microformats.org/wiki/xfolk>

1.4 Was sind Design-Patterns bei Mikroformaten?

Design-Patterns sind Entwurfsmuster. Im Bereich der Mikroformate sind es allgemeine Rahmenvorgaben, die bei der Entwicklung neuer Mikroformate berücksichtigt werden sollten. Dadurch soll verhindert werden, dass ein Wildwuchs an Auszeichnungen entsteht.

Ein Beispiel: Wenn ein Mikroformat entwickelt werden soll, das einen HTML-Hyperlink mit einer zusätzlichen semantischen Auszeichnung versehen soll, dann gibt dazu ein Design-Pattern vor, dass für solche Zwecke das `rel`-Attribut des `a`-Elements verwendet werden soll. Wenn also etwa jemand ein neues Mikroformat vorschlägt, das einen Hyperlink mit der Bedeutung „mein Verweisziel ist eine offizielle Spezifikation“ auszeichnen soll, so sollte der Vorschlag so etwas wie `rel="specification"` enthalten, nicht aber `class="specification"`.

Eine Übersicht der Design-Patterns, die von der microformats-Community empfohlen werden, ist auf <http://microformats.org/wiki/design-patterns> zu finden.



1.5 Wie werden Mikroformate standardisiert?

Damit unterschiedliche Suchdienste und Web-Browser Mikroformate tatsächlich als solche erkennen, ist es unerlässlich, diese so stark zu standardisieren, dass die Entwicklung von Software auf maßgebliche Spezifikationen zurückgreifen kann.

Es gibt kein abgeschottetes Konsortium, das für die Standardisierung von Mikroformaten zuständig ist. Wer Interesse daran hat, an der Entwicklung und Weiterentwicklung von Mikroformaten mitzuarbeiten, kann sich auf microformats.org am Standardisierungsprozess beteiligen. Gute Englischkenntnisse sind allerdings neben fachlichem Know How erforderlich, da mit dem Standardisierungsprozess viel internationale Kommunikation verbunden ist. Folgende Adressen sollten Sie kennen:

- ▶ <http://microformats.org/discuss/>
Dort werden verschiedene Mailinglisten angeboten:
microformats-discuss ist für allgemeine Diskussionen zu Mikroformaten gedacht,
microformats-new ist für neue Vorschläge für Mikroformate und Diskussionen darüber gedacht,
microformats-dev ist für Entwickler gedacht, die Mikroformate in Webanwendungen oder andere Software einbauen wollen,
microformats-rest ist für Diskussionen zu REST Web Services gedacht (eine gelungene Einführung hierzu finden Sie auf <http://www.oio.de/public/xml/rest-webservices.htm>).
Bei allen Mailinglisten können Sie sich wie bei allen klassischen Mailinglisten üblich an und abmelden.
- ▶ <http://microformats.org/wiki/process>
Diesen Artikel innerhalb des Wikis von microformats.org sollte jeder lesen, der neue Mikroformate entwickeln oder vorschlagen will. Das Wiki ist übrigens auch der Ort, an dem neue Vorschläge bis zur Spezifikationsreife fixiert werden. Bevor in dem Wiki jedoch Artikel mit neuen Vorschlägen angelegt werden, wird in jedem Fall eine Diskussion in der Mailingliste *microformats-new* gefordert. Erst wenn ein Vorschlag innerhalb der Mailinglisten-Community allgemeine Zustimmung findet, kann der eigentliche Spezifizierungsprozess beginnen. Der Artikel beschreibt auch, welche Inhalte innerhalb des Wikis für ein neu zu spezifizierendes Mikroformat angelegt werden müssen, und in welchen Phasen der Spezifizierungsprozess abläuft.



2 Beispiele für Mikroformate

Erläutert werden hier die bislang verbreitetsten Formate.

2.1 Mikroformat für Kontaktdaten (hCard)

Die Basis für das hCard-Format ist das vCard-Format (vCard steht für *visiting card*, also *Visitenkarte*). Dieses Format kennen Sie vielleicht aus Microsoft Outlook oder vergleichbaren Anwendungen für E-Mail und/oder Personal Information Management. Entsprechende Dateien haben die Standardendung *.vcf*. Das vCard-Format ist ein RFC-standardisiertes Textformat zum Speichern von Adress- und Kontaktdaten. Verwendet wird es vor allem für den Datenaustausch. So bieten beispielsweise viele E-Mail-Programme die Möglichkeit an, die eigenen Kontaktdaten als vcf-Datei automatisch an alle ausgehende E-Mails anzuhängen. Der Mail-Empfänger kann die vcf-Datei dann direkt in seine Adressdaten importieren.

Werfen wir zunächst einen Blick auf einen typischen Inhalt einer solchen vcf-Datei:

```
BEGIN:VCARD
VERSION:3.0
N:Lyse;Anna
FN:Anna Lyse
ORG:Beispiel-AG
ADR;WORK;;;Phantasiestr. 1;Nirgendwo;;D-99991
TEL;WORK;VOICE:+49 999 12345
TEL;TYPE=CELL:+49 144 234422
TEL;WORK;FAX:+49 999 54321
URL:http://de.example.com/
EMAIL;INTERNET:anna.lyse@example.com
END:VCARD
```

Jede Zeile der Datei beschreibt ein Datenfeld. Zu Beginn steht der Name des Datenfeldes in Großbuchstaben. Getrennt durch einen Doppelpunkt, folgt der zugehörige Feldwert. Sowohl Feldnamen als auch Feldwerte können nochmals strukturell unterteilt sein. Solche Unterstrukturen werden durch Semikolonzeichen angegeben. Im obigen Beispiel etwa ist das Feld `TEL;WORK;VOICE` ein solcher Fall. `TEL` steht für „Telekommunikationsdatum“, `WORK` für „Büro/Geschäftlich“ und `VOICE` für „Festnetztelefon“. Ebenso wäre auch die Angabe `TEL;HOME;VOICE` möglich, nämlich für die private Festnetztelefonnummer.

Das Mikroformat hCard ist nichts anderes als eine Vorgabe, wie das vCard-Format in gültiges HTML umgesetzt werden soll. Da das gesamte vCard-Format unterstützt wird, können Sie sich, wenn Sie Spezifikationen mögen, mit den möglichen vCard-Feldnamen und ihrer korrekten Verwendung vertraut machen. Die offizielle Spezifikation und Dokumentation des vCard-Formats finden Sie unter:

<http://www.ietf.org/rfc/rfc2426>
RFC (Request for Comment) 2426



Eine andere Möglichkeit, um sich mit dem vcf-Format vertraut zu machen, besteht darin, einfach Adressdatensätze aus einem E-Mail-Programm in vcf-Dateien zu exportieren. Voraussetzung ist natürlich, dass das Mailprogramm dies unterstützt. Mit einem Texteditor oder Text-Viewer können Sie anschließend die exportierte vcf-Datei ansehen.

Umsetzung ins hCard-Mikroformat

Sie können Kontaktdaten im hCard-Mikroformat im HTML-Quelltext einer Webseite dort platzieren, wo Sie möchten, also dort, wo die Daten auch angezeigt werden sollen.

Dabei haben Sie überraschend viele Freiheiten. Bei der Wahl der HTML-Elemente sind Sie nämlich vergleichsweise frei. Wichtig ist nur, dass Sie die logische Struktur der Daten abbilden, und dass Sie dabei die Feldnamen in Form von `class`-Attributen in HTML wiedergeben (kleingeschrieben).

Damit eine HTML-Baumstruktur als vCard-Abbildung erkennbar ist, muss sie in ein äußeres HTML-Element eingeschlossen werden, das `class="vcard"` zugewiesen bekommt. Beispiel:

```
<div class="vcard">
<!-- Inhalt der Kontaktdaten -->
</div>
```

Da es in HTML ein `address`-Element gibt, bietet sich aber auch dieses Element an:

```
<address class="vcard">
<!-- Inhalt der Kontaktdaten -->
</address>
```

Beide Beispiele sind erlaubt, genauso wie `` oder `<body class="vcard">` (letzteres bietet sich an, wenn eine Webseite nichts anderes als Kontaktdaten enthält).

Die vollständige Umsetzung unserer Beispiel-vCard könnte wie folgt aussehen:

```
<address class="vcard">
  <span class="n" style="display: none">
    <span class="family-name">Lyse</span>
    <span class="given-name">Anna</span>
  </span>
  <span class="fn">Anna Lyse</span><br>
  <span class="org">Beispiel-AG</span><br>
  <span class="adr work">
    <span class="street-address">Phantasiestr. 1</span><br>
    <span class="country-name">D</span>-
    <span class="postal-code">99991</span> <span
class="locality">Nirgendwo</span><br>
  </span>
  <span class="tel home voice">+49 999 12345</span> (privat)<br>
  <span class="tel cell">+49 144 234422</span> (geschäftlich)<br>
  <span class="tel work fax">+49 999 54321</span> (Fax geschäftlich)<br>
  <a class="url"
href="http://www.example.com/">http://www.example.com/</a><br>
```



```
<a class="email"
href="mailto:anna.lyse@example.com">anna.lyse@example.com</a>
</address>
```

Im Browser sieht das beispielsweise so aus:

```
Anna Lyse
Beispiel-AG
Phantasiestr. 1
D- 99991 Nirgendwo
+49 999 12345 (privat)
+49 144 234422 (geschäftlich)
+49 999 54321 (Fax geschäftlich)
http://www.example.com/
anna.lyse@example.com
```

Selbstverständlich können Sie die `class`-Angaben zu den Detaildaten der Kontaktdaten auch mit CSS formatieren. Auch zusätzliche Attribute wie `id=`, `style=` oder auch Event-Handler sind innerhalb der Elemente, die ein vcard-spezifisches `class`-Attribut enthalten, kein Problem. Das Gleiche gilt für ergänzenden Text zwischen solchen Elementen. In unserem Beispiel stehen etwa die Texte `(privat)`, `(geschäftlich)` usw. für die Kenntlichmachung der Telefoniedaten auf der Webseite außerhalb der für das Mikroformat bedeutungstragenden Elemente.

Beim Namen haben wir ein wenig getrickst. Für das Mikroformat eignet sich eine Struktur aus einem Elternelement mit `class="n"`, das Kindelemente mit den `class`-Namen `given-name` und `family-name` enthält. Für das Mikroformat ist es jedoch sinnvoll, außerdem noch ein Element mit `class="fn"` (`fn` steht für *full name*) zu notieren, das den gesamten Namen noch mal in der normalen Darstellungsform enthält. Damit der Name aber nicht doppelt auf der Webseite erscheint, haben wir das komplexe `span`-Konstrukt mit `style="display: none"` für die Anzeige unsichtbar gemacht.

Daten, die zur Anschrift gehören, können wie in unserem Beispiel in ein Element mit `class="adr"` als Kindelemente eingeschlossen werden. Einem `class`-Attribut dürfen auch mehrere Klassennamen zugeordnet werden, und zwar durch Leerzeichen getrennt. In unserem Beispiel haben wir mit `class="adr work"` kenntlich gemacht, dass es sich um die Geschäftsadresse handelt. Zusätzlich könnte auch ein paralleles Konstrukt mit `class="adr home"` notiert werden.

Auch bei den Telefonieangaben haben wir im Beispiel von den Mehrfachangaben zum `class`-Attribut Gebrauch gemacht, z.B. mit ``. Bei Telefonieangaben ist noch zu beachten, dass die Feldinhalte, also der Elementinhalt, im internationalen kanonischen Darstellungsformat für Telefonnummern notiert werden sollte. Dieses Format wird von manchen Programmen erkannt, die zwischen Landesvorwahl, Ortsvorwahl und Rufnummer unterscheiden.



Für Internet-Adressen bietet sich selbstredend das `a`-Element inklusive `href`-Attribut an. So werden im Beispiel die Mailadresse und die Webadresse der Firma anklickbar gemacht.

Es gibt zahlreiche Möglichkeiten, Kontaktdaten im Rahmen der erlaubten hCard-Syntax noch ganz anders zu strukturieren. Unser obiges Beispiel ist nichts weiter als ein Muster, das für viele Praxisfälle anwendbar ist.

2.2 Mikroformat für Kalendereinträge (hCalendar)

So wie das hCard-Mikroformat auf dem vCard-Format basiert, lehnt sich das hCalendar-Mikroformat an das iCalendar-Format an, das ebenso wie das vCard-Format ein Industriestandard ist. Allerdings unterstützen die meisten Anwendungen nur vCalendar, den Vorläufer des iCalendar-Formats. Auch dieses Format kennen Sie vielleicht aus Microsoft Outlook oder vergleichbaren Anwendungen als Export- und Import-Format für Terminkalendereinträge. Entsprechende Dateien haben die Standardendung `.vcs` oder `.ics`.

Um das hCalendar-Format besser zu verstehen, betrachten wir zunächst zwei Kalendereinträge im iCalendar-Format – zunächst ein Tagesereignis und dann einen Termin.

```
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VEVENT
DTSTART:20070406
DTEND:20070406
SUMMARY:Karfreitag
END:VEVENT
END:VCALENDAR
```

```
BEGIN:VCALENDAR
VERSION:2.0
BEGIN:VEVENT
DTSTART:20070510T200000
DTEND:20070510T235900
TZOFFSETFROM:+0100
LOCATION:Stragula Realwirtschaft
SUMMARY:Jahresversammlung 2007 der gescheiterten Existenzen
ATTENDEE;ROLE=OWNER;CN=Rainer Wahnsinn;MAILTO=rwahnsinn@example.com
ATTENDEE;ROLE=ATTENDEE;CN=Anna Lyse;MAILTO=alyse@example.com
END:VEVENT
END:VCALENDAR
```

Die gewählten Beispiele sind recht einfach und beschreiben nur wenige typische Daten. Die Spezifikation des iCalendar-Formats kennt wesentlich mehr Eintragungsmöglichkeiten, um auch Aspekte wie Teilnehmeranforderungen, wiederkehrende Termine, Alarmeinstellungen usw. zu speichern.

Die offizielle Spezifikation und Dokumentation des iCalendar-Formats finden Sie unter:



<http://tools.ietf.org/html/rfc2445>
RFC (Request for Comment) 2445

Die Datenstruktur weist die gleichen Merkmale auf wie diejenige des vCard-Formats. Der gesamte Kalendereintrag wird durch `BEGIN:VCALENDAR` eingeleitet und durch `END:VCALENDAR` abgeschlossen. Durch `BEGIN:VEVENT` bzw. `END:VEVENT` wird der Kalendereintrag typisiert. Termine und Tagesereignisse fallen unter die Kategorie `VEVENT`. Andere Kategorien sind `VTODO` (für Aufgaben) und `VJOURNAL` für Journaleinträge.

Terminanfang und -ende werden durch `DTSTART` bzw. `DTEND` definiert. Bei Tagesereignissen wie im ersten unserer Beispiele wird das Datum ausschließlich in Ziffernform notiert, wobei die ersten vier Ziffern für das Jahr stehen, die nächsten zwei Ziffern für den Monat und die letzten beiden für den Monatstag. Bei Uhrzeit-Terminen wie in unserem zweiten Beispiel folgt hinter der Datumsangabe ein `T` und dahinter die Uhrzeit als Folge von sechs Ziffern, wobei die ersten beiden Ziffern die Stunde bedeuten, die zweiten beiden die Minuten und die letzten beiden die Sekunden.

Bei Uhrzeitangaben ist auch die Angabe der Zeitzone sinnvoll. Dies geschieht durch `TZOFFSETFROM`. Deutschland beispielsweise hat die Zeitzone + 1 Stunde gegenüber der UTC-Zeit (Greenwich-Zeit). Notiert wird das in der Form `+0100`.

Wenn es für einen Termin einen Veranstaltungsort gibt, kann dieser durch `LOCATION` angegeben werden. Falls mit dem Ereignis auch eine Webadresse verknüpfbar ist, kann zusätzlich ein Feld `URL` notiert werden. `SUMMARY` ist das Feld für den Titel des Kalenderereignisses. Längere Beschreibungen können in einem zusätzlichen Feld `DESCRIPTION` stehen. Komplexer wird es wieder bei `ATTENDEE`-Feldern. In diesen Feldern können Teilnehmer eingetragen werden, beispielsweise bei Meeting-Ereignissen. Für jeden namentlichen Teilnehmer wird ein `ATTENDEE`-Feld notiert. Im Unterfeld `ROLE` kann die Funktion des jeweiligen Teilnehmers angegeben werden. Die Wertzuweisung `OWNER` steht für den Organisator, Einladenden oder Hauptverantwortlichen des Ereignisses, und die Zuweisung `ATTENDEE` für normale Teilnehmer ohne spezielle Funktion. In den Feldern `CN` kann der Name des Teilnehmers angegeben werden, und bei `MAILTO` dessen Mailadresse.

Umsetzung ins hCalendar-Mikroformat

Beim Umsetzen dieser Datenstruktur in ein spezifikations-konformes hCalendar-Mikroformat haben Sie wieder ähnliche Freiheiten wie beim hCard-Format. Die Umsetzung geschieht durch Klassennamen, die über das `class`-Attribut in HTML notiert werden. Allerdings werden auch einige weitere Konventionen erwartet. Betrachten wir zunächst eine Umsetzung unseres ersten Beispiels (Kalendereintrag für Karfreitag 2007):

```
<ul class="vcalendar">
  <!-- andere Feiertage -->
  <li class="vevent">
    <span class="summary">Karfreitag</span>:
    <abbr class="dtstart" title="20070406"
      style="border: none;">4. April 2007</abbr>
    <abbr class="dtend" title="20070406"
      style="display: none;">4. April 2007</abbr>
  </li>
</ul>
```



In dem Beispiel nehmen wir eine Aufzählungsliste mit Feiertagen an. Jeder Listeneintrag soll aus der Bezeichnung eines Feiertags und seinem Datum für das Jahr 2007 bestehen. Damit repräsentiert jedes `li`-Element ein Ereignis und bekommt aus diesem Grund im Start-Tag das Attribut `class="vevent"` zugewiesen. Dass die gesamte `ul`-Liste übrigens das Attribut `class="vcalendar"` erhält, ist keine zwingende Vorschrift. Der Klassenname `vevent` ist derjenige, der auslesender Software signalisiert, dass ein Kalendereintrag folgt.

In unserem Beispiel müssen die iCAL-Felder `SUMMARY`, `DTSTART` und `DTEND` für den Kalendereintrag umgesetzt werden. `SUMMARY` wird zu `class="summary"`, wobei es egal ist, welches zugehörige HTML-Element gewählt wird (in unserem Fall ist es ein `span`-Element).

`DTSTART` und `DTEND` werden ebenfalls zu `class`-Attributzuweisungen. Hierbei ist jedoch zu beachten, dass die hCalendar-Spezifikation für diese Felder das `abbr`-Element von HTML verlangt. Der Wert des Feldes wird, da die Notationsform zwar gut maschinenlesbar, für Menschen jedoch eine Zumutung ist, nicht als Elementinhalt, sondern im `title`-Attribut notiert. Im Elementinhalt des `abbr`-Elements kann dann ein beliebiger gewünschter Text stehen, um die Datums- und/oder Uhrzeitangabe zum Ereignis sprachabhängig und menschenlesbar wiederzugeben.

Da das `abbr`-Element in den meisten Browsern eine automatische Formatierung erhält (gepunktete Unterstreichung), die aber im hiesigen Zusammenhang meist unerwünscht ist, sollte die CSS-Angabe `border: none` dafür sorgen, dass die entsprechende Formatierung unterdrückt wird.

In unserem Fall wollen wir ja nur ein Kalenderdatum ausgeben. Da stört natürlich der Eintrag mit `class="dtend"`. Mit Hilfe der CSS-Angabe `display: none` unterdrücken wir die Anzeige.

Selbstverständlich können Sie solche CSS-Angaben auch in zentralen Stylesheets auf die Klasse anwenden, also in der Form:

```
abbr.dtstart { border: none; }
abbr.dtend { display: none; }
```

Die HTML-Umsetzung unseres zweiten iCalendar-Beispiels im hCalendar-Mikroformat könnte so aussehen:

```
<div class="vevent">
  <h1 class="summary">Jahresversammlung 2007 der
    gescheiterten Existenzen</h1>
  <address class="location vcard"><span class="fn">
    Stragula Realwirtschaft</span></address>
  <table class="times">
    <tr>
      <td>Offizieller Beginn:</td>
      <td><abbr class="dtstart" title="20070510T200000"
        style="border: none;">10.05.2007, 20.00 Uhr</abbr></td>
    </tr><tr>
      <td>Offizielles Ende:</td>
      <td><abbr class="dtend" title="20070510T235900"
        style="display: none;">Mitternacht</abbr></td>
    </tr>
  </table>
```



```
<h2>Teilnehmer</h2>
<ul>
  <li class="attendee">
    <span class="owner"><strong>Leiter:</strong>
    <span class="cn">Rainer Wahnsinn</span>,
    <a class="value" href="mailto:rwahnsinn@example.com">
      rwahnsinn@example.com</a>
    </span>
  </li>
  <li class="attendee">
    <span class="cn">Anna Lyse</span>,
    <a class="value" href="mailto:alyse@example.com">
      alyse@example.com</a>
    </span>
  </li>
</ul>
</div>
```

In diesem Beispiel nehmen wir an, dass eine ganze Webseite für die Anzeige des Ereignisses spendiert wird. Deshalb setzen wir das Feld mit `class="summary"` kurzerhand in eine `h1`-Seitenüberschrift. Ein `div`-Bereich, der alle Auszeichnungen des Ereignisses umspannt, erhält das Signal-Attribut `class="vevent"`.

Interessant wird es bei der Umsetzung des `LOCATION`-Feldes. Für eine tatsächlich existierende Örtlichkeit bietet es sich nämlich an, diese als `hCard`-Mikroformat auszuzeichnen. Deshalb haben wir die Angabe zu `LOCATION` in ein `address`-Element gepackt, dem mit `class="location vcard"` gleich zwei Mikroformat-Klassen zugewiesen werden. Zum einen wird signalisiert, dass dieses Element eine `location`-Angabe innerhalb des Kalenderereignisses enthält. Zum anderen wird ein `vCard`-Mikroformat eingeleitet. In unserem Beispiel haben wir nur den Namen einer Kneipe als Örtlichkeit angegeben. Der Inhalt des `address`-Elements beschränkt sich daher auf ein Element, das mit `class="fn"` den Namen der Kneipe enthält, in der die Veranstaltung stattfindet.

Die Angaben zu Anfang und Ende der Veranstaltung haben wir in eine kleine Tabelle gepackt, um die Inhalte tabulatorisch sauber auszurichten. Die eigentlichen Zeitangaben werden wie gefordert durch `abbr`-Elemente ausgezeichnet, wobei `class="dtstart"` und `class="dtend"` Anfangs- und Endzeitpunkt deklarieren, während das `title`-Attribut die normierte Datums-/Uhrzeitangabe enthält und der Elementinhalt aus einer frei wählbaren, auf der Webseite sichtbaren Darstellungsform der Zeitangaben besteht.

Die Teilnehmer erscheinen in einer Aufzählungsliste. Deren `li`-Elemente signalisieren durch die Angabe `class="attendee"`, dass es sich um Teilnehmerangaben zum Ereignis handelt. Teilnehmernamen und ihre Mailadressen sind im Beispiel durch `vCalendar`-Syntax ausgezeichnet. An diesen Stellen könnten jedoch auch wieder vollständige `vCard`-Mikroformate notiert werden, um zu kennzeichnen, dass es sich um Kontaktangaben handelt.

Im Browser sieht diese HTML-Konstruktion ohne weitere CSS-Formatierung in etwa so aus:



Jahresversammlung 2007 der gescheiterten Existenzen

Stragula Realwirtschaft

Offizieller Beginn: 10.05.2007, 20.00 Uhr

Offizielles Ende: Mitternacht

Teilnehmer

- **Leiter:** Rainer Wahnsinn, rwahnsinn@example.com
- Anna Lyse, alyse@example.com

Mit Hilfe von CSS lässt sich daraus jedoch ein ansprechend gestalteter Inhalt zaubern. Die Feinstrukturierung der Inhalte ist jedenfalls in ausreichendem Maße vorhanden, und das Markup ist ein wichtiger Input für Suchmaschinen, die Mikroformate bereits erkennen.

2.3 Mikroformat für Reviews (hReview)

Das Mikroformat hReview ist im Gegensatz zu hCard und hCalendar noch nicht im Stadium einer offiziellen Spezifikation, sondern noch im Stadium einer Draft, also eines Vorschlags. Einer der Gründe dafür ist möglicherweise, dass es nicht wie die beiden vorgenannten Formate einen anderen, bereits existierenden Standard umsetzt, sondern eine völlige Neuentwicklung ist. Es ist jedoch eines der Mikroformate, die besonders wichtig sind. Denn das hReview-Format ist für alle Arten von Besprechungen, Reviews, Kritiken und Bewertungen gedacht. Es soll Produkttests abdecken, Film- und Buchrezensionen, Restaurant-Tests usw. Unzählige Websites und Blog-Einträge haben solche „Reviews“ zum Inhalt. Durch konsequente Anwendung des hReview-Formats würden daraus strukturierte, feldorientierte Daten, die Suchmaschinen ein enormes Potential an semantischer Information liefern.

An folgendem einfachen Beispiel soll das hReview-Format vorgestellt werden:

```
<div class="hreview">
  <span class="type" style="display: none">product</span>
  <div class="item">
    <a class="url fn summary"
      href="http://www.amazon.de/exec/obidos/ASIN/3827323703">
      Stefan Münz: Professionelle Websites
    </a></div>
  <div class="description"><p>
```



```
Stefan Münz ist als Autor der HTML-Referenz SELFHTML
bekannt geworden. Das vorliegende Buch ist keine
Neuaufgabe dieser Programmier-Sammlung, sondern versteht
sich als Bestandsaufnahme des World Wide Webs. Von dort
aus beleuchtet Münz die gängigen Programmiersprachen und
-techniken, aber auch Designfragen und rechtliche
Implikationen.
(<abbr class="rating" title="5">*****</abbr>)
</p>
</div>
<p class="reviewer vcard">Besprechung von
  <a class="url fn" href="http://www.mediencity.de/">MedienCity</a>,
  <abbr class="dtreviewed" title="200508">August 2005</abbr>
</p>
</div>
```

Ohne weitere CSS-Formatierung sieht das im Browser reichlich unspektakulär aus:

[Stefan Münz: Professionelle Websites](#)

Stefan Münz ist als Autor der HTML-Referenz SELFHTML bekannt geworden. Das vorliegende Buch ist keine Neuaufgabe dieser Programmier-Sammlung, sondern versteht sich als Bestandsaufnahme des World Wide Webs. Von dort aus beleuchtet Münz die gängigen Programmiersprachen und -techniken, aber auch Designfragen und rechtliche Implikationen. (*****)

Besprechung von [MedienCity](#), August 2005

Der gesamte Inhalt der Beispielbesprechung ist in ein `div`-Element mit dem Attribut `class="hreview"` eingeschlossen. Dies ist für auslesende Software das Erkennungs-Attribut für hReview-Mikroformate.

Wie auch bei den Formaten hCard und hCalendar ist die Elementwahl beim hReview-Format abgesehen von einigen Ausnahmen frei. Da es sich bei der Besprechung um ein in sich geschlossenes Konstrukt handelt, das aus diversen, teilweise verschachtelten Einzel-elementen besteht, bietet es sich jedoch an, das gesamte Konstrukt in einen `div`-Bereich einzuschließen.

Im Beispiel folgt als nächstes ein inhaltsleeres `span`-Element, das zusätzlich mit Hilfe der CSS-Angabe `display: none` für die Anzeige im Browser unsichtbar gemacht wird. Dieses Element, das übrigens optional ist, legt die Art des Reviews fest, also eine durchaus wichtige Information, die aber in dieser Form nicht unbedingt angezeigt werden soll. Mit `class="type"` wird der auslesenden Software mitgeteilt, dass hier die Art des Reviews definiert wird. Die Art selbst steht im `title`-Attribut. Möglich sind hier die Wertzuweisungen `product` (für Produktbesprechungen), `business` (für Bewertungen von Unternehmen oder Geschäften) `event` (für Konzert-, Theater-, Kino- und ähnliche Kritiken), `person` (für Personenbewertungen), `place` (für Bewertungen von Restaurants und ver-



gleichbaren Örtlichkeiten), `website` (für Bewertungen von Webangeboten) und `url` (für Bewertungen einzelner Webseiten).

Im Gegensatz zum `type`-Feld ist das `item`-Feld eine Pflichtangabe. Seine Aufgabe ist es, den besprochenen „Gegenstand“ genauer zu spezifizieren. Wie diese genauere Spezifizierung aussieht, kann je nach dem, was gereviewt wird, sehr unterschiedlich ausfallen. In unserem Beispiel, wo ein Buch besprochen wird, bietet sich ein Direktlink zur Buchseite bei einem Buchversand wie Amazon an. Der Link wird im Beispiel mit `class="url fn summary"` ausgezeichnet. Der Klassenname `fn` („full name“) gehört dabei zu denen, die im Zusammenhang mit dem hReview-Format erkannt werden, ebenso wie der Klassenname `summary`. Mit `fn` signalisieren Sie, dass der Inhalt des Elements die volle Namensbezeichnung dessen enthält, was Gegenstand des Reviews ist. Der Klassenname `url` signalisiert, dass es sich um einen Link zu einer produktspezifischen Adresse handelt. An dieser Stelle könnte anstelle des Links auch ein `img`-Element notiert werden, also eine Grafik, die das besprochene Produkt abbildet. Dann sollte statt dem Klassenname `url` der Klassenname `photo` angegeben werden. Die Angabe `summary` hat eine ähnliche Funktion wie `fn`, nämlich die Auszeichnung der Überschrift eines Reviews. In unserem Beispiel sind beide Angaben im gleichen Element notiert. Die Überschrift kann aber selbstverständlich auch separat notiert werden, etwa als `<h3 class="summary">Überschrift des Reviews</h3>`.

Wenn statt eines Produkts eine Örtlichkeit, also etwa ein Lokal oder ein Laden besprochen wird, könnte das Element, das mit `class="item"` markiert wird, anstelle des Links mit der `fn`-Klasse auch ein komplettes hCard-Mikroformat einschließen. Und wenn es sich um eine Konzertkritik handelt, könnte der Inhalt ein komplettes hCalendar-Konstrukt sein, das die Konzerteckdaten enthält. Falls darin ein Veranstaltungsort vorkommt, kann dieser wiederum als hCard-Format innerhalb des hCalendar-Formats ausgearbeitet werden. Sie sehen schon – Mikroformate sind im Prinzip relativ einfach zu verstehen, können aber zu ziemlich komplexen HTML-Markup-Strukturen führen.

Der eigentliche Beschreibungstext des Reviews wird in ein Element eingeschlossen, das mit `class="description"` ausgezeichnet wird. Wenn es sich um eine längere Beschreibung handelt, die aus mehreren Absätzen und vielleicht auch anderen Elementen wie Grafiken besteht, ist das `div`-Element, wie es auch in unserem Beispiel verwendet wird, eine semantisch sinnvolle Wahl.

Bewertungen enthalten häufig auch Noten. Zum Standard des hReview-Formats gehört erfreulicherweise auch ein einheitliches numerisches Bewertungsschema. Dabei sind Punktevergaben von 1.0 (unterirdisch) bis 5.0 (überragend) möglich. Wenn die ganzzahlige Punkteskala also zu grob ist, können Sie auch Werte wie 3.4 oder 1.8 vergeben (Dezimalzeichen ist dabei der Punkt!). Andere Bewertungsschemata sind zwar laut Spezifikation ebenfalls erlaubt, doch es wäre schön, wenn endlich mal alle numerischen Bewertungen einem einheitlichen Schema folgen. Deshalb sei das genannte Schema empfohlen.

Die Bewertung selbst wird durch `class="rating"` markiert. Die numerische Angabe wird im gleichen Element-Tag im Attribut `title=` notiert. Wenn der Elementinhalt aus einer abstrakten Darstellung der Benotung besteht (wie in unserem Beispiel durch die Sternchen), oder wenn der Elementinhalt einfach die `title`-Angabe wiederholt, wird empfohlen, das `abbr`-Element zu verwenden (so wie in unserem Beispiel).

Den Abschluss eines Reviews bildet die Angabe, von wem das Review stammt. Angaben dazu werden in ein Element eingeschlossen, das mit dem Attribut `class="reviewer"`



ausgezeichnet wird. Es ist nicht zwingend erforderlich, ein solches Element zu notieren. Wenn Sie es jedoch notieren, müssen Sie außerdem ein hCard-Mikroformat angeben, das den Review-Autor oder die verantwortliche Redaktion kennzeichnet. In unserem Beispiel haben wir beides vereint durch die Angabe `class="reviewer vcard"`. Die einzige hCard-spezifische Angabe innerhalb des so ausgezeichneten `p`-Elements im Beispiel ist jedoch die Klassenangabe `fn` im Link zu MedienCity. Um die Anwendung der Mikroformate konsequent zu betreiben, wäre es durchaus sinnvoll, an dieser Stelle auch weitere Angaben wie Kontaktadresse, Telefonnummer usw. entsprechend der Regeln des hCard-Formats anzugeben.

Ein typischer Elementinhalt des Elements mit der Auszeichnung `class="reviewer"` ist ein Element, das den Zeitpunkt der Bewertung angibt. Ein solches Element wird durch das Attribut `class="dtreviewed"` gekennzeichnet. Im gleichen Tag wird ein `title`-Attribut notiert, das den Zeitpunkt des Reviews angibt. Dabei gelten die gleichen Notationsregeln wie im hCalendar-Format bei Elementen mit `class="dtstart"` oder `class="dtend"`. Da der zugehörige Elementinhalt in aller Regel die entsprechende menschenlesbare Datums-/Zeitangabe ist, wird empfohlen, für die gesamte Angabe das `abbr`-Element zu verwenden.

2.4 Mikroformate für Hyperlinks

Die zuvor behandelten Mikroformate hCard, hCalendar und hReview sind Datenstrukturen, die aus mehreren, teilweise verschachtelten Feldern bestehen. Je nach Datenumfang können die entsprechenden HTML-Markup-Strukturen durchaus komplex sein. Dagegen sind die Mikroformate für Hyperlinks denkbar simpel. Sie bestehen dank der Design-Pattern-Strategie alle nur in einem `rel`-Attribut, das im einleitenden `<a>`-Tag eines Hyperlinks notiert werden. Deshalb werden sie auch als **Elementar-Mikroformate** bezeichnet. Schema:

```
<a href="http://www.example.org/" rel="microformat">Verweistext</a>
```

Es handelt sich also um ganz gewöhnliche HTML-Verweise. Das `rel`-Attribut ist ein im HTML-Standard erlaubtes Attribut, das auf `a`-Elemente anwendbar ist. Es drückt die Beziehung aus, in welcher der verweisende Inhalt zum Verweisziel steht – also eine Möglichkeit, um Links zu **typisieren**. Das W3-Konsortium nennt in der HTML-Spezifikation eine Reihe von Angaben dafür, wie `top`, `up`, `next`, `prev`, `first`, `last`, `search`, `index` oder `alternate`. Diese Angaben sind jedoch vorzugsweise für Verweisziele innerhalb des eigenen Projekts gedacht. Die HTML-Spezifikation sieht allerdings auch eine Möglichkeit vor, andere als die vom W3-Konsortium vorgegebenen Wertzuweisungen anzuwenden. Autoren, die in ihrem HTML-Code von dieser Möglichkeit Gebrauch machen wollen, sollten ein sogenanntes Profil angeben, in dem der entsprechende Attributwert definiert wird. Bevor wir uns darum kümmern, wollen wir aber erst einige Mikroformate für Hyperlinks kennenlernen:



Hyperlink für Tags (Stichwörter)

Die englische Bezeichnung *tags*, die hierfür verwendet wird, hat nichts mit den Tags in Markup-Sprachen zu tun, sondern bezieht sich auf Wörter und Ausdrücke der natürlichen Sprache. Im Web-2.0-Bereich haben solche Tags bzw. Stichwörter eine enorm wichtige Funktion erlangt. Fast jedes Weblog bietet die Möglichkeit, die Blogeinträge durch Stichwörter zu markieren. Bislang vergebene Stichwörter werden im Navigationsbereich des Blogs angezeigt. Die Häufigkeit, also wie oft ein gleiches Stichwort bislang vergeben wurde, wird entweder durch eine Zahl angegeben, oder die Blog-Software generiert sogenannte *Tag-Clouds* (Stichwortwolken), in denen häufiger vergebene Stichwörter größer dargestellt werden und weniger häufig vergebene kleiner.

Solange ein einziger Autor sich um die Verstichwortung kümmert (wie in den meisten Blogs der Fall), steht er im Prinzip vor den klassischen Problemen der Taxonomie, also einer sinnvollen Systematisierung all seiner Stichwörter. Es gibt jedoch auch zahlreiche Services im Web, die eine Tag-Datenbank aufbauen durch Eingaben unzähliger Benutzer. Beispiele sind social-Bookmark-Services wie del.icio.us oder Mr.Wong, aber auch Blog-Vernetzungs-Services wie Technorati. Da die Verstichwortung in diesem Fall mehr oder weniger chaotisch entsteht, wurde das Kunstwort *Folksonomy* dafür geprägt.

Sowohl innerhalb von Blogs als auch bei Folksonomy-Webservices sind Tags mögliche Verweisziele. Innerhalb eines Blogs werden bei Anklicken eines Tag-Links alle Blogeinträge gefunden, denen das angeklickte Tag zugeordnet ist.

Beispiel im Webkompetenz-Blog:

<http://webkompetenz.blogspot.com/search/label/Mikroformate>

Beispiel beim Webservice Mr. Wong:

<http://www.mister-wong.de/tags/mikroformate/>

Wenn Sie im HTML-Text Links zu solchen Zielen setzen wollen, sollten Sie sich die Mühe machen, das dafür vorgesehene Mikroformat zu verwenden. Beispiele:

```
<a href="http://webkompetenz.blogspot.com/search/label/Mikroformate"
  rel="tag">Mikroformate bei Webkompetenz</a>
<a href="http://www.mister-wong.de/tags/mikroformate/"
  rel="tag">Mikroformate bei Mr. Wong</a>
```

Durch `rel="tag"` kennzeichnen Sie das Verweiszziel als eine (in der Regel) von einer Webanwendung erzeugten Seite, die Inhalte zu dem entsprechenden Stichwort versammelt. Das Stichwort selbst wird als Verweistext notiert.

Hyperlinks für Verzeichnis-Listings

Auch wenn sie nicht mehr so häufig zu finden sind wie in den Anfangsjahren des Web: Verzeichnislistings sind nicht zwangsläufig ein Zeichen vom Unvermögen eines Webmasters. Moderne Webserver wie Apache stellen ein ganzes Arsenal an Möglichkeiten bereit, um Verzeichnislistings attraktiv zu gestalten. Im Browser erscheint ein Verzeichnislisting dann, wenn folgende Voraussetzungen erfüllt sind:

- ▶ Das Verzeichnis muss sich innerhalb der Document Root befinden, also unterhalb des Startverzeichnisses für Webdokumente innerhalb einer Domain oder Subdomain.



- ▶ Das Verzeichnis darf keine in der Konfiguration des Webserver enthaltene Default-Datei enthalten (typische Dateinamen für Default-Dateien sind *index.htm*, *index.html*, *index.php*, *default.htm*, *default.html* usw.).
- ▶ In der Konfiguration des Webserver muss das Verzeichnis für Listings freigegeben sein. Andernfalls erhalten Anwender beim Aufruf eines Verzeichnisses, das keine Datei mit einem der konfigurierten Default-Dateinamen enthält, eine HTTP-Fehlermeldung.

Wenn dem Anwender eine ganze Reihe von Dateien angeboten werden sollen, ist es manchmal sinnvoller, einen Link auf ein über die Webserverkonfiguration gestaltetes Verzeichnislisting zu setzen, als zusätzlich zu den Dateien selbst immer noch eine Webseite mit Links zu diesen Dateien mitzupflegen. Ein Beispiel: <http://strille.net/works/>

Wenn Sie einen Link auf ein solches Verweisziel setzen wollen, ist es hilfreich, wenn Sie den Verweis als Mikroformat typisieren:

```
<a href="http://strille.net/works/" rel="directory">
  Flash project works on strille.net</a>
```

Durch `rel="directory"` kennzeichnen Sie das Verweisziel als Verzeichnis.

Hyperlinks für Lizenzinformationen

Inhalte mit einer gewissen Schöpfungshöhe sollten eine definierte Lizenzform haben, worin festgelegt wird, was mit den Inhalten geschehen darf und was nicht. Eine Möglichkeit, dies in standardisierter Form ohne viel Aufwand anzugeben, ist das Elementar-Mikroformat für Links zu Lizenzinhalten. Solche Links können beispielsweise im Seiten-Fuß anstelle der zumindest in Deutschland juristisch völlig belanglosen ©-Angabe notiert werden.

Das Linkziel kann eine eigene Seite mit rechtlichen Informationen zum Inhalt sein, oder eine allgemeine Lizenzform wie die von Creative Commons (<http://de.creativecommons.org/>). Der Link sollte direkt auf eine Seite mit Lizenzbestimmungen führen. Gerade Creative Commons verhält sich diesbezüglich vorbildlich. Denn wenn Sie über <http://de.creativecommons.org/lizenz.html> eine Lizenzform generieren lassen, wird der einzufügende HTML-Code für den Link am Ende bereits zum Kopieren präsentiert – und dieser Link enthält bereits das Mikroformat für Lizenzdaten. Beispiel eines Lizenz-Links:

```
<a rel="license"
  href="http://creativecommons.org/licenses/by-nd/2.0/de/">
  Creative Commons Attribution-No Derivative Works 2.0 Germany
  License</a>
```

Durch `rel="license"` kennzeichnen Sie das Verweisziel als Lizenzinformation.



Nofollow-Hyperlinks

Seit jeher gibt es ein Meta-Tag in HTML, das Suchmaschinen-Robots anweist, Links auf der entsprechenden Seite nicht weiter zu folgen. Das kann beispielsweise sinnvoll sein, wenn es sich um lauter Links zu Seiten handelt, die ein Login erfordern, oder um Links zu eher privaten Seiten, von denen der Anbieter nicht unbedingt wünscht, dass deren Inhalt weltweit auffindbar ist.

Es gibt jedoch auch ein Mikroformat, das die Nofollow-Funktion auf einzelne Hyperlinks im Dokument überträgt. Damit muss ein Autor sich nicht mehr für eine seitenweise Lösung entscheiden.

Das Nofollow-Mikroformat ist aber auch ein gutes Beispiel dafür, was passieren kann, wenn Mikroformate tatsächlich auf breiter Front ernstgenommen und eingesetzt werden. Zwei Fakten haben einen wilden Streit um dieses unscheinbare Elementar-Mikroformat entfacht: zum einen wird es bereits seit geraumer Zeit von Google interpretiert. Das heißt, die Robots der weltweit führenden Suchmaschine folgen keinen Links mehr, die diese Auszeichnung enthalten. Zum anderen hat Wikipedia seine Software-Installation so konfiguriert, dass alle Links zu externen Seiten, die in Wikipedia gesetzt werden, mit diesem Mikroformat ausgezeichnet werden. Wikipedia will damit unter anderem unlautere Versuche, sich durch Verlinkung bei Wikipedia bei Google einen höheren Page-Rank zu erschleichen, verhindern. Doch auf <http://www.nofollow.net/> wird argumentativ Widerstand organisiert.

Lassen wir das Für und Wider an dieser Stelle außen vor. Das Format selbst lautet (Beispiel):

```
<a href="http://de.wikipedia.org/" rel="nofollow">Wikipedia</a>
```

Durch `rel="nofollow"` weisen Sie Suchmaschinen-Robots an, dem Link nicht zu folgen.

Profil für rel-Angaben

Weiter oben haben wir bereits angedeutet, dass die Wertzuweisungen an das `rel`-Attribut, die von den hier vorgestellten Hyperlink-Mikroformaten benutzt werden, durch ein sogenanntes Profil abgedeckt sein sollten. Das Profil selbst besteht wird im sogenannten XMDP-Format als Datei in einem für einen Webserver zugänglichen Dateibereich (also unterhalb einer Document Root) abgelegt. **XMDP** steht für *XHTML Meta Data Profiles* (Details siehe <http://gmpg.org/xmdp/>). Es handelt sich um eine normale XHTML-Datei, die jedoch nicht so sehr zur Anzeige im Browser gedacht ist, sondern als Referenzinformation für den Browser. Für die vier Hyperlink-Mikroformate, die wir zuvor besprochen haben, können wir folgende Profildatei erzeugen. Die Datei deckt alle vier Formate ab:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head><title>sample HTML profile</title>
<meta http-equiv="Content-Type" content="text/html;
      charset=iso-8859-1" />
</head>
```



```
<body>
<dl class="profile">
  <dt id="rel">rel</dt>
  <dd><p>
    <a rel="help"
href="http://www.w3.org/TR/html401/struct/links.html#adef-rel">
    HTML4 definition of the 'rel' attribute.</a>
    Here are additional values.</p>
  <dl>
    <dt id="tag">tag</dt>
    <dd>Indicates that the referred resource serves as a "tag",
      or keyword/subject, for the referring page.</dd>
    <dt id="directory">directory</dt>
    <dd>Indicates that the referred resource serves as a
      directory listing for the referring page.</dd>
    <dt id="license">license</dt>
    <dd>Indicates that the referred resource is a license for the
      referring page.</dd>
    <dt id="nofollow">nofollow</dt>
    <dd>Indicates that the referred resource was not necessarily
      linked to by the author of the page, and thus said reference
      should not afford the referred resource any additional
      weight or ranking by user agents.</dd>
  </dl>
  </dd>
</dl>
</body>
</html>
```

Es handelt sich also um eine vollständige XHTML-Datei. Der Inhalt besteht aus einer Definitionsliste (`dl`). Die zu definierenden Ausdrücke (`dt`) sind die Attributzuweisungen für das `rel`-Attribut. Die Inhalte der zugehörigen `dd`-Elemente sind frei formulierbare Erläuterungen.

Nun müssen wir die Datei noch im HTML-Code von HTML-Seiten referenzieren, auf denen wir Gebrauch von Hyperlink-Mikroformaten machen. Die Schnittstelle dafür wird im einleitenden `<head>`-Tag notiert. Dort ist laut HTML-Standard das Attribut `profile=` erlaubt, dem die URL-Adresse eines Dokuments zugewiesen wird, das Profile definiert. Angenommen, wir nennen obige Datei *rel-profiles.html*, dann können wir in einem HTML-Dokument, das Hyperlink-Mikroformate enthält, notieren:

```
<head profile="rel-profiles.html">
```

Für die Referenzierung gelten die gleichen Regeln wie beim Referenzieren von Grafiken oder bei Hyperlinks. Es lassen sich also vollständige URLs ebenso angeben wie absolute Pfadnamen ab Document Root oder auch – wie im Beispiel – relative Angaben.



3 Weiterführendes zu Mikroformaten

Abschließend noch einige ausgewählte Webadressen und Buchtitel, die sich mit Mikroformaten näher befassen.

3.1 Microformats.org

<http://www.microformats.org/> (en)

Die offizielle Einstiegsseite rund um Mikroformate

<http://microformats.org/wiki/> (en)

Das Wiki, in dem die Spezifikationen zu Mikroformaten entwickelt werden

<http://microformats.org/discuss/> (en)

Übersichtsseite zu den Mailinglisten, in den u.a. Vorschläge für Mikroformate eingebracht werden können

<http://microformats.org/code/> (en)

Übersichtsseite für eine Reihe von Tools, z.B. formularbasierte HTML-Code-Generatoren für die Formate hCard, hCalendar und hReview

3.2 Deutschsprachige Quellen

<http://mikroformate.de/>

In Blog-Form gehaltene Einzelartikel zum Thema Mikroformate von Michael Jendryschik

<http://mikroformate.org/>

Kollektiv-Blog über Mikroformate von Dennis Blöte, Sebastian Küpers und Marc Tobias Kunisch

<http://groups.google.de/group/mikroformate>

Google-Group zum Thema Mikroformate

<http://blog.dopefreshtight.de/artikel/microformats-und-die-bots/>

Blog-Artikel, der sich kritisch mit Mikroformaten auseinandersetzt



3.3 Internationale Quellen

<http://suda.co.uk/projects/microformats/cheatsheet/microformats.cheatsheet.pdf> (en)
Die Syntax aller wichtigen Mikroformate auf einen Blick (PDF-Dokument)

<http://microformatique.com/> (en)
Blog zum Thema Mikroformate

<http://www.whymicroformats.com/> (en)
Einführung und Blog zu Mikroformaten